



**UNIVERSIDADE FEDERAL DA BAHIA – UFBA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO - DEEC
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA - PPGE**

Ricardo Menezes Prates

**DETECÇÃO DE DEFEITOS E CLASSIFICAÇÃO DE ISOLADORES
DE REDES DE DISTRIBUIÇÃO UTILIZANDO INTELIGÊNCIA
COMPUTACIONAL E GERAÇÃO ARTIFICIAL DE IMAGENS**

TESE DE DOUTORADO

Salvador – Bahia – Brasil, 2020



UFBA



ESCOLA POLITÉCNICA

DETECÇÃO DE DEFEITOS E CLASSIFICAÇÃO DE ISOLADORES DE
REDES DE DISTRIBUIÇÃO UTILIZANDO INTELIGÊNCIA
COMPUTACIONAL E GERAÇÃO ARTIFICIAL DE IMAGENS

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

Ricardo Menezes Prates

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, PPGEE, da Universidade Federal da Bahia, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Eduardo F. de Simas Filho
Rodrigo P. Ramos

Salvador
Dezembro de 2020

P912 Prates, Ricardo Menezes.

Detecção e classificação de isoladores de redes de distribuição utilizando inteligência computacional e geração artificial de imagens / Ricardo Menezes Prates. – Salvador, 2020.

233 f.: il. color.

Orientador: Prof. Dr. Eduardo F. de Simas Filho.

Coorientador: Prof. Dr. Rodrigo P. Ramos.

Tese (doutorado) – Universidade Federal da Bahia. Escola Politécnica, 2020.

1. Redes elétricas. 2. Inspeção visual. 3. Processamento digital de imagens. 4. Redes neurais. I. Simas Filho, Eduardo F. de. II. Ramos, Rodrigo P. III. Universidade Federal da Bahia. IV. Título.

CDD: 621.319

DETECÇÃO DE DEFEITOS E CLASSIFICAÇÃO DE ISOLADORES DE
REDES DE DISTRIBUIÇÃO UTILIZANDO INTELIGÊNCIA
COMPUTACIONAL E GERAÇÃO ARTIFICIAL DE IMAGENS

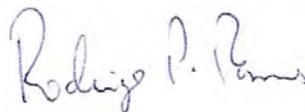
Ricardo Menezes Prates

TESE SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DA BAHIA (PPGEE / UFBA) COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

Examinada por:



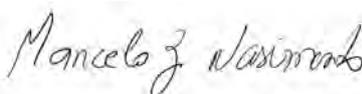
Eduardo Furtado de Simas Filho, D.Sc. - Orientador (PPGEE/UFBA)



Rodrigo Pereira Ramos, D.Sc. - Coorientador (CENEL/UNIVASF)



Antônio Carlos Lopes Fernandes Jr, D.Sc. - Avaliador Interno (PPGEE/UFBA)



Marcelo Zanchetta do Nascimento, D.Sc. - Avaliador Externo (FACOM/UFU)

José Manoel de Seixas, D.Sc. - Avaliador Externo (PEE/COPPE-UFRJ)

SALVADOR, BA - BRASIL

DEZEMBRO DE 2020

Resumo do Tese apresentado ao PPGEE/UFBA como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica (D. Eng.)

DETECÇÃO DE DEFEITOS E CLASSIFICAÇÃO DE ISOLADORES DE
REDES DE DISTRIBUIÇÃO UTILIZANDO INTELIGÊNCIA
COMPUTACIONAL E GERAÇÃO ARTIFICIAL DE IMAGENS

Ricardo Menezes Prates

Dezembro/2020

Orientadores: Eduardo F. de Simas Filho

Rodrigo P. Ramos

Programa: Engenharia Elétrica

As Redes Primárias de Distribuição Aéreas de Energia Elétrica (RPDA) correspondem a um grande percentual dos circuitos elétricos de média tensão presentes mundialmente. Nessas redes, as atividades de Inspeção Visual (IV) são usualmente realizadas sem o auxílio de sistemas automatizados, requerendo um investimento significativo em termos de tempo e de recursos humanos. A IV também está sujeita a avaliações subjetivas e erros humanos, o que pode acarretar em resultados incorretos e/ou imprecisos de inspeção. Nesta perspectiva, o presente trabalho de doutorado se propôs a apresentar um conjunto de técnicas inteligentes direcionadas a inspeção visual automatizada de RPDA, objetivando tanto a identificação dos modelos de isoladores quanto o diagnóstico de suas respectivas inconformidades visíveis a olho nu. Para o referido propósito, foram desenvolvidos diferentes modelos computacionais baseados em técnicas de Processamento Digital de Imagens (PDI) e de Inteligência Artificial (IA). Como parte adicional desta pesquisa, foi criado um *dataset* de imagens de componentes isoladores de RPDA coletados em um estúdio fotográfico e também em uma Rede de Distribuição de Ensaio (RDE),

construída em ambiente externo. Para otimização do desempenho dos modelos computacionais, foram avaliados diferentes tipos de algoritmos inteligentes e de técnicas de aprendizagem profunda, dando ênfase às Redes Neurais Convolucionais (RNC), configurações híbridas e arranjos de Redes Generativas Adversárias (GANs). Além disso, foram implementadas abordagens adicionais, tais como: *data augmentation*; transferência de conhecimento; *multi-task learning*; *background invariance* e geradores de imagens artificiais. Os resultados obtidos indicam desempenhos satisfatórios das técnicas propostas na identificação da classe e presença de inconformidade em componentes isoladores de distribuição, trazendo uma série de inovações em relação a outras abordagens existentes na literatura.

Abstract of Thesis presented to PPGEE/UFBA as a partial fulfillment of the requirements for the degree of Doctor of Electrical Engineering (D. Eng.)

DEFECTS DETECTION AND INSULATOR CLASSIFICATION OF POWER
DISTRIBUTION NETWORKS USING USING COMPUTATIONAL
INTELLIGENCE AND ARTIFICIAL IMAGE GENERATION

Ricardo Menezes Prates

December/2020

Advisors: Eduardo F. de Simas Filho

Rodrigo P. Ramos

Department: Electrical Engineering

Overhead Power Distribution Lines (OPDL) correspond to a large percentage of the medium-voltage electrical systems available worldwide. In these networks, Visual Inspection activities (VI) are usually performed without resorting to automated systems, requiring a significant investment of time and human resources. VI are also vulnerable to subjective evaluation and human error, which can lead to incorrect and/or inaccurate inspection results. In this perspective, the present PhD study proposes to introduce a set of intelligent techniques aimed at vision-based automatic inspection of medium voltage OPDL, aiming at the components identification as well as the diagnosis of defects visible to the naked eye. For this purpose, different computational models were developed based on Digital Image Processing (DIP) and Artificial Intelligence (AI) techniques. As an additional part of this research, it was created an image database of OPDL components collected in a photographic studio and from a realistic OPDL created outdoors. To optimize the computational models performance, different types of intelligent algorithms and deep learning techniques were evaluated, with emphasis on Convolutional Neural Networks (CNNs), hybrid configurations and frameworks of Generative Adversarial

Networks (GANs). In addition, other approaches were implemented, such as: data augmentation; transfer learning and Multi-Task Learning (MTL); background invariance and Image Generators (IG). The preliminary results indicate satisfactory performance of the proposed techniques in class identification and defect detection of distribution insulators, providing a series of innovations in relation to other available solutions.

Agradecimentos

Confesso que a escrita dessa nota de agradecimento não é uma tarefa fácil, não somente pelo ato em si, mas pelo receio de não fazer justiça ou deixar de mencionar todas aquelas pessoas que contribuíram, direta ou indiretamente, nesta árdua e emocionante empreitada acadêmica – a obtenção do título de doutor em Engenharia Elétrica pela prestigiada **Universidade Federal da Bahia (UFBA)**.

Agradeço primeiramente aos meus pais, Sônia e Valternan, “pilares sólidos” na formação do meu caráter, que sempre me fizeram valorizar o trabalho honesto e esforço intelectual como um caminho virtuoso para conquistar minhas metas acadêmicas e profissionais. Querida mãe, afirmo sem medo de errar, que senhora é uma das protagonistas nesta e em outras realizações minhas. Também não posso deixar de agradecer ao meu irmão, Valternan Filho (Tuca), pelo apoio incondicional e a disposição incansável para me ouvir em todos os momentos dessa trajetória. Aos meus jovens e queridos irmãos (Rafael e Sofia) e ao meu afilhado Enzo, agradeço por todo o carinho dedicado e por serem uma fonte de inspiração ao longo da minha busca por aprimoramento pessoal. Aos meus grandes amigos e demais entes queridos, a exemplo Adriano, Joelzio (Joca), Manú, Sônia, Chicão etc. Muito obrigado por tudo!

Agradeço ao Prof. Eduardo Simas pela dedicação e cuidados voltados para minha orientação. O Senhor é um profissional exemplar e uma excelente pessoa. Obrigado pela confiança que depositou em mim e neste projeto! Agradeço também ao Prof. Rodrigo Ramos, meu coorientador, mentor e grande amigo. Sua valiosa colaboração foi um fator determinante para todas as realizações alcançadas nesta pesquisa. Aproveito o ensejo para externar meus agradecimentos aos demais membros do CENEL/UNIVASF, me sinto muito honrado em tê-los como colegas de trabalho! Sou muito grato também ao Prof. Jaime Cardoso e ao colega Ricardo Cruz, tendo em vista que os Senhores contribuíram significativamente tanto na minha orientação quanto no refinamento dos temas investigados nesta Tese. O INESC TEC/Porto foi um ambiente bastante prolífico e que me proporcionou fontes valiosas de aprendizado. Prof. Cardoso, muito obrigado por esta oportunidade!

Meu respeitoso agradecimento aos professores José Seixas, Marcelo Nascimento e Antônio Júnior, que gentilmente aceitaram os convites para participar como membros da banca examinadora da minha defesa de Tese de doutorado. A contribuição dos Senhores foi de grande valia para o aperfeiçoamento deste trabalho. E, com toda propriedade, dedico a minha mais profunda gratidão a todos os professores e técnicos do PPGEE, da UFBA. A todos os colegas de Pós-graduação gostaria de externar minha satisfação de poder conviver com vocês durante a realização deste Trabalho.

Agradecimento especial:

Dedico um espaço reservado nestes agradecimentos para prestar uma breve homenagem a uma pessoa cujo papel foi fundamental em todo este processo: minha belíssima esposa, **Renata de Carvalho Gomes Prates**. Sou muito grato por todo amor e cuidados dedicados, tanto em momentos alegres quanto nas adversidades. Você me ajudou a superar obstáculos que pensava ser intransponíveis. Te amo e espero estar ao seu lado nos próximos desafios!

A todos o meu muito obrigado!

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xx
1 Introdução	1
1.1 Objetivo	4
1.1.1 Objetivos Específicos	4
1.2 Metodologia	5
1.3 Organização do documento	7
2 Redes Primárias de Distribuição Aéreas (RPDA)	9
2.1 Isoladores de Distribuição	11
2.2 Inspeção Visual (IV)	13
2.3 Defeitos em Isoladores de Distribuição	14
2.4 Sistema Inteligente de Inspeção Visual (SIV)	16
2.4.1 Estado da Arte da Inspeção Automática	18
3 Sistemas Inteligentes de Classificação	20
3.1 Problemas de Decisão entre Hipóteses	22
3.1.1 Decisão entre Múltiplas Hipóteses	25
3.2 Algoritmos de Classificação	26
3.2.1 Rede Neural Artificial (RNA)	27
3.2.2 <i>k-Nearest Neighbors</i> (<i>k</i> NN)	41
3.2.3 Máquinas de Vetores de Suporte (SVM)	42
3.2.4 Floresta Aleatória (<i>Random Forest</i>)	45
3.2.5 Rede Bayesiana (<i>Bayes Net</i>)	46

3.2.6	Modelo Oculto de Markov (HMM)	48
3.3	Processamento Digital de Imagens (PDI)	49
3.3.1	Tarefas Fundamentais em PDI	51
3.4	Aprendizagem Profunda	54
3.5	Rede Neural Convolutacional (RNC)	58
3.5.1	A operação de convolução	61
3.6	Arquitetura da Rede Neural Convolutacional	63
3.6.1	Camada Convolutacional	63
3.6.2	Camada de Agrupamento (<i>Pooling</i>)	66
3.6.3	Camada Densa	68
3.6.4	Camada de sobreamostragem (<i>Up-sampling</i>)	69
3.6.5	Funções de Ativação	71
3.7	Estratégias de treinamento	74
3.7.1	Penalidades na Função Objetivo	75
3.7.2	Técnicas de <i>Data Augmentation</i>	78
3.7.3	Transferência de Conhecimento	79
3.7.4	<i>Dropout</i>	80
3.8	Aplicações das Redes Neurais Convolutacionais	81
3.8.1	Classificação de Imagens	82
3.8.2	Segmentação	84
3.8.3	Identificação de Objetos	88
3.9	<i>Multi-Task Learning</i> (MTL)	89
3.10	<i>Multi-View</i>	89
3.11	Redes Generativas Adversárias (GANs)	90
3.11.1	Redes Generativas Adversárias Condicionais (cGANs)	94
4	Definição do Problema e Metodologia	97
4.1	Aquisição de Imagens	100
4.1.1	Estação de Coleta de Imagens (ECI)	102
4.1.2	Rede de Distribuição de Ensaios (RDE)	104
4.2	Gerador de Imagens Artificiais (GIA)	107
4.2.1	Segmentação Semântica com emprego de GIA	111
4.3	Aplicações em RPDA	113

4.3.1	Classificação por Atributos Geométricos	114
4.3.2	Classificação por Técnicas Híbridas	119
4.3.3	Diagnóstico de defeitos por aprendizagem profunda	122
4.4	Aplicações com Redes Generativas Adversárias	127
4.4.1	Geração de <i>dataset</i> Artificial com cGANs	129
4.4.2	Técnica de <i>Background Invariance</i>	132
4.5	Critérios de Avaliação de Desempenho	137
4.5.1	Desempenho de Classificação	137
4.5.2	Desempenho Estatístico de Geradores de Imagens	140
4.6	Técnica de Validação Cruzada	142
5	Resultados e Discussões	145
5.1	Gerador de Imagens Artificiais (GIA)	145
5.1.1	Avaliação Estatística dos <i>datasets</i> de Imagens	148
5.1.2	Segmentação Semântica de Imagens	149
5.2	Classificação por Atributos Geométricos	151
5.3	Classificação por Técnicas Híbridas	157
5.4	Diagnóstico de Defeitos por Aprendizagem Profunda	161
5.5	Geração de <i>dataset</i> Artificial com cGANs	168
5.6	Técnica de <i>Background Invariance</i>	173
6	Considerações Finais	177
6.1	Perspectivas para Trabalhos Futuros	180
	Referências Bibliográficas	181
A	Trabalhos Publicados	203
A.1	Artigos aprovados em congressos	203
A.2	Artigos publicados em revista	207
B	Outras Informações Relevantes	209
B.1	Estágio em Instituição de Ensino Estrangeira	209
B.2	Desenvolvimento de <i>dataset</i>	209

Lista de Figuras

1.1	Exemplos de tipos de isoladores de redes de distribuição.	2
1.2	Exemplos de defeitos presentes em isoladores de redes de distribuição	2
2.1	Estágios pertencentes à infraestrutura de fornecimento de energia elétrica	10
2.2	Diagrama unifilar de um alimentador típico (a) e estrutura de distribuição de 13,8 kV com transformador instalado (b).	10
2.3	Imagens de Isoladores de distribuição coletadas em diferentes ângulos de θ (inclinação vertical)	11
2.4	Materiais conectados aos isoladores de distribuição em RPDA.	12
2.5	Imagens de componentes quebrados.	14
2.6	Imagens de componentes cerâmicos rachado e fraturado.	15
2.7	Componentes deformados (A e B) e com alterações de cor (C e D).	15
2.8	Inconformidades relacionadas à ausência de componentes.	16
2.9	Ninhos de passarinho localizados próximos aos isoladores.	16
2.10	Ilustração de uma forma automatizada para inspeção de RPDA através do uso de <i>Drones</i>	17
3.1	Fluxo de operação de um sistema de classificação.	20
3.2	Problema de classificação entre duas hipóteses (decisão binária).	23
3.3	Problemas de decisão entre múltiplas hipóteses (decisão p -ária).	25
3.4	Modelo de neurônio artificial e respectivas conexões na RNA.	27
3.5	Planta arquitetural de Rede Neural Artificial (RNA) do tipo <i>Multi-Layer Perceptron</i> com duas camadas ocultas.	28
3.6	Diagrama de fluxo de atualização da retro-propagação erro de uma RNA do tipo MLP com duas camadas ocultas.	36

3.7	Exemplo de uso do modelo k NN para definição da classe pertencente à observação \mathbf{i} , de acordo com os K padrões mais próximos (para k iguais a 1, 5 ou 14).	41
3.8	Ilustração de um hiperplano bidimensional ótimo para valores linearmente separáveis.	42
3.9	Regiões de decisão do algoritmo SVM para diferentes tipos de <i>Kernels</i>	44
3.10	Representação simplificada de um conjunto de árvores de decisão que constituem o classificador <i>Random Forest</i>	45
3.11	Exemplo simplificado de grafo acíclico dirigido (DAG) de uma rede bayesiana.	47
3.12	Representação genérica do Modelo Ocultos de Markov (HMMs).	48
3.13	Tarefas fundamentais pertencentes a área de Processamento Digital de Imagens (PDI). Adaptado de [113].	50
3.14	Ciclo de Gartner para Inteligência Artificial, 2019. Adaptado de [122].	55
3.15	Ciclo de Gartner para Inteligência Artificial, 2020. Adaptado de [123].	55
3.16	Arquitetura típica de uma Rede Neural Convolutiva (RNC).	59
3.17	Exemplo de produto escalar entre filtros receptivos ($2 \times 2 \times 3$) e região do tensor de entrada de dimensões $4 \times 4 \times 3$	64
3.18	Filtros receptivos (<i>Kernel</i>) da rede neural convolutiva.	65
3.19	Exemplo de um conjunto de mapas de ativação (tensor $224 \times 224 \times 16$) gerados por um estágio convolutivo, com 16 <i>kernels</i> do tipo $3 \times 3 \times 3$	66
3.20	Camadas de subamostragem - <i>max pooling</i> e <i>average pooling</i>	67
3.21	Representação das camadas achatadas da RNC.	69
3.22	Operação de Convolação Transposta (CT) entre uma matriz de dimensões 2×2 e um <i>kernel</i> do tipo 2×2	70
3.23	Exemplo de máscara de predição, para imagem de entrada, gerada por um algoritmo de aprendizagem profunda composto por camadas de sobreamostragem	70
3.24	Gráficos gerados pelas principais funções de ativação implementadas em algoritmos de aprendizagem profunda.	74
3.25	Imagens geradas por <i>data augmentation</i>	79
3.26	Representação das rotinas de <i>dropout</i>	81

3.27	Planta arquitetural da RNC VGG-19.	82
3.28	Representação simplificada da ResNet-50.	83
3.29	Representação simplificada dos módulos <i>inception</i>	84
3.30	Exemplos de respostas produzidas por algoritmos de segmentação. . .	85
3.31	Exemplo de segmentação realizada com o algoritmo Mask R-CNN. . .	85
3.32	Exemplo de algoritmo de segmentação com configuração inspirada na U-Net.	86
3.33	Exemplificação de imagem com regiões demarcadas por uma RNC desenvolvida para identificação de objetos.	88
3.34	RNC empregada para operação de <i>Multi-Tasking Learning</i> (MTL). . .	89
3.35	RNC empregada para configuração de <i>Multi-View</i>	90
3.36	Diagrama de funcionamento das <i>redes generativas adversárias</i> (GANs). 91	
3.37	Comportamento de $\mathcal{L}_B(y, \hat{p})$ em função do tipo de variável de entrada do <i>modelo discriminador</i> (D).	93
3.38	Representação simplificada de um arranjo de <i>redes generativas adver-</i> <i>sárias condicionais</i> (cGANs).	95
4.1	Fluxograma macro de atividades desenvolvidas para a pesquisa de Doutorado.	97
4.2	Diagrama de atividades desenvolvidas para a etapa experimental. . .	99
4.3	Estação de Coleta de Imagens (ECI).	103
4.4	<i>Layout</i> da estação de coleta de imagens.	103
4.5	Exemplos de tipos de inconformidades coletadas na ECI.	103
4.6	Exemplos de componentes defeituosos coletados na ECI: (a) ICP; (b) IPC; (c) ICB; (d) IVV.	104
4.7	Fotografia (vista superior) da Rede de Distribuição de Ensaio (RDE). 105	
4.8	Fotografia do poste central da RDE.	105
4.9	Esquemático da Rede de Distribuição de Ensaio (RDE).	105
4.10	Imagens de Isoladores íntegros (a, b, c e d) e defeituosos (e, f, g e h) coletadas em diferentes ângulos.	106
4.11	Diagrama de fluxo da informação do Gerador de Imagens Artificiais (GIA).	107

4.12	Imagem do dataset ECI (parte superior) e máscaras produzidas com a técnica <i>Graph Cut</i> (parte inferior).	108
4.13	Exemplos de imagens de amarrações metálicas e condutores livres segmentadas pela técnica <i>Graph Cut</i>	108
4.14	Imagem de cenário alterada pela técnica de Inpainting.	110
4.15	Algoritmo de segmentação semântica U-Net adaptado.	111
4.16	Diagrama da metodologia proposta de segmentação semântica através emprego do Gerador Imagens Artificiais (GIA).	112
4.17	Exemplos de três pares de treinamento gerados para o algoritmo de segmentação semântica: imagem de entrada (\mathbf{X}_{im} - imagem à esquerda) e máscara de saída para o <i>label</i> (\mathbf{Y}_{im} - imagem à direita). . .	113
4.18	Diagrama da metodologia proposta.	115
4.19	Atributos geométricos da imagem (máscara) de isolador de distribuição do tipo ICP.	118
4.20	Arranjo dos 3 Modelos Ocultos de Markov (HMMs) empregados para classificação de parâmetros geométricos de isoladores de distribuição.	119
4.21	Diagrama da metodologia proposta.	120
4.22	Arquitetura simplificada da ResNet-50 (a) e configuração dos classificadores híbridos (b).	121
4.23	Diagrama da metodologia proposta para classificação de isoladores e defeitos.	123
4.24	Arquitetura da RNC VGG-19 com a configuração <i>multi-task learning</i>	126
4.25	Exemplo de mapa de calor produzido para uma imagem de entrada.	126
4.26	Arranjo das arquiteturas das redes convolutivas adversárias condicionais (cGANs) propostas e quadro de legendas referente aos blocos de processamento da informação.	130
4.27	Diagrama de fluxo da informação adotado para o desenvolvimento da proposta de geração de planos de fundos adversários durante o estágio de treinamento.	133
4.28	<i>Datasets</i> de isoladores de distribuição empregados para as etapas de treino e de testes da metodologia proposta	135

4.29	Planos de fundos introduzidos nos <i>datasets</i> (1) MNIST e (2) Fashion-MNIST: (a) original, (b) listras, (c) tabuleiro, (d) quadro, (e) círculos, (f) relógio e (g) ruído aleatório.	136
4.30	Exemplo de matriz de confusão para três classes distintas.	139
4.31	Exemplo de curvas ROC sobrepostas para três classes distintas. . . .	140
4.32	Técnica de Validação Cruzada (VC) convencional.	143
4.33	Técnica de Validação Cruzada Adaptada (VCA).	143
5.1	Diagrama processamento da informação do Gerador de Imagens Artificiais (GIA).	146
5.2	Imagens geradas com ativação dos estágios C e A.	147
5.3	Imagens geradas com ativação dos estágios A, C e D.	147
5.4	Imagens geradas com ativação dos estágios A, C, D e E.	147
5.5	Exemplos de imagens do <i>dataset</i> de teste após o estágio de <i>Stress test</i>	148
5.6	Exemplos de imagens de componentes íntegros segmentadas pela técnica do limiarização múltipla de Otsu.	152
5.7	Exemplos de imagens de componentes íntegros segmentadas pelo algoritmo adaptado da Rede Neural Convolutacional (RNC) U-Net. . . .	152
5.8	Exemplos de imagens de componentes defeituosos segmentadas pela técnica de limiarização múltipla de Otsu.	152
5.9	Exemplos de imagens de componentes defeituosos segmentadas pelo algoritmo adaptado da U-Net.	152
5.10	(a) Exemplos de vetores pertencentes ao <i>dataset</i> de entrada; (b) valor da acurácia média dos testes em relação ao número de estados observáveis do arranjo de HMMs.	154
5.11	Acurácia geral de acordo com o tipo de modelo – dados do <i>dataset</i> coletados na ECI.	155
5.12	MC do modelo HMM para classificação do tipo de isolador.	155
5.13	Imagens de isoladores (posicionados acima) e respectivas máscaras produzidas pelo algoritmo de segmentação U-Net treinado com o Gerador de Imagens Artificiais (GIA).	156
5.14	Acurácia geral da HMM para os <i>datasets</i> íntegros e defeituosos coletados na RDE.	157

5.15	Parâmetros de desempenho da ResNet-50 com camada de saída softmax.	158
5.16	Acurácia de classificação das técnicas propostas para o <i>dataset</i> coletado na ECI - valores em %.	159
5.17	MCs da ResNet-50 para DATA-DEF.	160
5.18	Exemplos de erros de classificação para VGG-19 + SVM.	160
5.19	Acurácia de classificação das técnicas propostas para o <i>dataset</i> coletado na RDE - valores em %.	160
5.20	Gráficos Boxplots para o desempenho de classificação da CNN VGG-19.	162
5.21	Acurácia dos classificadores de acordo com os estágios de treinamento: I) treinamento convencional; II) Inclusão de rotina de <i>data augmentation</i> (A); III) segmentação das imagens do dataset de treinamento (C); IV) ajuste das camadas $C_{(n-1)}$ e $C(n)$ (B); V) inclusão dos estágios A e B; VI) inclusão dos estágios A, B e C; VII) adição de planos de fundo (D) e adição dos estágios A, B e C; VIII) inclusão de componentes menores (E) e adição dos estágios A, B, C e E; IX) adição dos estágios A, B e C e adoção da modalidade de treino <i>multi-task learning</i> (F); X) adição dos estágios A, B, C, D e F; XI) adição dos estágios A, B, C, D, E e F.	163
5.22	Curvas ROC para a VGG-19 do Tipo 1 de acordo com as especificações de treinamento.	164
5.23	MCOs + <i>max-pooling</i> obtidos sem a adição de planos de fundo.	166
5.24	MCOs + <i>max-pooling</i> com a adição de planos de fundo.	166
5.25	MCOs + <i>average-pooling</i> obtidos sem a adição de planos de fundo.	166
5.26	MCOs + <i>average-pooling</i> obtidos com a adição de planos de fundo.	166
5.27	MCs da VGG-19 (Tipo 1) para classificação do tipo de isolador.	167
5.28	MC da VGG-19 (Tipo 3) para classificação de inconformidades.	167
5.29	Exemplos de erros de classificação de isoladores.	168
5.30	Exemplos de imagens geradas pelas cGANs para diferentes modalidades de treinamento.	169

5.31	Exemplos de imagens segmentadas do <i>dataset</i> real de IVV (posicionadas à esquerda) e amostras falsas geradas pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$; a) imagens de isoladores íntegros e b) exemplos de componentes defeituosos.	172
5.32	Exemplos de imagens de ICP íntegros e defeituosos produzidos pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$; componentes íntegros posicionados acima e defeituosos na parte inferior.	172
5.33	Exemplos de geração de planos de fundo dinâmicos no decorrer das épocas (t).	175
5.34	Exemplos de erros de classificação realizados pelo classificador treinado de forma convencional, que não ocorrem no treino de forma de adversária.	175
5.35	Mapas de Calor (MCO) para a forma de treinamento convencional.	176
5.36	Mapas de Calor (MCO) para a forma de treinamento através da técnica de <i>background invariance</i> (Back-Inv).	176

Lista de Tabelas

4.1	Tabela de Imagens Coletadas para os isoladores de distribuição íntegros (INT) e defeituosos (DEF).	106
5.1	Valores de IS e FID para os <i>datasets</i> apresentados.	149
5.2	Coefficientes <i>Dice</i> de segmentação das técnicas propostas - valores em %.150	
5.3	Estrutura de dados empregada no treinamento dos modelos.	153
5.4	Acurácias dos modelos para classificação do tipo de isolador (valores em%).	154
5.5	Precisão das classes para DATA-DEF.	159
5.6	Valores médios obtidos com a técnica de Validação Cruzada Adaptada (VCA) – acurácia em %.	162
5.7	Valores de IS e FID para os <i>datasets</i> de imagens produzidos pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$, considerando os <i>datasets</i> íntegros (INT) e defeituosos (DEF) da base de dados real de IVV.	171
5.8	Tabela das Acc de classificação do <i>dataset</i> MNIST: (A) listras; (B) tabuleiro; (C) borda; (D) círculos; (E) relógio e (F) aleatório - valores em %.	174
5.9	Tabela das Acc de classificação do <i>dataset</i> Fashion MNIST: (A) listras; (B) tabuleiro; (C) borda; (D) círculos; (E) relógio e (F) aleatório - valores em %.	174
5.10	Acurácias de identificação de material/ <i>design</i> do componente isolador de acordo com o sistema de classificação adotado – imagens pertencentes ao <i>dataset</i> coletado na RDE (valores em %).	176

Símbolos e Abreviaturas

Símbolos

$p_x(x)$	Função de densidade de probabilidade
θ	Ângulo polar
ϕ	Ângulo azimutal
E_{f_i}	Eficiência de discriminação da classe i
$f_a(\cdot)$	Função <i>average pooling</i>
$f_m(\cdot)$	Função <i>max pooling</i>
$h(\cdot)$	Função de convolução para valores contínuos
$h[.]$	Função de convolução para valores discretos
$\varphi(\cdot)$	Função de ativação não linear
$\mathcal{L}(\cdot)$	Função Custo
X_i	Sequência Oculta de Estados da HMM
A	Matriz de Probabilidade de Transição de Estados
B	Matriz de Probabilidade das Observações
O	Sequência de Observações
$\pi[.]$	Distribuição de probabilidades inicial dos estado
$\lambda(\cdot)$	Conjunto de parâmetros do Modelo Oculto de Markov (HMM)
γ_l	Limiar de decisão de classificadores
η	Taxa de aprendizagem em redes neurais
\mathbf{G}_a	Vetor de atributos geométricos
\mathbf{G}_{Qa}	Vetor de Parâmetros Quantizado
$h_{x,y}$	Matriz do mapa de calor

Abreviaturas

BN	<i>Bayes Net</i> / Rede Bayesiana
CA	Camada de Agrupamento
CD	Camada Densa ou Completamente Conectada
CT	Convolução transposta
D	Modelo Discriminador das GANs
DSC	<i>Dice Segmentation Coefficient</i> / Coeficiente de Dice
ECI	Estação de Coleta de Imagens
FID	<i>Fréchet Inception Distance</i>
G	Modelo Gerador das GANs
GANs	<i>Generative Adversarial Networks</i> / Redes Generativas Adversárias
GIA	Gerador de Imagens Artificiais
GPU	<i>Graphics Processing Units</i> / Unidade Gráfica de Processamento
HMM	<i>Hidden Markov Model</i> / Método Oculto de Markov
IA	Inteligência Artificial
ICB	Isolador Cerâmico Bicolor
ICP	Isolador Cerâmico de Pino
IPC	Isolador Polimérico Cinza
IPH	Isolador Pilar Híbrido
IVT	Isolador de Vidro Transparente
IVV	Isolador de Vidro Verde
IS	<i>Inception Score</i>
IV	Inspeção Visual
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
k -NN	<i>k-nearest neighbors</i> / k -Vizinhos mais Próximos
MA	Mapa de Ativação ou Mapa de Características
MC	Matriz de Confusão
MPL	<i>Multilayer Perceptron</i> / Perceptron de Múltiplas Camadas
MTL	<i>Multi-Task Learning</i> / Múltiplas Tarefas de Saída
PDI	Processamento Digital de Imagem
RPDA	Redes Primárias de Distribuição Aéreas
RDE	Rede de Distribuição de Ensaio

RF	<i>Random Forest</i> / Floresta Aleatória
RNA	Rede Neural Artificial
RNC	Redes Neural Convolutacional
RNR	Redes Neural Recorrente
ROC	<i>Receiver operating characteristic</i>
SIV	Sistema Inteligente de Inspeção Visual
SGD	<i>Stochastic Gradient Descent</i> / Descida Estocástica de Gradiente
SVM	<i>Support Vector Machine</i> / Máquina de Vetor de Suporte
VC	Validação Cruzada
VCA	Validação Cruzada Adaptada
VANT	Veículo Aéreo Não Tripulado

Capítulo 1

Introdução

No Brasil, as Redes Primárias de Distribuição Aéreas de Energia Elétrica (RPDA) correspondem a um percentual superior a 95% do total de circuitos elétricos de distribuição em média tensão, estando presentes tanto em zonas urbanas quanto rurais [1]. Outros países, tais como Estados Unidos e Japão, também adotam essa topologia de rede, dividindo espaço com as redes de distribuição do tipo subterrânea [2, 3].

No que diz respeito à conformidade de tensão de fornecimento de energia elétrica, é correto dizer que este parâmetro está intimamente relacionado com a qualidade dos serviços de manutenção praticados pelas concessionárias de energia elétrica nas redes de distribuição de baixa e média tensão. Como pré-requisito para realização das referidas atividades, a primeira etapa a ser realizada é o serviço de Inspeção Visual (IV). A IV consiste na identificação de componentes e diagnóstico de inconformidades, visíveis a olho nu, presentes em RPDA, como, por exemplo: elementos defeituosos (quebrados, fraturados, deformados e poluídos), mal posicionados, componentes ausentes ou não cadastrados, fugas de corrente, pontos quentes, dentre outros [4].

A partir da determinação do tipo de componente e respectivo defeito, torna-se possível a atuação de forma preventiva nas redes de distribuição. Essas atividades proporcionam melhoria na qualidade do produto (tensão elétrica em regime permanente) e reduzem a probabilidade de ocorrência de interrupções no fornecimento de energia [5].

Atualmente, as atividades de inspeção visual em campo nas redes aéreas de média e alta tensão são, em parte, realizadas por profissionais treinados pela concessionária,

na modalidade de inspeção por “patrulha a pé” ou assistida por helicóptero [6, 7]. As referidas atividades são baseadas em métodos empíricos de avaliação, onde estes profissionais atuam em proximidade das RPDA e apontam as classes de componentes e a presença de defeitos de forma visual e intuitiva. Na maior parte dos casos, as inspeções visuais são realizados com os circuitos energizados, o que pode trazer riscos à segurança destes profissionais [8, 9].

Os procedimentos de inspeção visual também estão sujeitos a subjetividades e interpretações heurísticas, que podem levar a identificação/diagnósticos imprecisos ou incorretos, resultando em falhas de cadastros e planejamento inadequado das ações de manutenção. Outro ponto que dificulta a realização da IV é a grande diversidade de padrões de construção, design/material dos componentes e tipos de inconformidades. Como um dos principais componentes presentes nas RPDA, os isoladores de distribuição [10], por exemplo, possuem grande diversidade em termos de formato e tipo de material construtivo, além de uma ampla variedade de tipos de defeitos conhecidos (ver Figuras 1.1 e 1.2). Essas características elevam consideravelmente a complexidade das atividades de inspeção visual e reduzem o nível de acurácia das informações coletadas pelos inspetores de RPDA.



Figura 1.1: Exemplos de tipos de isoladores de redes de distribuição.

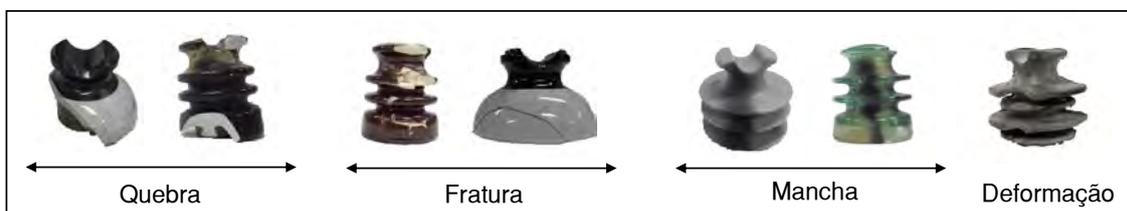


Figura 1.2: Exemplos de defeitos presentes em isoladores de redes de distribuição

Dessa forma, verifica-se que ainda é latente a necessidade de melhoria das técnicas de inspeção visual de defeitos presentes nas redes aéreas de distribuição de energia elétrica. Para isso, uma abordagem se mostra promissora: desenvolvimento e implementação de técnicas automatizadas/inteligentes de aquisição de dados e

análise das informações. De forma mais específica, a referida abordagem pode ser explorada para composição de sistemas inteligentes direcionados à identificação de componentes e ao diagnóstico de defeitos visíveis em RPDA apresentando o mínimo possível de intervenção humana.

Os sistemas inteligentes e as técnicas de visão computacional podem ser utilizados para redução de subjetividades inerentes aos atuais métodos de inspeção visual. Essas aplicações vêm sendo estudadas há mais de 30 anos, mas a sua popularização e utilização em larga escala tiveram início recentemente, com a redução nos custos para aquisição de sistemas digitais de captura de imagens, como filmadoras e *webcams* [11]. Diversas metodologias foram testadas com sucesso para otimização de tarefas nas mais diversas áreas, tais como: detecção de defeitos em processos industriais [12, 13], sistemas médicos [14] e na indústria agrícola [15, 16]. Como perspectiva adicional, foi observado que o emprego de visão computacional e técnicas inteligentes para inspeção de RPDA é um assunto atual e, até onde sabemos, pouco explorado. Revisões recentes da literatura a respeito deste tema podem ser encontradas nas referências [17, 18].

Outro fator importante e que contribuiu para a evolução e abrangência do campo da visão computacional, foi a implementação de técnicas baseadas em aprendizagem profunda (do inglês, *deep learning*), em especial, as Redes Neurais Convolucionais (RNC), as Redes Neurais Recorrentes (RNR) [19, 20] e as redes generativas adversárias (do inglês, *Generative Adversarial Networks* - GANs) [21]. As RNC, por exemplo, apresentaram recentemente ganhos significativos nas atividades de classificação de imagens [22], segmentação semântica e instanciada [23, 24, 25] e na identificação de objetos em imagens e vídeos em tempo real [26, 27]. A aprendizagem profunda também está presente em sistemas visuais avançados, tais como: carros autônomos [28] e algoritmos de interpretação de imagens [29, 30]. Estes modelos computacionais possuem uma ampla faixa de generalização entre tarefas e podem ser facilmente ajustados para novas finalidades através de ajustes em hiperparâmetros e alterações nas suas técnicas de treinamento. Tais benefícios proporcionam oportunidades para o emprego da aprendizagem profunda em tarefas de inspeção automatizadas/inteligentes do tipo visual [31, 32, 33, 34].

Diante do que foi exposto, esta pesquisa de doutorado tem como objetivo realizar

o desenvolvimento de técnicas inteligentes direcionadas a identificação de componentes e diagnóstico de inconformidades presentes em RPDA, através do emprego de Processamento Digital de Imagem (PDI) e inteligência computacional. Com este propósito, serão exploradas diferentes arquiteturas de algoritmos inteligentes, com foco em tarefas de classificação, segmentação semântica, geração de imagens artificiais e redução da sensibilidade de modelos computacionais a fontes de macro interferências de plano de fundo – abordagem conhecida como *background invariance* [35]. No melhor dos cenários, as técnicas propostas podem contribuir para o desenvolvimento de aplicações inteligentes para composição de sistemas de inspeção visual de redes primárias de distribuição aéreas, gerando dados, tecnologias e perspectivas para futuros trabalhos acadêmicos voltados à área de inspeção automatizada.

1.1 Objetivo

Desenvolvimento de aplicações inteligentes destinadas à classificação de componentes isoladores e diagnóstico de inconformidades presentes em Redes Primárias de Distribuição Aéreas de Energia Elétrica (RPDA), por meio de técnicas de Processamento Digital de Imagem (PDI) e de inteligência computacional.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral, este trabalho apresentou os seguintes objetivos específicos:

- 1 Coletar base de dados representativa do problema;
- 2 Desenvolver algoritmos inteligentes direcionados à classificação de componentes isoladores de RPDA, a partir da extração de atributos geométricos.
- 3 Desenvolver técnicas de aprendizagem profunda e configurações híbridas (*ensemble*) com foco na otimização do processo de classificação do material/*design* de isoladores defeituosos;
- 4 Avaliar o desempenho dos classificadores propostos em *datasets* de componentes isoladores íntegros e defeituosos;

- 5 Desenvolver um Gerador de Imagens Artificiais (GIA) direcionado para geração de *datasets* de treino com características desejadas de ambientes realistas;
- 6 Aplicar as metodologias desenvolvidas na classificação conjunta do tipo de componente e respectivo *status* de conformidade de isoladores de distribuição situados em redes aéreas;
- 7 Propor técnica para geração de *datasets* de imagens artificiais de componentes isoladores (íntegros e defeituosos) através de redes generativas adversárias condicionais (cGANs);
- 8 Desenvolver técnica de *background invariance* direcionada à redução de sensibilidade dos modelos computacionais a macro interferências de planos de fundo.

1.2 Metodologia

Para a condução do trabalho de doutorado, inicialmente foi necessário desenvolver um conjunto de *datasets* originais relacionados com o tema desta pesquisa. As referidas base de dados dizem respeito a imagens de componentes e estruturas de RPDA de 13,8 kV [36, 37, 38], tais como: isoladores de distribuição com diferentes formatos e tipos de materiais construtivos (cerâmico, polimérico, vítreo e híbrido); componentes íntegros e defeituosos; elementos complementares de RPDA (amarração metálica de isoladores, pinos de sustentação, condutores elétricos, dentre outros) e diferentes estruturas de sustentação de redes de distribuição aéreas.

Os procedimentos de coleta de imagens ocorreram em dois ambientes distintos. Para o primeiro ambiente, foi utilizado um estúdio fotográfico situado na UFBA, intitulado de ECI (Estação de Coleta de Imagens), desenvolvido pelo aluno de doutorado (PPGEE/UFBA) André P. Marotta. No segundo caso, a aquisição de imagens ocorreu em uma Rede de Distribuição Aérea de Ensaio (RDE), não-energizada, localizada em ambiente externo, implantada nas dependências da Universidade Federal do Vale do São Francisco (UNIVASF). Os procedimentos de coleta foram concebidos com o propósito de viabilizar a aquisição de imagens/vídeos de forma controlada e segura. Objetivou-se também que os referidos *datasets* conseguissem reproduzir con-

dições realistas inerentes às IV atualmente praticadas por profissionais envolvidos nas atividades de inspeção.

Este trabalho pode ser dividido em quatro propostas correlatas. A primeira delas consiste na classificação de componentes isoladores através de técnicas de extração de atributos geométricos e da implementação de algoritmos inteligentes clássicos, tais como: Redes Neurais Artificiais (RNA), Máquina de Vetor de Suporte (*Support Vector Machine* - SVM), *Random Forest*, *k-nearest neighbors* (*k*-NN) e Modelo Oculto de Markov (*Hidden Markov Model* - HMM) [39, 40, 41, 42].

A segunda proposta está relacionada ao desenvolvimento de algoritmos de aprendizagem profunda, a exemplo das redes neurais convolucionais, para classificação do material/*design* de componentes íntegros e defeituosos em ambientes controlados e estruturas reais. Para esta etapa, fez-se necessário o emprego de configurações híbridas e de técnicas de transferência de conhecimento com foco na otimização de desempenho dos modelos.

Para o terceiro caso, foi avaliado o desempenho das RNC em tarefas individualizadas e em configurações de aprendizado com múltiplas saídas (*Multi-Task Learning* - MTL). Através da combinação de funções custo na etapa de treinamento, foi estudado se a relação intrínseca entre as tarefas possui o potencial de promover melhorias no processo de classificação como um todo. Dentro do escopo desta pesquisa, objetivou-se classificar o tipo de componente e o seu respectivo *status* de conformidade – identificando o isolador como íntegro ou defeituoso. Como ideia complementar, foi desenvolvido um Gerador de Imagens Artificiais (GIA) direcionado à criação dinâmica de datasets de treinamento a partir da inclusão artificial de detalhes realísticos. O objetivo principal desta etapa foi desenvolver algoritmos inteligentes direcionados à classificação de componentes e defeitos em imagens de RPDA, provenientes de ambientes realistas, com base nos *datasets* de treinamento coletados na ECI.

Para o quarto e último caso, foram explorados diferentes arranjos de redes generativas adversárias para o desenvolvimento de aplicações direcionadas a potenciais demandas de sistemas inteligentes de inspeção visual, a exemplo da geração de *datasets* diversificados de componentes defeituosos e da redução do nível de sensibilidade dos classificadores de imagens a interferências de plano de fundo. A partir das refe-

ridas demandas, foram propostas estratégias para geração de *datasets* de imagens artificiais de componentes isoladores, como também foi desenvolvida uma técnica de *background invariance* direcionada à redução da sensibilidade dos modelos computacionais a macro interferências de plano de fundo presentes em imagens realistas de RPDA.

Na etapa de análise dos resultados, foram comparados diversos parâmetros de desempenho dos classificadores, a exemplo da acurácia e *F1-score*, levando-se em consideração as condições de balanceamento dos *datasets*. Outro ponto também investigado foi o nível de qualidade e a diversidade de *datasets* produzidos pelas metodologias propostas de geração de imagens. Com este propósito, foram utilizadas as técnicas *Inception Score* (IS) e *Fréchet Inception distance* (FID) – estes parâmetros serão definidos no Capítulo 4. Com foco no aprofundamento das análises, foram também produzidas matrizes de confusão (MCs) [43] e curvas ROC [44] nos estágios de treinamento e teste dos classificadores. Tais ferramentas permitiram avaliar a relação de custos e benefícios apresentados na classificação, como também demonstraram quais classes enfrentam maiores dificuldades para serem preditas.

No que tange à avaliação visual das simulações realizadas, foram gerados mapas de calor (*heat-maps*) e máscaras de predição, viabilizando também a comparação de desempenho entre os algoritmos de segmentação trabalhados nesta pesquisa. A fim de tornar os *datasets* de testes mais desafiadores, foram aplicadas também técnicas de ampliação de dados (*data augmentation*) para realização de um estágio de “teste de estresse” da base de dados avaliada - técnica conhecida como *stress test*. Os resultados obtidos foram então comparados com a literatura especializada que trata do tema de inspeção inteligente de redes de distribuição aéreas.

1.3 Organização do documento

Esta Tese de Doutorado está organizada do modo descrito a seguir:

- Para uma melhor organização do documento, a Revisão Bibliográfica foi dividida em dois temas principais, apontados nos Capítulos 2 e 3.
- Para o Capítulo 2, será realizada uma apresentação geral das Redes Primárias de Distribuição Aéreas de Energia Elétrica (RPDA), bem como das atividades

de inspeção visual e potenciais sistemas inteligentes de inspeção de RPDA.

- No Capítulo 3 será apresentada a fundamentação teórica relacionada aos conceitos fundamentais de sistemas inteligentes de classificação com ênfase nos problemas de identificação entre múltiplas hipóteses. Também serão abordadas as principais características de algoritmos de aprendizagem profunda direcionados à área de visão computacional e suas potenciais aplicações.
- No Capítulo 4 serão apresentados os procedimentos de coleta de *datasets* de imagens realizados nesta pesquisa, e também serão descritas as principais metodologias e técnicas propostas, dando ênfase aos geradores de imagens artificiais, aplicações inteligentes em RPDA e demandas de investigação relacionadas à redes generativas adversárias.
- No Capítulo 5 serão avaliados os resultados experimentais obtidos na pesquisa de Doutorado.
- Conclusões sobre os resultados obtidos na Tese de Doutorado e considerações a respeito de potenciais trabalhos futuros serão apresentados no Capítulo 6.
- No Apêndice A serão apresentados os trabalhos publicados ao longo desta Tese. Finalmente, o Apêndice B apresenta algumas atividades complementares relacionadas ao trabalho de doutorado.

Capítulo 2

Redes Primárias de Distribuição Aéreas (RPDA)

As *Redes Primárias de Distribuição Aéreas de Energia Elétrica (RPDA)*, também conhecidas como *circuitos de distribuição primários*, fazem parte da infraestrutura de fornecimento de energia elétrica e estão interligadas às usinas geradoras através dos estágios de *transmissão* e *sub-transmissão* [45]. Essas redes são constituídas por circuitos elétricos de média tensão, normalmente estabelecidos entre a faixa de 4 até 36,2 kV [38]. A subestação de distribuição é o ponto de origem da RPDA, onde uma série de circuitos terminais, intitulados de *alimentadores*, se dividem para o suprimento de consumidores de pequeno, médio e grande porte [2, 46]. Perto do usuário final, um novo transformador converte a média tensão de distribuição para um nível secundário, em baixa tensão, geralmente especificados entre 120 e 240 V [2].

A Figura 2.1 ilustra, de forma simplificada, os diversos estágios pertencentes à infraestrutura de fornecimento de energia elétrica. Dentre os referidos estágios, esta imagem dá ênfase aos circuitos de distribuição primários, correspondentes às redes de distribuição aéreas que possuem níveis de tensão na faixa de 11,9 até 15 kV – topologia de RPDA que foi mais explorada nesta pesquisa de doutorado [31, 33, 34, 35]. Esta figura também mostra uma estrutura de distribuição – composta por poste, cruzeta e isoladores –, típica de redes aéreas projetadas para níveis de tensão de até 15 kV [47, 48].

Dando enfoque às topologias típicas dos sistemas elétricos de distribuição, pode-se afirmar que as RPDA são usualmente extensas e bastante ramificadas. Elas

podem se diferir em carga instalada, número de fases e comprimento – variando de apenas alguns metros até centenas de quilômetros. Quanto às modalidades de fornecimento, é comum existir diferentes configurações no mesmo alimentador – trechos monofásicos, bifásicos e trifásicos podem operar simultaneamente no mesmo circuito. As referidas configurações de redes primárias de distribuição aéreas correspondem a um grande percentual dos sistemas elétricos de média tensão, presentes mundialmente [1]. Elas também estão localizadas tanto em ambientes urbanos quanto rurais, dividindo espaço com as redes de distribuição do tipo subterrânea [46, 3].

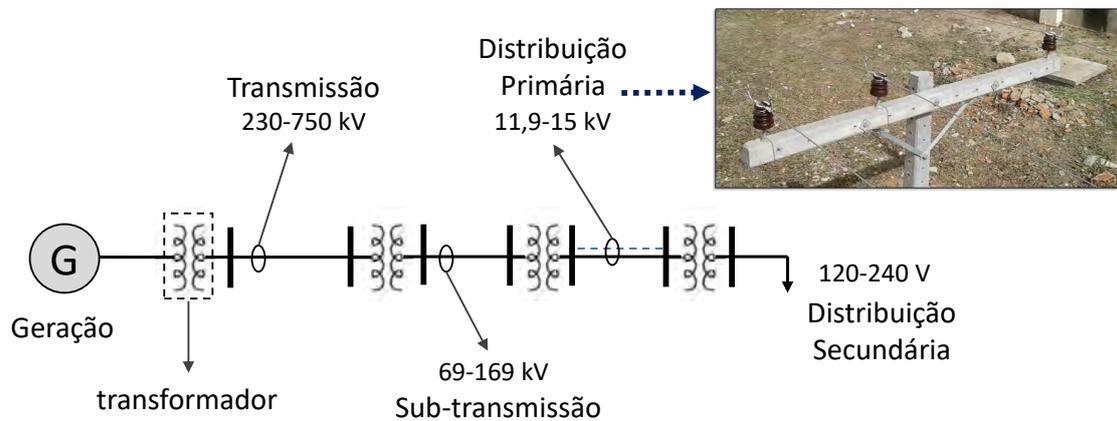


Figura 2.1: Estágios pertencentes à infraestrutura de fornecimento de energia elétrica

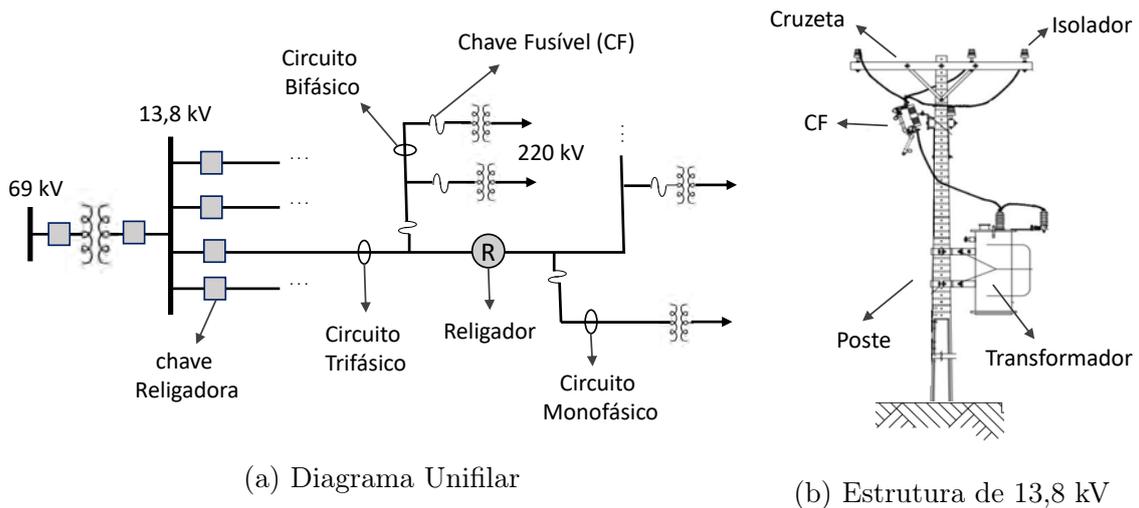


Figura 2.2: Diagrama unifilar de um alimentador típico (a) e estrutura de distribuição de 13,8 kV com transformador instalado (b).

Considerando os componentes e equipamentos presentes nas RPDA, normal-

mente existem diversos estágios de proteção e manobra direcionados a separar os circuitos para manutenções em caso de inconformidades no fornecimento de energia [2]. Os dispositivos mais utilizados para proteção de redes de distribuição primárias são as chaves fusíveis, religadores de tensão e chaves telecomandadas [46]. A Figura 2.2a ilustra o diagrama unifilar de um alimentador típico e seus respectivos equipamentos de proteção. Na Figura 2.2b, por sua vez, é apresentado o *design* de uma estrutura de distribuição de 13,8 kV, integrada a diversos dispositivos, tais como *componentes isoladores de distribuição*, chaves fusíveis e transformador [48].

2.1 Isoladores de Distribuição

Os *Isoladores de Distribuição* são componentes constituintes das RPDA que têm a função de promover o isolamento elétrico de materiais condutores em relação ao aterramento, às estruturas de sustentação e outros componentes interligados ao sistema elétrico de distribuição [37, 36]. Estes dispositivos também são utilizados para dar suporte mecânico rígido ou flexível aos condutores elétricos dos circuitos de distribuição e podem ser empregados em redes constituídas por cabos nus ou isolados [38, 49].



Figura 2.3: Imagens de Isoladores de distribuição coletadas em diferentes ângulos de θ (inclinação vertical) .

Estes componentes são geralmente especificados em termos do tipo de material

construtivo, sendo classificados como: cerâmicos, vítreos, poliméricos e híbridos - constituído por compósitos ou materiais distintos. A Figura 2.3 ilustra seis tipos de isoladores que operam na faixa de 15 kV [37, 36], nomeadamente: Isolador Cerâmico de Pino (ICP), Isolador Cerâmico Bicolor (ICB), Isolador Polimérico Cinza (IPC), Isolador de Vidro Verde (IVV), Isolador de Vidro Transparente (IVT) e Isolador Pilar Híbrido (IPH). Através dessa figura, também é possível observar que estes componentes apresentam, além de uma grande diversidade de materiais construtivos, diferenças acentuadas de formato/*design*, cor, textura e tamanho.

Os isoladores de distribuição, quando instalados em conformidade com as normas técnicas, são usualmente interligados em conjunto com outros materiais, dispositivos e componentes menores [36]. A Figura 2.4 exemplifica os principais componentes conectados aos isoladores de distribuição de 15 kV, listados como: (A) pino de sustentação; (B) amarração metálica e (C) condutor elétrico. O pino de sustentação e amarração, por sua vez, possui a funcionalidade de manter o isolador seguro à estrutura da RPDA e aos condutores elétricos, respectivamente. Vale salientar que o estado de conformidade de cada um dos elementos listados são pré-requisitos importantes para manter as funções elétricas e mecânicas desejadas para o funcionamento adequado dos sistemas elétricos de distribuição [50].

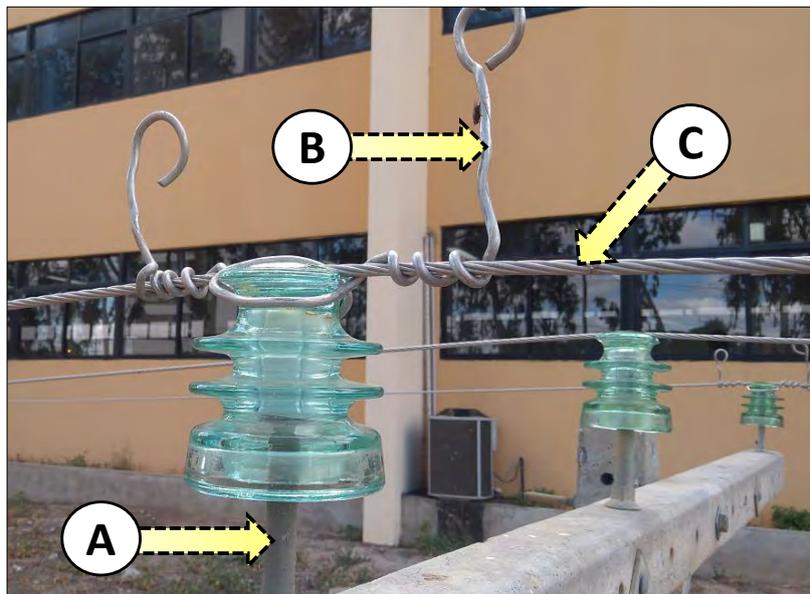


Figura 2.4: Materiais conectados aos isoladores de distribuição em RPDA.

2.2 Inspeção Visual (IV)

As atividades de *Inspeção visual* (IV) podem ser compreendidas como um conjunto de procedimentos direcionados à busca e identificação de defeitos, visíveis a olho nu ou com auxílio de binóculo, existentes em redes de distribuição de energia elétrica [51]. O tipo de inconformidade presente nas RPDA é definida de acordo com a especificidade do dano provocado ao sistema elétrico, podendo ser estratificada da seguinte forma:

- **Defeito:** Anomalia ou imperfeição presente no circuito que pode impactar negativamente no desempenho do sistema elétrico e/ou evoluir para futuramente ocasionar uma interrupção no fornecimento de energia. Este tipo de inconformidade pode ser corrigida através de manutenções preventivas.
- **Falha:** Anomalia ou imperfeição presente no circuito que produz uma interrupção no fornecimento de energia elétrica. Estes tipos de defeitos só podem ser corrigidos através de manutenções corretivas.

Dentre as modalidades de inspeção visual comumente praticadas no setor de distribuição de energia elétrica, três formas de avaliação podem ser mencionadas [51]:

1. **Inspeção expedita:** realizada em pontos pré-determinados da RPDA com a finalidade de detectar a presença de defeitos que estão afetando o fornecimento de energia elétrica.
2. **Inspeção minuciosa:** realizada de forma detalhada e criteriosa, em todas as estruturas, equipamentos e materiais das RPDA.
3. **Inspeção de emergência:** conduzida por equipes de prontidão através da procura de falhas que estão afetando o fornecimento de energia elétrica.

Durante a realização das referidas modalidades de inspeção, diversos tipos de componentes das RPDA são monitorados, a exemplo de: isoladores de distribuição; condutores elétricos; postes e cruzetas; chaves seccionadoras; para-raios; transformadores; conexões e emendas, dentre outros. Esta grande diversidade de componentes avaliados nas IVs demonstram o alto nível de complexidade inerente às atividades de inspeção visual e à sua importância para a manutenção da qualidade do fornecimento de energia elétrica.

2.3 Defeitos em Isoladores de Distribuição

Dentre os componentes monitorados nas atividades de inspeção visual, os isoladores de distribuição requerem uma atenção especial, haja vista que estes componentes isolam eletricamente os condutores energizados das estruturas de sustentação das RPDA. Eles também estão expostos à ação de influências externas, a exemplo de mudanças climáticas e descargas atmosféricas, e podem apresentar alteração de composição química, degradação mecânica e redução da rigidez dielétrica com o passar do tempo [52]. Um defeito de isolação pode facilmente evoluir para falhas catastróficas no sistema elétrico e/ou ocasionar riscos à segurança de pessoas e animais que estejam nas proximidades [8, 9].

Considerando as atividades de inspeção visual direcionados à identificação de defeitos em isoladores de distribuição, em condições realistas de operação, é possível listar os principais tipos de defeitos visíveis a olho nu e prováveis agentes causadores [4]:

1. **Quebra, rachadura ou fratura:** Danos mecânicos ocasionados na superfície dos componentes em decorrência de desgaste natural, descargas atmosféricas, excesso de tração de condutores elétricos ou ações de vandalismo. Estas inconformidades podem gerar alterações visuais significativas nos isoladores em relação ao formato, cor ou textura (ver Figuras 2.5 e 2.6).



Figura 2.5: Imagens de componentes quebrados.

2. **Deformação:** Danos físicos presentes no corpo do componente polimérico ocasionados por temperaturas elevadas ou altas trações mecânicas. As origens típicas para este tipo de defeito estão relacionadas às descargas atmosféricas, curtos-circuitos e arcos elétricos acometidos nas redes de distribuição. Estes defeitos usualmente provocam alterações de formatos e cor dos componentes (ver Figuras 2.7a e 2.7b).



(a) Rachadura.



(b) Fratura.

Figura 2.6: Imagens de componentes cerâmicos rachado e fraturado.

3. **Manchas:** Alterações de cor nos componentes proporcionadas, principalmente, por curtos-circuitos ou efeitos da poluição atmosférica (ver Figuras 2.7c e 2.7d) [52].



(a) A



(b) B



(c) C



(d) D

Figura 2.7: Componentes deformados (A e B) e com alterações de cor (C e D).

4. **Ausência de componentes:** Inexistência de isolador, amarração ou outro componente. Podem ser ocasionadas por falhas na execução do projeto ou devido a danos mecânicos que ocasionem a completa destruição dos componentes - relatados no item 1 (ver Figura 2.8).
5. **Objetos estranhos:** Presença de artefatos estranhos nas proximidades do isolador, tais como ninho de pássaros e animais de pequenos porte. A Figura 2.9 ilustra como este tipo de inconformidade pode dificultar a visualização completa dos componentes e também podem servir como uma fonte de interferência para eventuais tarefas de inspeção visual.



(a) Componente isolador ausente.



(b) Amarrações ausentes.

Figura 2.8: Inconformidades relacionadas à ausência de componentes.



Figura 2.9: Ninhos de passarinho localizados próximos aos isoladores.

2.4 Sistema Inteligente de Inspeção Visual (SIV)

Para ampliar o nível de compreensão sobre as potenciais alternativas para o desenvolvimento de sistemas artificiais de inspeção visual de RPDA, apresentar-se-á, de agora em diante, a denominação de *Sistema inteligente de Inspeção Visual (SIV)*. Para esta pesquisa, considera-se o SIV como um conjunto de tecnologias, composto tanto por *hardware* quanto por *software*, que atua de forma combinada para execução autônoma de determinadas tarefas inerentes às atividades de inspeção visual, a exemplo da classificação do material/*design* de componentes e do diagnóstico de defeitos.

Dentre as possíveis abordagens para o advento de tecnologias inteligentes de inspeção visual, Prates et al. [33] sugeriram o desenvolvimento de um sistema de monitoramento automatizado, acoplado a um VANT (Veículo Aéreo não Tripulado), que se utiliza de técnicas de aprendizado de máquina e de processamento digital de imagens com o propósito de executar tarefas específicas das SIVs, a exemplo da identificação dos principais tipos de componentes isoladores e da avaliação dos seus respectivos *status* de conformidade. A Figura 2.10 ilustra uma metodologia, considerada como apropriada pelos autores, para compor um sistema de inspeção

visual automático de RPDA. Neste exemplo hipotético de aplicação, o *drone* seria composto de um sistema de navegação autônomo e teria uma câmera CCD (*charge-coupled device*) acoplada. Em um primeiro momento, ele realizaria o sobrevoo e coleta de imagens/vídeos da rede de distribuição, para posteriormente executar as etapas de processamento das informações. A troca de sinais entre o *drone* e o computador ocorreria através de comunicação remota, sem fios.

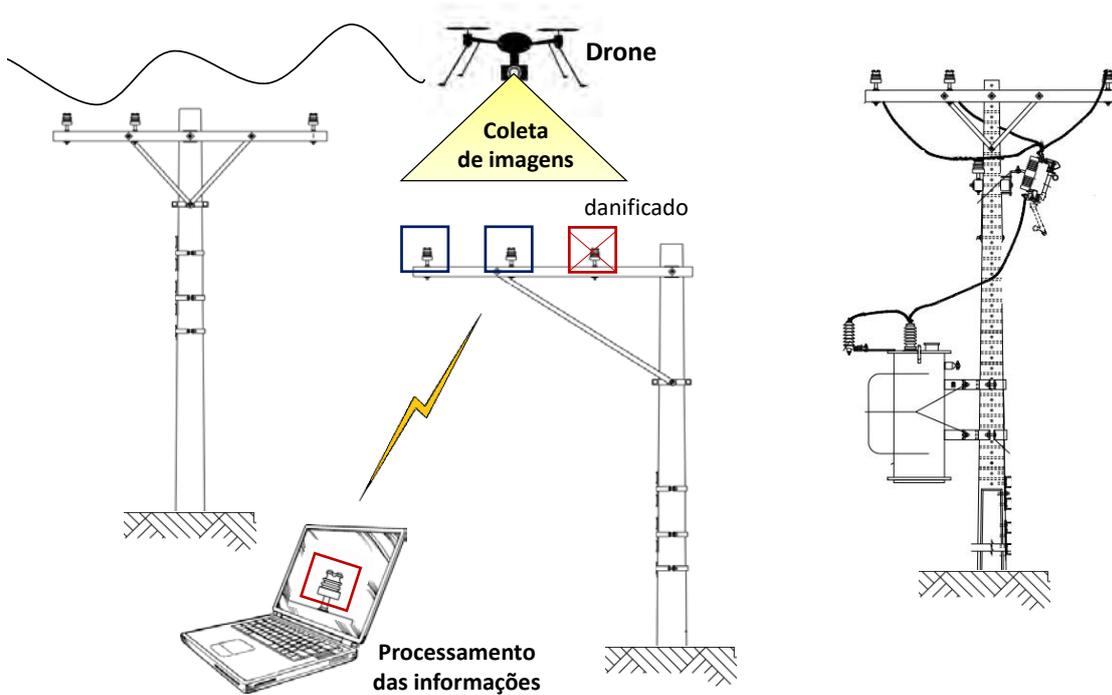


Figura 2.10: Ilustração de uma forma automatizada para inspeção de RPDA através do uso de *Drones*.

Como pré-requisito para o progresso do desenvolvimento de sistemas inteligentes de inspeção adaptados às características de mérito das RPDA apresentadas neste capítulo, necessita-se desenvolver uma série metodologias que farão parte das etapas de processamento da informação das SIVs. Dentre as demandas identificadas para área de visão computacional, pode-se mencionar algumas aplicações importantes, a exemplo de: modelos inteligentes para extração de características e classificação dos tipos de componentes avaliados, independente do seu nível de conformidade [34, 31, 53, 32]; metodologias de segmentação de imagens [32] e de elevação da imunidade a ruídos de planos de fundo (*background invariance*) [35]; técnicas de regularização e otimização do processo de aprendizagem de modelos inteligentes direcionadas ao diagnóstico de inconformidades de componentes específicos das

RPDA [33]. A Subseção 2.4.1 apresentará as principais abordagens, ainda consideradas no *estado da arte*, vinculadas aos estágios necessários para compor um sistema de inspeção visual automático de redes elétricas de distribuição.

2.4.1 Estado da Arte da Inspeção Automática

Os sistemas de inspeção visual automatizados de redes primárias de distribuição aérea são considerados como tecnologias de alto nível de complexidade. Dentre as dificuldades inerentes ao desenvolvimento de SIVs, algumas já foram relatadas em trabalhos acadêmicos, tais como: alta variabilidade de planos de fundos provenientes de ambientes reais (urbanos e rurais); diferenças nas condições climáticas e de iluminação de ambientes externos; presença de circuitos energizados e sem isolamento; além de mudanças rápidas de visão das câmeras pertencentes aos equipamentos de coleta de imagens [54, 55]. Outro obstáculo relevante está relacionado à geração de *datasets* representativos para todo o universo de redes de distribuição, tendo em vista que os ativos monitorados podem apresentar diferenças significativas de formato, cor, textura e características visuais de potenciais defeitos [51, 52].

Em relação à etapa de aquisição de imagens, alguns autores sugeriram, como soluções viáveis, a aplicação de máquinas móveis, como Veículos Aéreos Não Tripulados (VANTs) ou robôs escaladores articulados [6, 7]. Cada uma dessas abordagens possui características particulares de nível de autonomia de energia, capacidade de acoplamento de carga e grau de liberdade para coleta das informações [7].

No que se refere ao desenvolvimento de sistemas inteligentes de análise de inconformidades em RPDA, existem diferentes metodologias incipientes relatadas na literatura, que focam principalmente na identificação de componentes e detecção de uma gama restrita de tipos de inconformidades. Nestas pesquisas, diferentes tipos de algoritmos inteligentes foram empregados, a exemplo de: Sistema Nebuloso de Inferência Adaptativo (*Adaptive Neuro Fuzzy Inference System* - ANFIS), ν (SVM) e *Hidden Markov Model* (HMM) [7, 56]. Uma revisão da literatura cobre trabalhos existentes de abordagens relacionadas às SIVs considerando diferentes aspectos tecnológicos, tais como: direcionamento do VANT para o objeto desejado, a influência da precisão da câmera e uma ampla gama de métodos de classificação [17].

Considerando o emprego de dispositivos ópticos avançados em aplicações po-

tenciais para inspeção visual autônoma, além dos dispositivos CCD de câmeras digitais, há a possibilidade de utilizar diferentes tipos de sensores de alta sensibilidade [57, 58], ou dispositivos que cubram diferentes faixas espectrais, a exemplo da radiação infravermelha [59]. Isto possui potencial para fornecer respostas ópticas otimizadas a problemas específicos além de melhorar a identificação de defeitos em materiais isolantes e conexões pertencentes às redes de distribuição.

Levando em consideração a implementação de técnicas de aprendizagem profunda para o desenvolvimento de metodologias relacionadas às tarefas de inspeção visual inteligente de RPDA, foi apresentado um estudo inicial sobre as possibilidades e desafios da utilização de redes convolutivas aplicadas aos seguintes estágios de processamento da informação: metodologias de navegação por VANT; técnicas de detecção de objetos; mapeamento e identificação de componentes de redes elétricas [18]. Apesar do elevado potencial da aprendizagem profunda no campo da visão computacional, ao melhor conhecimento dos autores, ainda não existem tecnologias consolidadas ou uma diversidade de estudos que apresentem soluções adequadas para as demandas de geração artificial de base de dados, classificação de componentes de distribuição e diagnóstico de seus respectivos níveis de conformidade.

Capítulo 3

Sistemas Inteligentes de Classificação

As principais etapas de um sistema de reconhecimento de padrões são a aquisição de dados, o pré-processamento, a extração de características e a classificação dos padrões [60, 61]. A Figura 3.1 exemplifica um diagrama de operação típico de um sistema de classificação. Estes sistemas também podem ser interpretados como modelos capazes de associar um determinado evento de entrada a uma classe de padrões previamente conhecida [62] ou a novas classes (também intituladas como novidades), reconhecendo características que se diferem, em certos aspectos, dos dados disponíveis durante as etapas de ajuste dos parâmetros do sistema [63, 64].

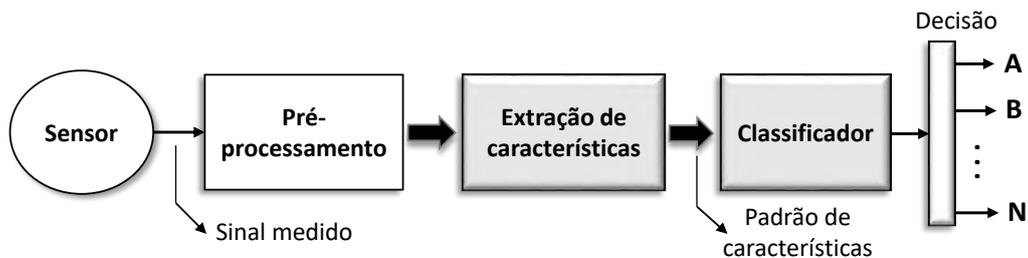


Figura 3.1: Fluxo de operação de um sistema de classificação.

Um sistema inteligente de classificação, por sua vez, deve ser constituído por um modelo matemático capaz de ser otimizado através de um conjunto de dados de entrada e/ou saída. De forma simplificada, pode-se dizer que o referido sistema é capaz de “aprender” através de experiências anteriores (etapa de treinamento) em relação a um determinado tipo de tarefa. Ao longo do processo de aprendizado, realiza-se a otimização contínua do modelo até que seu desempenho alcance um determinado patamar [65]. No caso do fluxo de operação ilustrado na Figura 3.1, a

inteligência do sistema está situada no classificador, etapa onde ocorre o processo de decisão. Dentre a grande diversidade de algoritmos inteligentes, também é possível identificar determinadas arquiteturas que podem aglutinar mais de uma etapa inerente ao fluxo de operação do sistema de classificação. Na aprendizagem profunda, por exemplo, as etapas de extração de características e classificação costumam ficar distribuídas em diferentes camadas do modelo computacional [66].

Os projetos de sistemas inteligentes de classificação são normalmente constituídos por duas etapas principais. A primeira delas está relacionada às atividades de seleção, treinamento e validação do modelo. A segunda etapa é responsável por testar o modelo já treinado, avaliando sua capacidade de identificar corretamente novas amostras. De modo geral, o objetivo do treinamento é encontrar uma função discriminante que seja capaz de classificar os eventos de interesse, para que o modelo apresente baixa ocorrência de classificações incorretas [67]. Assim, um classificador pode ser definido por uma função C_L , expressa da seguinte forma

$$C_L : \mathbb{R}^N \rightarrow \Omega, \quad (3.1)$$

onde $\Omega = \{\omega_1, \dots, \omega_p\}$ é o espaço de classes do problema, N é o número de atributos necessários para realizar a classificação e p é o número de classes possíveis. Um sistema de classificação canônico, apresentado por Kuncheva [68], é considerado como um conjunto de p funções discriminantes que podem ser organizadas da forma $G = \{g_1(\mathbf{x}), \dots, g_p(\mathbf{x})\}$, em que

$$g_i : \mathbb{R}^N \rightarrow \mathbb{R}, i = 1, \dots, p, \quad (3.2)$$

e cada um produz uma pontuação para a respectiva classe. A função com maior resultado de saída é escolhida para compor o rótulo (*label*) do classificador. Essa escolha do *label* é conhecida como *regra de associação máxima*, podendo ser expressa como

$$C_L(\mathbf{x}) = \omega_i \in \Omega \iff g_{\max}(\mathbf{x}) = \max [g_i(\mathbf{x})], i = 1, \dots, p \quad (3.3)$$

As funções discriminantes particionam o espaço de características \mathbb{R}^N em p regiões de decisão (ou regiões de classificação), denotadas por $\mathcal{R}_1, \dots, \mathcal{R}_p$, sendo:

$$\mathcal{R}_i = \{\mathbf{x} | \mathbf{x} \in \mathbb{R}^N, g_i(\mathbf{x}) = \max [g_k(\mathbf{x}), k = 1, \dots, p]\}, i = 1, \dots, p. \quad (3.4)$$

A região de decisão para a classe ω_i é o conjunto de pontos para os quais a i -ésima função discriminante tem a maior pontuação. Essas regiões de decisão são especificadas pelo classificador C_L , ou, de forma equivalente, pelas funções discriminantes presentes em G .

3.1 Problemas de Decisão entre Hipóteses

Considerando a discriminação entre duas hipóteses (decisão binária), H_0 e H_1 , o problema de classificação pode ser entendido através do diagrama apresentado na Figura 3.2. Para estas hipóteses, normalmente se associa H_1 à classe de interesse (hipótese positiva) e H_0 representa à classe não relevante (hipótese negativa). Considerando que as referidas hipóteses são associadas a uma única saída, de acordo com uma determinada regra de decisão estabelecida pelo usuário, é possível representar um ponto no espaço de observação através de um conjunto de valores, expresso por \mathbf{r} , mapeados em uma região do espaço de dimensão p . Obtém-se, por conseguinte, o vetor:

$$\mathbf{r} = [r_1, r_2, \dots, r_p]. \quad (3.5)$$

O *mecanismo de transição probabilística*, também representado na Figura 3.2, gera pontos de acordo com as *funções de densidade de probabilidade condicionais* $P_{\mathbf{r}|H_0}(\mathbf{R}|H_0)$ e $P_{\mathbf{r}|H_1}(\mathbf{R}|H_1)$. Segundo Shanmugan e Breipohl [69], quando as referidas funções são conhecidas (ou podem ser estimadas), pode-se utilizar procedimentos clássicos para tomada de decisão, tais como: Neyman-Pearson, Minimax, Bayes e Máximo a Posteriori (MAP).

A decisão entre as hipóteses também pode ser realizada através de uma sequência de observações do processo avaliado. Considerando o sinal (fonte) de interesse como \mathbf{s} e que a fonte de sinais pode conter ruídos aditivos, representados pela variável \mathbf{n} , o sinal observado \mathbf{r} pode ser obtido como

$$H_0 : \mathbf{r} = \mathbf{n} \quad (3.6)$$

$$H_1 : \mathbf{r} = \mathbf{n} + \mathbf{s} \quad (3.7)$$

Espera-se que o sistema de classificação opere sobre o sinal recebido (\mathbf{r}) e seu processo de decisão ocorra para uma das duas hipóteses. Uma regra de decisão define

o particionamento do espaço de observação nas regiões Z_1 e Z_0 , para as quais são associadas as hipóteses H_1 e H_0 , respectivamente, a depender dos valores observados em \mathbf{r} .

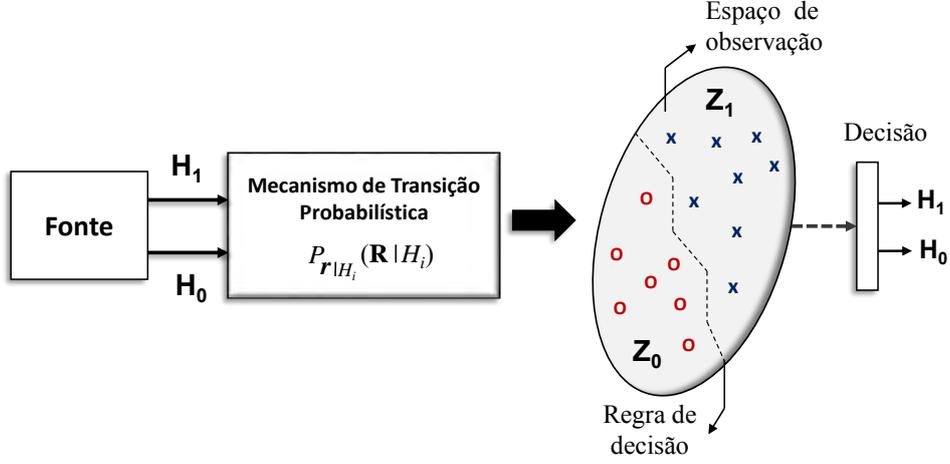


Figura 3.2: Problema de classificação entre duas hipóteses (decisão binária).

De acordo com Van Tress et al. [70], o critério de Bayes pode ser empregado baseando-se em dois pressupostos. No primeiro deles, as saídas das fontes são governadas por atribuições de probabilidades *a priori*, indicadas por $\mathcal{P}_{r0}(\cdot)$ e $\mathcal{P}_{r1}(\cdot)$. Estas probabilidades representam as informações do observador sobre os valores provenientes da fonte. O segundo pressuposto está relacionado às funções custo inerentes aos quatro cursos de ação possíveis, sendo eles: $\mathcal{L}_{H0,H0}$, $\mathcal{L}_{H1,H0}$, $\mathcal{L}_{H1,H1}$ e $\mathcal{L}_{H0,H1}$. O primeiro subscrito indica a hipótese escolhida e o segundo, a hipótese verdadeira. De posse dessas informações, é possível traçar a função risco \mathcal{R}_f em termos das funções de densidades de probabilidades condicionais e das regiões de decisão:

$$\begin{aligned}
 \mathcal{R}_f = & \mathcal{P}_{r0}(\mathcal{L}_{H0,H0}) \int_{z_0} P_{\mathbf{r}|H_0}(\mathbf{R}|H_0) d\mathbf{R} \\
 & + \mathcal{P}_{r0}(\mathcal{L}_{H1,H0}) \int_{z_1} P_{\mathbf{r}|H_0}(\mathbf{R}|H_0) d\mathbf{R} \\
 & + \mathcal{P}_{r1}(\mathcal{L}_{H1,H1}) \int_{z_1} P_{\mathbf{r}|H_1}(\mathbf{R}|H_1) d\mathbf{R} \\
 & + \mathcal{P}_{r1}(\mathcal{L}_{H0,H1}) \int_{z_0} P_{\mathbf{r}|H_1}(\mathbf{R}|H_1) d\mathbf{R},
 \end{aligned} \tag{3.8}$$

onde $\mathcal{P}_{r0}(\mathcal{L}_{H1,H1})$, por exemplo, apresenta a probabilidade *a priori* relativa ao curso de ação em que H_1 foi a hipótese escolhida, sendo condizente com a hipótese ver-

dadeira, vinculada à classe de interesse. Conforme apresentado na equação 3.8, de acordo com o critério de Bayes, é necessário delimitar as regiões de decisão Z_1 e Z_0 , de tal maneira que a função risco seja minimizada.

Ainda considerando a equação 3.8, pode-se inferir duas modalidades de cálculo de probabilidade:

1. Probabilidade de detecção (P_D):

$$P_D = \int_{z_1} P_{\mathbf{r}|H_1}(\mathbf{R}|H_1)d\mathbf{R}, \quad (3.9)$$

Este parâmetro está relacionado com a medida de acerto de discriminação da classe de interesse, podendo ser definido como a probabilidade do classificador de identificar a hipótese positiva (H_1) corretamente.

2. A probabilidade de falso alarme (P_F):

$$P_F = \int_{z_1} P_{\mathbf{r}|H_0}(\mathbf{R}|H_0)d\mathbf{R}. \quad (3.10)$$

A P_F está relacionada à aprovação do classificador de exemplos da hipótese negativa (H_0) como pertencentes à classe de hipóteses positivas (H_1).

A partir da simplificação da equação 3.8, é possível definir as regiões de decisão presentes no espaço de observação através da declaração 3.11 [70]:

$$\frac{P_{\mathbf{r}|H_1}(\mathbf{R}|H_1)}{P_{\mathbf{r}|H_0}(\mathbf{R}|H_0)} \underset{H_0}{\geq} \frac{\mathcal{P}_{r0}(\mathcal{L}_{H_1,H_0} - \mathcal{L}_{H_0,H_0})}{\mathcal{P}_{r1}(\mathcal{L}_{H_0,H_1} - \mathcal{L}_{H_1,H_1})}. \quad (3.11)$$

A razão de verossimilhança, $L(\mathbf{R})$, entre as funções densidade de probabilidade da fonte, em cada hipótese, maximiza a discriminação do sinal de interesse:

$$L(\mathbf{R}) = \frac{P_{\mathbf{r}|H_1}(\mathbf{R}|H_1)}{P_{\mathbf{r}|H_0}(\mathbf{R}|H_0)}. \quad (3.12)$$

O valor de patamar λ pode ser calculado através da equação 3.13

$$\lambda = \frac{\mathcal{P}_{r0}(\mathcal{L}_{H_1,H_0} - \mathcal{L}_{H_0,H_0})}{\mathcal{P}_{r1}(\mathcal{L}_{H_0,H_1} - \mathcal{L}_{H_1,H_1})}, \quad (3.13)$$

Dessa forma, o emprego do critério de Bayes viabiliza a realização do Teste da Razão de Verossimilhança (em inglês, *Likelihood Ratio Test* - LTR), definido por:

$$L(\mathbf{R}) \underset{H_0}{\geq} \underset{H_1}{\lambda}. \quad (3.14)$$

Se a razão de semelhança $L(\mathbf{R})$ for maior que λ , decide-se por H_1 , caso contrário, escolhe-se H_0 . No limite, quando $L(\mathbf{R}) = \lambda$, pode-se decidir tanto por H_1 como por H_0 [68].

3.1.1 Decisão entre Múltiplas Hipóteses

Considerando um sistema de classificação em que são avaliadas múltiplas hipóteses (decisão p -ária), conforme diagrama ilustrado na Figura 3.3, é possível verificar que o espaço de observação foi dividido em p regiões, estabelecidas na regra de decisão para $P_{\mathbf{r}|H_i}(\mathbf{R}|H_i)$ [69].

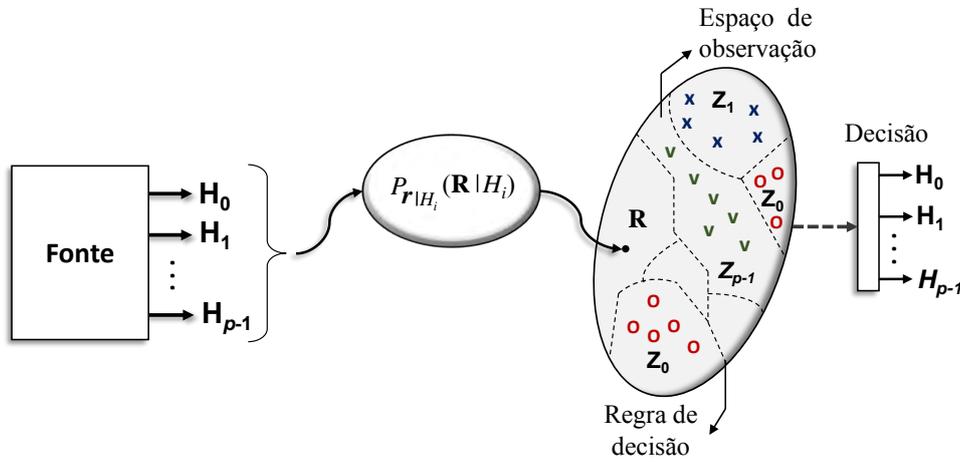


Figura 3.3: Problemas de decisão entre múltiplas hipóteses (decisão p -ária).

Para este caso, a expressão do risco é construída a partir da equação 3.15 [70], que incorpora todas as classes H_{p-1} presentes no processo de decisão:

$$\mathcal{R}_f = \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} \mathcal{P}_{rj}(\mathcal{L}_{H_i, H_j}) \int_{z_i} P_{\mathbf{r}|H_j}(\mathbf{R}|H_j) d\mathbf{R} \quad (3.15)$$

Os espaços de observação são definidos por z_i ; as probabilidades a priori são \mathcal{P}_{rj} e as funções custos são obtidas através de \mathcal{L}_{H_i, H_j} - o primeiro subscrito aponta a i -ésima hipótese escolhida e o segundo corresponde j -ésima afirmação verdadeira.

Este sistema de classificação também pode ser tratado através de um conjunto de p problemas de detecção binária entre H_i e H_j , considerando $j = 1, \dots, I - 1, I + 1, \dots, p$ ou utilizando discriminadores apropriados para problemas de múltiplas hipóteses.

3.2 Algoritmos de Classificação

Nas áreas de inteligência computacional e estatística, os algoritmos de classificação aparecem como ferramentas poderosas para identificação de quais tipos de categorias (subpopulações) pertencem as novas observações num determinado conjunto de dados. Na terminologia de aprendizado de máquina [71], o termo *aprendizado supervisionado* (em inglês, *supervised learning*) pode ser considerado como uma instância de aprendizagem por comparação, que consiste em alguma medida de performance da diferença entre as saídas dos modelos e seus respectivos rótulos. Isto implica que uma base de dados de observações corretamente identificadas está disponível. Nos casos semi-supervisionado ou não-supervisionado, por outro lado, o processo de aprendizagem pode ocorrer através de agrupamentos (em inglês, *clustering*), que envolve a organização de dados em categorias com base em alguma medida/comparação de similaridade ou distância [67].

Um modelo computacional concebido para classificação de padrões é conhecido como classificador. O termo “classificador”, em algumas situações, também se refere a uma função matemática implementada por um algoritmo que mapeia dados de entrada em uma determinada categoria. Atualmente, existe uma grande variedade de modelos conhecidos na literatura, com níveis de complexidade e custos computacionais distintos. O processo de escolha do classificador é uma atividade heurística que requer análises apuradas e um grande número de testes para cada tipo de *dataset* trabalhado. Estas limitações na escolha dos algoritmos de classificação podem ser explicadas à luz dos teoremas *no free lunch* [72], que estabelecem que, para qualquer tipo modelo inteligente, elevados desempenhos sobre uma determinada classe de problemas são compensados pela redução de performance sobre outras classes [73, 74]. Deste modo, ainda não é possível determinar a existência de um “modelo universal” mais eficiente para todos os tipos de tarefas.

No entanto, alguns modelos computacionais já demonstraram vantagens competitivas para determinados tipos de base de dados. Certos classificadores se mostraram mais adequados para sinais de áudio, por exemplo, enquanto outros se adéquam melhor a dados matriciais, como imagens e vídeos. Neste capítulo, será abordado um conjunto de classificadores clássicos que ainda são muito empregados em aplicações da área de inteligência computacional.

3.2.1 Rede Neural Artificial (RNA)

As Redes Neurais Artificiais (RNA) podem ser interpretadas como modelos computacionais, biologicamente inspirados, capazes de aproximar funções não-lineares complexas e de realizar reconhecimento estatístico de padrões [75, 76]. Para Haykin [77], a RNA é um tipo de processador maciçamente distribuído, composto de unidades de processamento simples, conhecidas como neurônios ou perceptrons. Estes neurônios possuem um conjunto de pesos sinápticos (ou elos de conexão), que são capazes de armazenar informações experimentais e torná-las disponíveis para uso posterior do modelo computacional.

Neurônio Artificial: Para melhor entendimento do comportamento da RNA, faz-se necessária uma breve descrição de um dos seus principais componentes constituintes - o neurônio artificial. A Figura 3.4 ilustra um modelo para o neurônio artificial e seu respectivo fluxo de sinal.

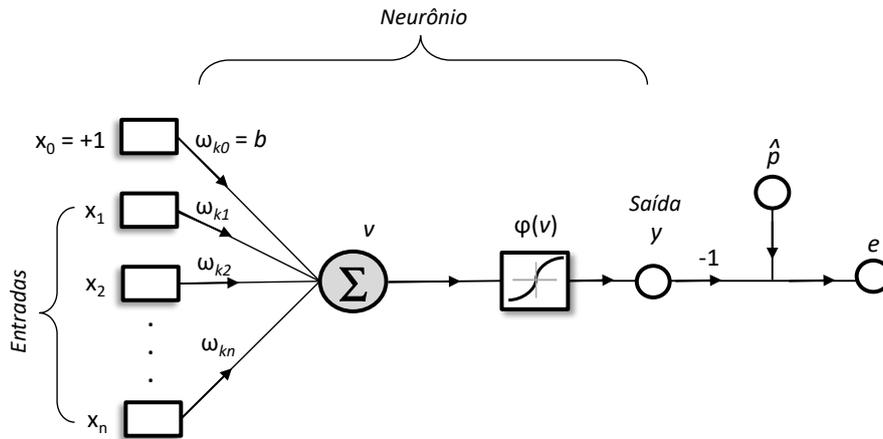


Figura 3.4: Modelo de neurônio artificial e respectivas conexões na RNA.

Este modelo apresenta três elementos principais:

1. Um conjunto de pesos sinápticos ou elos de conexão ($w_{k1}, w_{k2}, \dots, w_{km}$), que serão multiplicados pelos sinais de entrada (x_1, x_2, \dots, x_m);
2. Um bloco somador $v_k(\cdot)$, que fará a somatória dos sinais de entrada ponderados pelos respectivos pesos sinápticos, promovendo uma combinação linear;
3. Uma função de ativação $\varphi(v_k)$, que é empregada para fornecer um comportamento não-linear à função de entrada e também para restringir a amplitude

de saída de um neurônio [78].

Obtém-se então a seguinte equação para a saída do combinador linear:

$$v_k = \sum_{j=1}^m w_{kj}x_j + b_k = \sum_{j=0}^m w_{kj}x_j, \quad (3.16)$$

onde b_k é o viés ou *bias*, representado na Figura 3.4 como $w_{k0} = b$. O sinal de saída do neurônio será definido por

$$y_k = \varphi(v_k), \quad (3.17)$$

No caso da função de ativação $\varphi(\cdot)$, será realizada uma descrição mais aprofundada na Subseção 3.6.5.

Perceptron de Múltiplas Camadas (MLP): As redes neurais artificiais normalmente são constituídas por diferentes estágios de processamento, também conhecidos como camadas. Para melhor definição dos referidos estágios, podemos exemplificar um modelo de rede como mostrado na Figura 3.5, formado por um conjunto de unidades sensoriais que representa a camada de entrada; uma ou mais camadas ocultas (constituídas por neurônios) e uma camada de saída composta por nós computacionais [77].

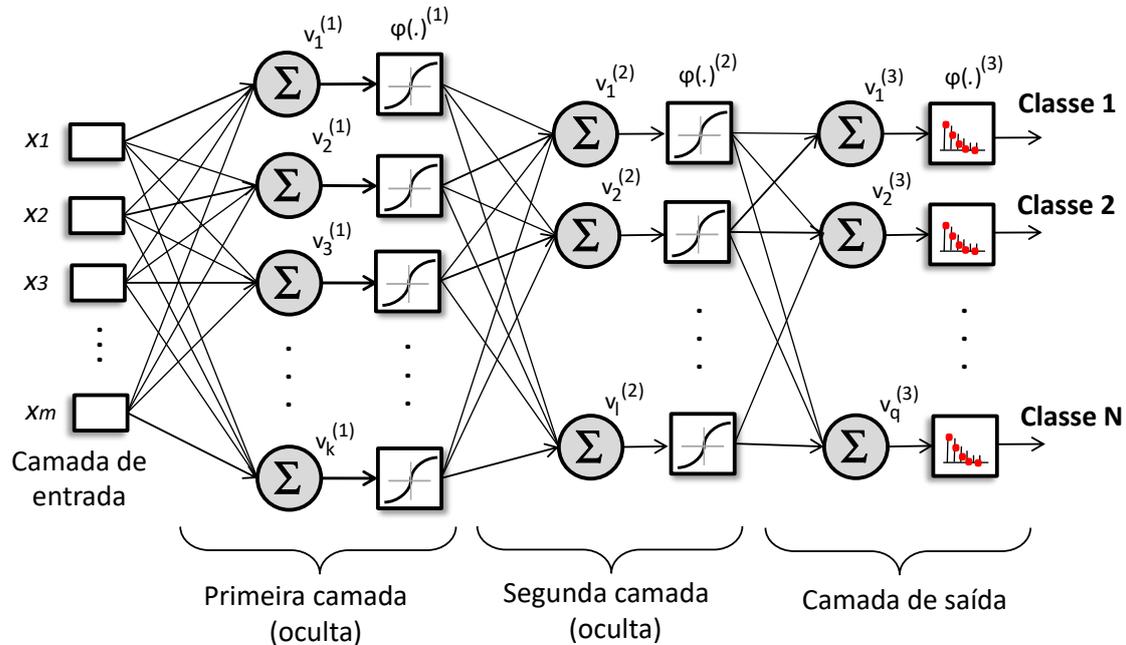


Figura 3.5: Planta arquitetural de Rede Neural Artificial (RNA) do tipo *Multi-Layer* Perceptron com duas camadas ocultas.

Segundo Goodfellow et al. [79], estes modelos matemáticos também são conhecidos como redes neurais *feedforward* (em português, “alimentados para frente”). Para este tipo de RNA, um sinal de entrada, definido em $\mathbf{x} = [x_1, x_2, \dots, x_m]$, se propaga para frente através da RNA, camada por camada, resultando em uma categoria (classe), expressa por

$$\mathbf{y} = f(\mathbf{x}; \mathbf{W}). \quad (3.18)$$

Sendo que \mathbf{W} representa os parâmetros dos pesos sinápticos e dos *bias*. Posteriormente, os valores de \mathbf{W} podem ser ajustados para se obter a melhor aproximação da função desejada. Este tipo de rede neural é formalmente intitulada de Perceptron de Múltiplas Camadas (em inglês, *Multilayer Perceptron* - MLP). A Figura 3.5 ilustra o grafo arquitetural de uma RNA do tipo MLP, com a presença da camada de entrada, duas camadas ocultas e uma camada de saída. Nesta imagem, também é possível identificar a presença de diferentes tipos de função de ativação, a exemplo da função *tangente hiperbólica* (apresentadas matematicamente como $\varphi(.)^{(1)}$ e $\varphi(.)^{(2)}$) e a função Softmax ($\varphi(.)^{(3)}$).

Na arquitetura da RNA do tipo MLP, não estão presentes conexões de realimentação entre os seus neurônios, em que as saídas dos perceptrons são conectadas novamente às suas próprias entradas. Quando a rede neural artificial é adaptada com conexões com realimentação, ela é usualmente chamada de rede neural recorrente (em inglês, *Recurrent Neural Network* - RNN) [80].

A rede neural *Multilayer Perceptron* ainda se mostra uma ferramenta bastante útil para resolver problemas de classificação e regressão através da modalidade de treinamento supervisionado, a exemplo do algoritmo de retropropagação de erro (*backpropagation*) [67], que será mostrado mais adiante neste capítulo.

Aprendizagem por correção de erro: Voltando para o caso do neurônio artificial (Figura 3.4), é possível descrever uma forma simples de correção dos pesos sinápticos através da minimização do sinal de erro da saída, $\varepsilon_k(n)$. Este parâmetro pode ser definido por

$$\varepsilon_k(n) = \hat{p}_k(n) - y_k(n), \quad (3.19)$$

onde $\hat{p}_k(n)$ é a saída desejada e $y_k(n)$ corresponde à saída real do k -ésimo neurônio. A variável n equivale ao índice da iteração do algoritmo de aprendizagem. Conforme apresentado por Hastie et al. [67], torna-se possível definir o valor instantâneo do erro quadrático para o neurônio k como

$$E(n) = \frac{1}{2}\varepsilon_k^2(n), \quad (3.20)$$

Os ajustes iterativos dos pesos sinápticos continuam até que um critério de parada externo seja satisfeito. Esta regra da minimização de $E(n)$ é chamada de *regra delta*, onde o ajuste $\Delta w_{kj}(n)$ aplicado ao peso w_{kj} no passo n , é definido por [71]:

$$\Delta w_{kj}(n) = \eta \varepsilon_k(n) x_j(n), \quad (3.21)$$

onde η é a taxa de aprendizado positiva restrita ao intervalo $0 < \eta \leq 1$, que controla o ajuste aplicado ao vetor de pesos sinápticos em cada iteração n . O valor atualizado do peso w_{kj} pode ser determinado por:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n). \quad (3.22)$$

Também com foco no ajuste dos pesos sinápticos, existem diferentes algoritmos direcionados à etapa de otimização. Um exemplo é o algoritmo de descida estocástica de gradiente (em inglês, *Stochastic Gradient Descent* - SGD) [75]. O SGD tem demonstrado desempenhos satisfatórios para a otimização de diferentes arquiteturas de redes neurais, uma vez que este algoritmo se comporta bem para casos mais desafiadores – quando existem funções custo de alta dimensionalidade e com um elevado número de mínimos locais [79].

Algoritmo de Retropropagação: O algoritmo de retropropagação (conhecido, em inglês, como *backpropagation error*), foi desenvolvido com o objetivo de compor uma forma computacionalmente eficiente para otimização dos pesos sinápticos de diferentes modelos inteligentes, a exemplo das redes neurais dos tipos MLP, convolucionais e recorrentes [79]. De forma simplificada, a operação de retropropagação participa iterativamente no treinamento dos parâmetros ajustáveis dos referidos modelos computacionais, a partir de condições iniciais definidas de forma arbitrária ou com auxílio de algum pré-ajuste (implementação de técnica de transferência de conhecimento) [33]. Em determinados casos, este algoritmo viabiliza que os modelos

empregados desenvolvam estruturas internas adequadas para domínios particulares dos tipos de tarefas a serem executadas.

Rumelhart et al. [81], em sua relevante publicação na revista Nature, intitulada de “*Learning Representations by Back-propagating Errors*”, descreve o referido método de aprendizado e o apresenta como uma forma iterativa de ajuste dos pesos sinápticos de modelos inteligentes de modo a representar características importantes dos domínios das tarefas propostas. Objetiva-se, com esta abordagem, minimizar a diferença entre o resultado computado para o vetor de saída (\mathbf{y}) e o valor desejado, expresso pelo vetor de rótulos, $\hat{\mathbf{p}}$. De acordo com o mesmo autor, o algoritmo de retropropagação de erro se inicia com a operação de diferenciação de uma função que representa o erro de classificação obtido na camada de saída do modelo computacional. Este erro é conhecido como função custo ou índice de desempenho, representado matematicamente como \mathcal{L} . Neste cenário, o valor de \mathcal{L} determina o desempenho da aprendizagem do modelo. Esta função também está associada aos valores de $\varepsilon_k(n)$, estabelecidos na equação 3.19, podendo ser expressa como

$$\mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \mathcal{L}(\mathbf{y}, \hat{\mathbf{p}}), \quad (3.23)$$

sendo que \mathbf{x} e \mathbf{W} representam os sinais de entrada e os pesos sinápticos, respectivamente. Levando em consideração um dos principais requisitos do algoritmo de retro-propagação, a etapa de diferenciação, torna-se necessário computar as derivadas parciais da função custo em relação aos seus respectivos pesos sinápticos, de acordo com $\partial\mathcal{L}/\partial w_{kj}$. Para simplificação deste processo, é possível representar a operação de diferenciação através da regra da cadeia do cálculo, segundo a forma:

$$\frac{\partial\mathcal{L}}{\partial w_{kj}} = \frac{\partial\mathcal{L}}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial v_k} \frac{\partial v_k}{\partial w_{kj}}. \quad (3.24)$$

Segundo Haykin [77], a equação 3.24 representa o fator de sensibilidade do modelo computacional. A partir desta operação, serão definidas as direções mais adequadas para o ajuste dos parâmetros w_{kj} e b_{kj} do modelo no espaço de pesos. Considerando um tipo específico de função custo, que representa o erro médio quadrático (em inglês, *Mean Square Error* - MSE), é possível definir o valor instantâneo da energia média do erro da função, $\mathcal{L}_Q(\mathbf{y}, \hat{\mathbf{p}})$ [76]. Para o n -ésimo padrão de de treinamento,

a função custo do tipo MSE pode ser expressa por:

$$\mathcal{L}_Q(n) = \frac{1}{2} \sum_{k=1}^K [\varepsilon_k(n)]^2 = \frac{1}{2} \sum_{k=1}^K [\hat{p}_k(n) - y_k(n)]^2, \quad (3.25)$$

onde a constante K determina o número de classes do modelo. A partir da equação 3.25, combinada à 3.24, é possível definir alguns parâmetros que compõem o fator de sensibilidade para este caso particular de $\mathcal{L}(\mathbf{y}, \hat{\mathbf{p}})$, sendo eles: $\partial \mathcal{L}_Q / \partial \varepsilon_k = \varepsilon_k$; $\partial \varepsilon_k / \partial y_k = -1$; $\partial y_k / \partial v_k = \varphi'(v_k)$ e $\partial v_k / \partial w_k = y_k$. Por conseguinte, a equação 3.24 poderá ser expressa como

$$\frac{\partial \mathcal{L}_Q(n)}{\partial w_{kj}(n)} = -\varepsilon_k(n) \varphi'(v_k(n)) y_k(n). \quad (3.26)$$

Uma forma simples de implementar o gradiente descendente no espaço de pesos, em que será dado o processo de otimização da função que representa o modelo inteligente, pode ocorrer através da correção dos valores de $\partial \mathcal{L}_Q / \partial w_{kj}$ a partir de uma constante de proporcionalidade, definida como η [81]. De acordo com a *regra do delta*, apresentada na equação 3.27, pode-se obter $\Delta w_{kj}(n)$ conforme:

$$\Delta w_{kj}(n) = -\eta \frac{\partial \mathcal{L}_Q(n)}{\partial w_{kj}(n)}. \quad (3.27)$$

O emprego do sinal negativo indica a direção de descida do gradiente no espaço de pesos. É possível também obter o gradiente local $\delta_k(n)$ através da equação [77]:

$$\delta_k(n) = \frac{\partial \mathcal{L}_Q(n)}{\partial \varepsilon_k(n)} \frac{\partial \varepsilon_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_j(n)}, \quad (3.28)$$

em que $\delta_k(n)$ pode ser apresentado, de forma mais compacta, por

$$\delta(n) = -\frac{\partial \mathcal{L}_Q(n)}{\partial v_k(n)} = \varepsilon_k(n) \varphi'(v_k(n)). \quad (3.29)$$

Deste modo, o parâmetro de ajuste $\Delta w_{kj}(n)$ pode ser escrito de forma equivalente como

$$\Delta w_{kj}(n) = -\eta \delta(n) y_k(n). \quad (3.30)$$

Considerando o comportamento da equação 3.30, o valor de $\Delta w_{kj}(n)$ não converge tão rapidamente como em técnicas que empregam derivações de segunda ordem, contudo a retropropagação é mais simples de ser implementada e possui menor

custo computacional [81]. Como abordagem adicional, focada na otimização de desempenho do modelo computacional, é possível incorporar um *termo de aceleração* ou *momento* α , resultando na equação 3.31:

$$\Delta w_{kj}(n) = -\eta \delta(n) y_k(n) + \alpha \Delta w_{kj}(n-1). \quad (3.31)$$

Para este segundo caso, o cálculo do gradiente é utilizado para otimização do ponto de velocidade no espaço de pesos, ao invés de ficar restrito somente à sua posição, conforme foi apresentado na equação 3.30.

Com o intuito de promover o aprofundamento teórico do algoritmo de retropropagação, faz-se necessária a realização de uma análise complementar do processo de treinamento da rede neural do tipo perceptron de múltiplas camadas, cuja arquitetura ainda é muito utilizada na resolução de problemas práticos de classificação/regressão inteligente do tipo supervisionada [31]. Delimitando o foco no treinamento de modelos para múltiplas classes, implementa-se a função de ativação do tipo Softmax na camada de saída [67]. Para cálculo da função custo, emprega-se a função de Entropia Cruzada, expressa por \mathcal{L}_s , que pode ser escrita na forma [82]:

$$\mathcal{L}_s[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \frac{1}{K} \sum_{k=1}^K -y_k \log f_k(\mathbf{x}). \quad (3.32)$$

A constante K determina o número de classes do modelo e $f_k(\mathbf{x})$ é a função de ativação Softmax, que compõe um dos elementos do vetor $\mathbf{f} = [f_1(\mathbf{a}), \dots, f_K(\mathbf{a})]$, definido por

$$f_k(\mathbf{x}) = \frac{\exp[a_k(\mathbf{x}, \mathbf{W})]}{\sum_{c=1}^K \exp[a_c(\mathbf{x}, \mathbf{W})]}. \quad (3.33)$$

As variáveis de entrada da equação 3.33, que geram os parâmetros do vetor $\mathbf{a}(\mathbf{x}; \mathbf{W}) = [a_1(\mathbf{x}, \mathbf{W}_1), \dots, a_K(\mathbf{x}, \mathbf{W}_K)]$, podem ser computadas através de

$$a_k(\mathbf{x}; \mathbf{W}) = \mathbf{W}_k^T \mathbf{x} + \mathbf{b}. \quad (3.34)$$

em que \mathbf{x} é o vetor de entrada e \mathbf{W} representa as matrizes de pesos sinápticos, \mathbf{W}_k , e *bias*, \mathbf{b} . A partir do mesmo procedimento de regra da cadeia, apresentado na equação 3.24, é possível definir $\partial \mathcal{L}_s / \partial \mathbf{W}_k$ como:

$$\frac{\partial \mathcal{L}_s}{\partial \mathbf{W}_k} = \frac{\partial \mathbf{a}}{\partial \mathbf{W}_k} \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \frac{\partial \mathcal{L}_s}{\partial \mathbf{f}} = \frac{\partial \mathbf{a}}{\partial \mathbf{W}_k} \frac{\partial \mathcal{L}_s}{\partial \mathbf{a}}. \quad (3.35)$$

Considerando $\partial\mathcal{L}_s/\partial\mathbf{a}$ e $\partial\mathbf{a}/\partial\mathbf{W}_k$, é possível fazer uma série de equacionamentos para facilitar o cálculo do fator de sensibilidade da retro-propagação [83, 78]. Para o primeiro termo, obtém-se a seguinte relação:

$$\frac{\partial\mathcal{L}_s}{\partial\mathbf{a}} = \mathbf{y} - \hat{\mathbf{p}}. \quad (3.36)$$

No caso de $\partial\mathbf{a}/\partial\mathbf{w}_k$, derivando individualmente cada termo de a_k , é possível formular a relação $\partial a_j/\partial\mathbf{w}_k$ como segue:

$$\frac{\partial a_j}{\partial\mathbf{W}_k} = \begin{cases} \frac{\partial}{\partial\mathbf{w}_k} \mathbf{W}_k^T \mathbf{x}, & \text{se } j = k \\ 0, & \text{se } j \neq k \end{cases} \quad (3.37)$$

A relação apresentada na equação 3.37 produz uma matriz esparsa, contendo o vetor \mathbf{x} na sua k -ésima coluna e os termos restantes iguais a zero [78]. Por conseguinte, a combinação das equações 3.36 e 3.37 resulta em:

$$\frac{\partial\mathcal{L}_s}{\partial\mathbf{W}_k} = -(\hat{\mathbf{p}} - \mathbf{y})\mathbf{x}^T = -\mathcal{E}\mathbf{x}^T. \quad (3.38)$$

Isto implica que o cálculo do fator de sensibilidade deste modelo é dado pela multiplicação do erro, $\mathcal{E} = (\hat{\mathbf{p}} - \mathbf{y})$, pela matriz de entrada, \mathbf{x}^T . Em um segundo momento, considerando um modelo com funções de ativação similares em todas as N camadas ocultas e uma função Softmax na camada de saída, os gradientes do k -ésimo vetor de uma matriz de parâmetros serão dados por:

$$\frac{\partial\mathcal{L}_s}{\partial\mathbf{W}_k^{(N+1)}} = -\frac{\partial\mathbf{a}^{(N+1)}}{\partial\mathbf{W}_k^{(N+1)}}\mathcal{E}^{(N+1)} = -\mathbf{H}_k^L\mathcal{E}^{(N+1)}, \quad (3.39)$$

onde $(N+1)$ corresponde ao índice da camada de saída, \mathbf{H}_k^L é uma matriz contendo os valores das funções de ativação das camadas ocultas correspondentes e $\mathcal{E}^{(N+1)}$ corresponde ao erro da camada de saída. Para a camada anterior, de número N , pode-se considerar:

$$\frac{\partial\mathcal{L}_s}{\partial\mathbf{W}_k^{(N)}} = -\frac{\partial\mathbf{a}^{(N)}}{\partial\mathbf{W}_k^{(N)}}\frac{\partial\mathbf{h}^{(N)}}{\partial\mathbf{a}_k^{(N)}}\frac{\partial\mathbf{a}^{(N+1)}}{\partial\mathbf{h}_k^{(N)}}\mathcal{E}^{(N+1)}, \quad (3.40)$$

em que o valor do erro $\mathcal{E}^{(N)}$ pode ser definido como

$$\mathcal{E}^{(N)} \equiv \frac{\partial\mathbf{h}^{(N)}}{\partial\mathbf{a}_k^{(N)}}\frac{\partial\mathbf{a}^{(N+1)}}{\partial\mathbf{h}_k^{(N)}}\mathcal{E}^{(N+1)}. \quad (3.41)$$

De forma simplificada, a equação 3.40 pode ser reescrita utilizando a igualdade contida na equação 3.41, da forma:

$$\frac{\partial \mathcal{L}_s}{\partial \mathbf{W}_k^{(N)}} = -\frac{\partial \mathbf{a}^{(N)}}{\partial \mathbf{W}_k^{(N)}} \mathcal{E}^{(N)}, \quad (3.42)$$

sendo que o erro $\mathcal{E}^{(N)}$, definido em termos de $\mathcal{E}^{(N+1)}$, é calculado de forma recursiva. Para um valor arbitrário de $n_c \leq N$, o valor de $\mathcal{E}^{(n_c)}$ pode ser obtido de acordo com os parâmetros da equação 3.43:

$$\mathcal{E}^{(n_c)} = \frac{\partial \mathbf{h}^{(n_c)}}{\partial \mathbf{a}_k^{(n_c)}} \frac{\partial \mathbf{a}^{(n_c+1)}}{\partial \mathbf{h}_k^{(n_c)}} \mathcal{E}^{(n_c+1)} = \mathbf{S}^{(n_c)} [\mathbf{W}^{(n_c+1)}]^T \mathcal{E}^{(n_c+1)}, \quad (3.43)$$

onde $\mathbf{S}^{(n_c)} = \partial \mathbf{h}^{(n_c)} / \partial \mathbf{a}_k^{(n_c)}$ contém as derivadas parciais das funções de ativação pertencentes às camadas ocultas, localizadas em função da entrada de pré-ativação. O termo $[\mathbf{W}^{(n_c+1)}]^T = \partial \mathbf{a}^{(n_c+1)} / \partial \mathbf{h}_k^{(n_c)}$ é definido em função da seguinte relação [78]:

$$\mathbf{a}^{(n_c+1)}(\mathbf{h}^{(n_c)}) = \mathbf{W}^{(n_c+1)} \mathbf{h}^{(n_c)} + \mathbf{b}^{(n_c+1)}. \quad (3.44)$$

Como resultado, os gradientes da n -ésima camada podem ser computados através de multiplicações matriciais:

$$\frac{\partial \mathcal{L}_s}{\partial \mathbf{W}_k^{(n_c)}} = -\mathbf{H}_k^{(n_c-1)} \mathbf{S}^{(n_c)} [\mathbf{W}^{(n_c+1)}]^T \dots \mathbf{S}^{(N)} [\mathbf{W}^{(N+1)}]^T \mathcal{E}^{(N+1)} \quad (3.45)$$

A partir da análise do conjunto de equações apresentadas para o algoritmo de retropropagação e realizando uma releitura do comportamento da RNA exemplificada na Figura 3.5, pode-se afirmar que a atualização dos gradientes segue um fluxo condizente ao ilustrado no diagrama da Figura 3.6. Neste diagrama de fluxo, cuja representação está limitada a apenas duas camadas ocultas, é possível identificar tanto a relação entre os parâmetros apresentados na equação 3.44 quanto as multiplicações matriciais em cascata que compõem a equação 3.45. Exemplos de metodologias inteligentes que empregam a referida arquitetura de RNA para tarefas de classificação multi-classes podem ser encontradas em [31, 53].

De acordo com LeCun et al. [84], é possível otimizar o processo de retro-propagação através da implementação de técnicas heurísticas. Dentre as abordagens avaliadas pelo autor, vale mencionar: procedimentos de tratamento da base de dados de treinamento, tais como mistura (*shuffling*) e normalização; critérios de inicialização dos pesos sinápticos; modalidades de treinamento (estocástica ou por lote) e ajuste de

hiper-parâmetros, a exemplo da taxa de aprendizado η e do termo de momento α . Em suma, o emprego dessas técnicas, implementadas de forma isolada e/ou combinadas, pode gerar ganhos consideráveis no desempenho de treinamento do modelo e proporcionar melhor adaptação a diferentes domínios das tarefas a serem executadas.

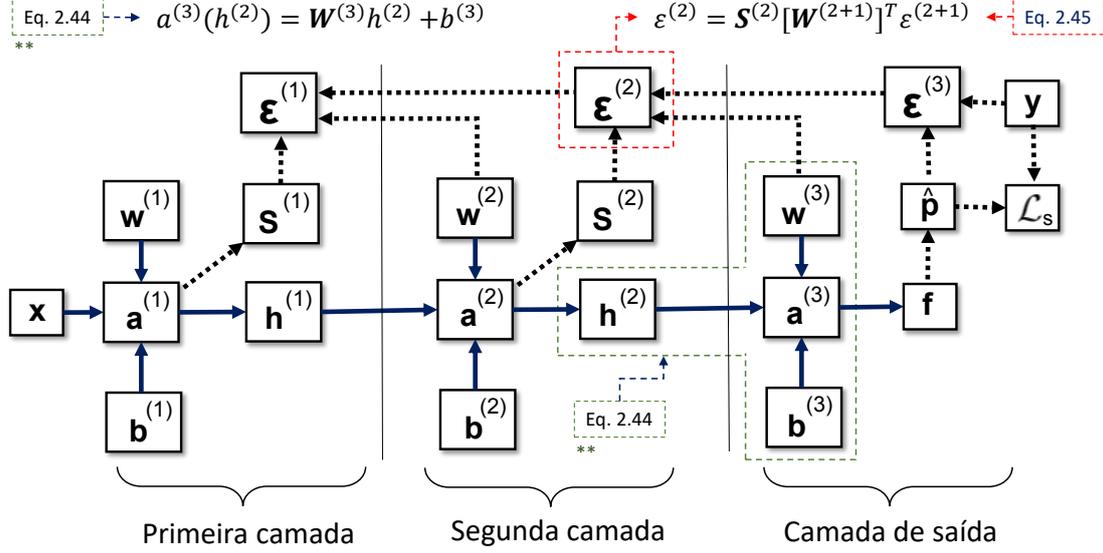


Figura 3.6: Diagrama de fluxo de atualização da retro-propagação erro de uma RNA do tipo MLP com duas camadas ocultas.

Com foco na melhoria da capacidade de generalização dos modelos, Moore et al. [85] apresentou um conjunto de técnicas para punição da complexidade de algoritmos que apresentam problemas de variância. No campo de inteligência computacional, altas variâncias podem significar que os modelos estão sujeitos a ruídos ou ajustados para dados de entrada muito específicos, incorrendo em uma condição indesejada de sobre-treinamento (em inglês, *overfitting*) [78]. Estas modalidades de penalização atuam na etapa de retropropagação e também são conhecidas como uma forma de regularização. Nestes casos, incorpora-se mais um termo $\lambda_R \mathcal{L}_R(\mathbf{w})$ à função custo, resultando em

$$\mathcal{L}(f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}) = \mathcal{L}_D[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \lambda_R \mathcal{L}_R(\mathbf{W}), \quad (3.46)$$

onde $\mathcal{L}_D[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}]$ é a medida de desempenho da rede, já apresentada na equação 3.23. O parâmetro $\mathcal{L}_R(\mathbf{W})$, por sua vez, é conhecido como função de punição da complexidade, que depende unicamente dos pesos sinápticos \mathbf{W} . A constante λ_R é

chamada de *parâmetro de regularização* e representa a importância relativa do termo de punição da complexidade em relação ao termo de medida de desempenho [77].

Tanto o algoritmo de retropropagação quanto as técnicas de penalização da função custo também podem ocorrer em modelos inteligentes mais exóticos, como é caso das Redes Generativas Adversárias (GAN), apresentadas por Goodfellow et. al [21]. Para o cálculo das penalidades, conhecidas como *fatores de diversidade*, adicionam-se novos parâmetros à função custo [86]. Obtém-se, por conseguinte, uma combinação das equações de otimização de redes generativas, propostas por Goodfellow, com funções $\lambda_R \mathcal{L}_z(G)$ que penalizam comportamentos indesejados no treinamento dos modelos computacionais, a exemplo do *mode collapse* [87]. De forma simplificada, a equação resultante pode ser definida por:

$$\min_G \max_D \mathcal{L}_{cGAN}(D, G) - \lambda_R \mathcal{L}_z(G) \quad (3.47)$$

onde G e D representam, respectivamente, os modelos do Gerador e Discriminador das GANs [21]. Vale ressaltar que é necessário que a função custo do modelo computacional seja integralmente diferenciável para que a retropropagação de erro se torne viável. Isto implica que os parâmetros vinculados à regularização e aos fatores de diversidade também devem ser respeitar as características supracitadas. As abordagens heurísticas de otimização dos algoritmos e as técnicas de regularização serão descritas de forma mais aprofundada no Capítulo 3.4.

Algoritmos de otimização: Durante o estágio de treinamento de diversos tipos de modelos inteligentes, a exemplo das redes neurais convencionais e das redes convolutivas, torna-se necessário o cálculo da função custo, $\mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}]$, e posterior execução de um algoritmo de otimização para ajuste dos parâmetros do modelo de forma rápida e assertiva [88]. Para implementação do algoritmo de otimização, é muito comum o uso do *gradiente descendente* (em inglês, *Gradient Descent* - GD) [89] em combinação com o método de retropropagação de erro (*backpropagation*) - apresentado na subseção anterior.

Contudo, o gradiente descendente não se mostra como uma rotina adequada para o treinamento de modelos que apresentem um número elevado de parâmetros ajustáveis - como ocorre na maioria dos algoritmos de aprendizagem profunda [90]-,

ou para base de dados de treino que possuam um grande número de exemplos - existem atualmente *datasets* com milhões de imagens [91]. Esta rotina necessita calcular os gradientes para todas as instâncias na base de dados de uma única vez, acarretando em um custo computacional demasiadamente alto, que poderia dificultar ou até inviabilizar o treinamento de muitos modelos computacionais [92]. Deste modo, de forma a mitigar os problemas inerentes ao gradiente descendente e melhorar do desempenho de treinamento, existem algumas alternativas de algoritmos de otimização que serão descritas a seguir.

- **SGD:** O *gradiente descendente estocástico* (do inglês, *Stochastic Gradient Descent* - SGD) pode ser definido como um método iterativo desenvolvido para suprir algumas deficiências específicas do algoritmo de otimização do tipo gradiente descendente [89]. O SGD substitui o gradiente real, que é calculado a partir de todo o conjunto de dados do *dataset* de treino, por uma estimativa aproximada. Deste modo, o gradiente será calculado a partir de exemplos aleatórios dos dados de treino e de seus respectivos rótulos [92]. A sua equação característica pode ser vista na forma:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \nabla \mathcal{L}[f(\mathbf{x}; \mathbf{W}_t), \hat{\mathbf{p}}]. \quad (3.48)$$

Uma adaptação possível para o SGD é chamada de *gradiente descendente por lote* (em inglês, *Mini-batch gradient descent*) [93]. Nesta operação, o algoritmo de otimização realiza uma atualização de parâmetros para cada *mini-batch* (amostra aleatória de um conjunto de dados) com n número de exemplos. A equação resultante pode ser definida como

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \nabla \mathcal{L} \left[f(\mathbf{x}^{(i:i+n)}; \mathbf{W}_t), \hat{\mathbf{p}}^{(i:i+n)} \right]. \quad (3.49)$$

Segundo Ruder [92], a equação 3.49 proporciona alguns benefícios importantes para a etapa de treinamento de algoritmos inteligentes, tais como: redução na variação das atualizações dos parâmetros, o que pode ocasionar uma convergência mais estável da função custo, e também permite o uso de matrizes otimizadas de bibliotecas de aprendizado profundo com potencial para melhorar o desempenho de computação dos gradientes.

- **Momentum:** De acordo com Qian [94], o *momentum* é uma adaptação do método SGD que ajuda a acelerar o algoritmo na direção de otimização desejada e atenua possíveis oscilações inerentes ao treinamento. Este algoritmo adiciona um novo hiper-parâmetro, chamado de termo de aceleração (α), que possibilita controlar a velocidade de mudança dos parâmetros ajustáveis do modelo computacional. Sua equação pode ser apresentada como:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \alpha(\mathbf{W}_t - \mathbf{W}_{t-1}) + (1 - \alpha)[- \eta \nabla \mathcal{L}(\mathbf{W}_t)] \quad (3.50)$$

O termo de *momentum* aumenta de valor quando os gradientes apontam nas mesmas direções e reduz as atualizações para os casos em que gradientes mudam de direção. Como resultado deste comportamento, obtém-se uma convergência mais rápida e redução de possíveis oscilações nas curvas de otimização [92].

- **RMSProp:** Este método advém de um estudo não publicado, proposto por Geoff Hinton, e possui semelhanças com o algoritmo SGD com *momentum* [92]. Este algoritmo restringe as oscilações na direção vertical. Deste modo, torna-se possível aumentar a taxa de aprendizagem e dar passos maiores na direção horizontal, o que pode resultar numa convergência mais rápida. A principal diferença entre o RMSProp e o SGD está na forma como os gradientes são calculados. Calcula-se primeiramente o *momento de segunda ordem* de $\nabla \mathcal{L}$, expresso como g , gerando a seguinte equação:

$$g_{t+1} = \gamma g_t + (1 - \gamma) \nabla \mathcal{L}(\mathbf{W}_t)^2, \quad (3.51)$$

onde γ é um hiper-parâmetro para ajuste do termo g . A atualização dos parâmetros, com adição de *momentum*, é então realizada através de

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \alpha(\mathbf{W}_t - \mathbf{W}_{t-1}) + (1 - \alpha) \left[- \frac{\eta \nabla \mathcal{L}(\mathbf{W}_t)}{\sqrt{g_{t+1} + \mathcal{E}}} \right]. \quad (3.52)$$

- **Adam:** Segundo os próprios autores deste método (Kingma e Ba [95]), a rotina Adam pode ser entendida como um algoritmo para otimização de funções objetivo estocásticas de primeira ordem, baseado em estimativas adaptativas

de momentos de ordem inferior. Os autores também afirmam que este método é simples de ser implementado e é bastante eficiente computacionalmente. Também é possível afirmar que o Adam opera de forma similar a outros algoritmos de otimização, a exemplo do RMSProp, contudo o RMSProp gera as suas atualizações de parâmetros utilizando um momentum no gradiente redimensionado, enquanto as atualizações da técnica Adam são diretamente estimadas utilizando uma média do momentos de primeira e de segunda ordem do gradiente. Nesta técnica, primeiramente se calcula o termo *momento de primeira ordem* de $\nabla\mathcal{L}$, dado como m , expresso da forma:

$$m_{t+1} = \alpha_{t+1} g_t + (1 - \alpha_{t+1})\nabla\mathcal{L}(\mathbf{W}_t)^2; \quad (3.53)$$

realiza-se então a implementação de um fator de decaimento, gerando o novo termo \hat{m}_{t+1} , que pode expresso como:

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \alpha_{t+1}}. \quad (3.54)$$

Em uma etapa posterior, calculam-se os *momentos de segunda ordem* de $\nabla\mathcal{L}$, g , e aplica-se novamente um fator de decaimento. Resultando nas equações 3.55 e 3.56:

$$g_{t+1} = \gamma_{t+1} g_t + (1 - \gamma_{t+1})\nabla\mathcal{L}(\mathbf{W}_t)^2, \quad (3.55)$$

$$\hat{g}_{t+1} = \frac{g_{t+1}}{1 - \gamma_{t+1}} \quad (3.56)$$

Os valores de α e γ estão presentes para controlar \mathbf{W} e g , respectivamente. Por último, de posse dos termos apresentados, faz-se a atualização dos parâmetros do algoritmo da forma:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{g}_{t+1} + \mathcal{E}}}. \quad (3.57)$$

Um exemplo de aplicação utilizando o algoritmo de otimização Adam pode ser encontrado em [33].

3.2.2 *k*-Nearest Neighbors (*k*NN)

O classificador *k vizinhos mais próximos* (do inglês, *k-Nearest Neighbors* - *k*NN), foi desenvolvido por Fukunaga e Narendra, em 1975 [96]. Ele é um algoritmo simples e de fácil implementação, mas ainda vem apresentando resultados satisfatórios nos dias atuais. O *k*NN, como um modelo de aprendizagem não paramétrico e baseado em instâncias, não possui rotinas de otimização e/ou ajuste de parâmetros durante a etapa de treinamento. O valor da constante *k*, por sua vez, é o único hiper-parâmetro deste classificador que é fornecido pelo usuário.

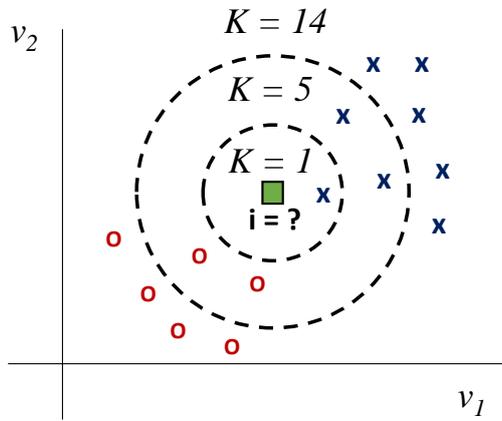


Figura 3.7: Exemplo de uso do modelo *k*NN para definição da classe pertencente à observação *i*, de acordo com os *K* padrões mais próximos (para *k* iguais a 1, 5 ou 14).

Dada uma nova observação \mathbf{x} , o conjunto de treinamento é reordenado pela distância relativa, $Z = \{\|z_1 - x\|, \|z_2 - x\|, \dots, \|z_{n_s} - x\|\}$, e este conjunto é truncado para ser de tamanho *k*. Os respectivos rótulos y_1, y_2, \dots, y_{n_s} são então agregados num vetor de frequência \mathbf{f}_k para cada classe *k*, e a previsão final é produzida por maioria de votos, $\hat{y}_i = \max_k f_k$. Normalmente, a distância euclidiana, $D(\mathbf{m}, \mathbf{n})$, também conhecida como norma *L2*, é usada como função de distância. Ela pode ser expressa matematicamente por:

$$D(\mathbf{m}, \mathbf{n}) = \sqrt{\sum_{i=1}^v (m_i - n_i)^2}, \quad (3.58)$$

onde \mathbf{m} e \mathbf{n} são os vetores a serem comparados e *v* representa o número de características presentes na base de dados. Uma aplicação bem sucedida do *k*-NN para

avaliação não destrutiva é apresentada em [97]. A Figura 3.7, por sua vez, exemplifica, a escolha da classe do vetor de observação \mathbf{i} , de acordo com diferentes valores definidos para k .

3.2.3 Máquinas de Vetores de Suporte (SVM)

A máquina de Vetor de Suporte (em inglês, *Support Vector Machine* - SVM) é uma técnica de aprendizagem de máquina introduzida em 1995 [98] que tem sido empregada com sucesso em vários campos de pesquisa e aplicações industriais [99, 100, 101]. O seu principal objetivo é definir uma fronteira de decisão ótima que melhor separe as classes avaliadas. Isto é alcançado através da determinação de um hiperplano de separação entre as classes que minimize a probabilidade de erro de um determinado conjunto de treinamento.

O algoritmo SVM transforma o espaço de características em um espaço de ordem mais elevada através do emprego de *funções de núcleo* - também conhecidas como funções de *kernel* [98]. A Figura 3.8 apresenta a construção geométrica de um hiperplano ótimo para um espaço de entrada bidimensional. Para estes casos, a “margem de separação” (separação entre o hiperplano e o ponto mais próximo) é definida por um parâmetro ρ_0 [77].

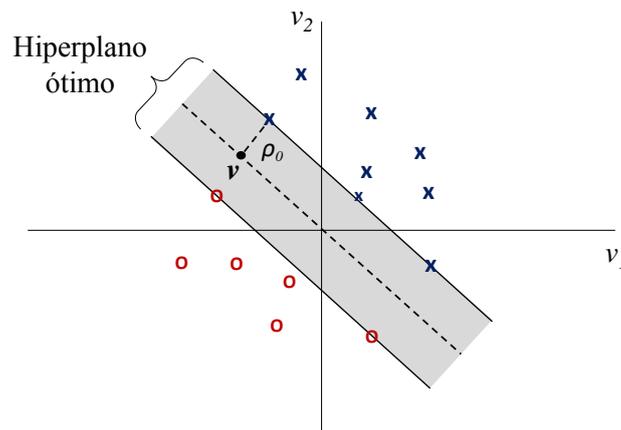


Figura 3.8: Ilustração de um hiperplano bidimensional ótimo para valores linearmente separáveis.

Neste algoritmo, é considerado um conjunto de treinamento definido como \mathbf{z}_i , que representa um vetor de características dimensionais n_f , com amostras n_s . Define-

se $y_i \in \{-1, +1\}$ como valores-alvo indicando uma i -ésima classe. A solução é obtida encontrando os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^{n_s}$ que minimizam a função objetivo

$$Q(\alpha) = \frac{1}{2} - \sum_{i=1}^{n_s} \alpha_i + \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \alpha_i \alpha_j y_i y_j K(\mathbf{z}_i, \mathbf{z}_j), \quad (3.59)$$

sujeitas a

$$\sum_{i=1}^{n_s} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq J \quad \text{para } i = 1, 2, \dots, n_s, \quad (3.60)$$

onde J é uma constante positiva dada pelo usuário, e $K(\cdot, \cdot)$ é uma função de *kernel* parcialmente positiva-definida [77]. Se $\{\alpha_i\}_{i=1}^{n_s}$ é uma solução ótima de Equações (3.59) e (3.60), a função de decisão é obtida como

$$\beta_{SVM}(\mathbf{z}) = \sum_{i=1}^{n_s} \alpha_i y_i K(\mathbf{z}_i, \mathbf{z}) + b, \quad (3.61)$$

em que b é um parâmetro de viés (*bias*). Um novo ponto de dados é discriminado como uma das classes de acordo com o valor de $\beta_{SVM}(\mathbf{z})$. Vale mencionar que a versão original deste método está restrita somente à resolução de problemas de classificação binária [98], contudo existem atualmente diferentes adaptações deste algoritmo direcionadas à classificação multi-classes [39] e também para tarefas de regressão [102].

No que diz respeito à problemas de classificação de conjuntos de dados não linearmente separáveis, os algoritmos SVM também podem ser modelados para incorporar diversos tipos de não-linearidades, sendo estas relacionadas com a implementação de diferentes funções de núcleo - $K(\mathbf{z}_i, \mathbf{z}_j)$ [103]. Para este propósito, além da conhecida função de *Kernel* do tipo linear, representada como $K(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j$, outras funções também podem ser aplicadas, a exemplo da polinomial, que pode ser obtida por

$$K(\mathbf{z}_i, \mathbf{z}_j) = (\gamma \mathbf{z}_i^T \mathbf{z}_j + C)^d, \quad \gamma > 0. \quad (3.62)$$

Outros exemplos são a função de base radial (em inglês, *Radial Basis Function* - RBF), que pode ser expressa como [104]

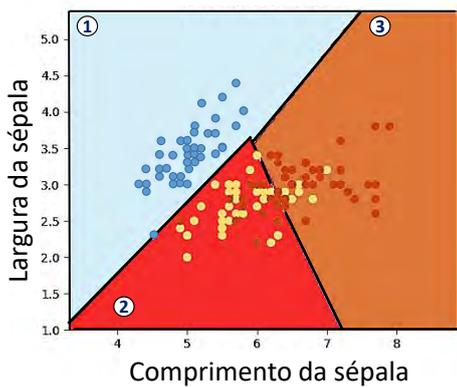
$$K(\mathbf{z}_i, \mathbf{z}_j) = \exp(-\gamma \|\mathbf{z}_i - \mathbf{z}_j\|^2), \quad \gamma > 0; \quad (3.63)$$

e a função sigmoide, também conhecida como função de núcleo de tangente hiperbólica, definida por

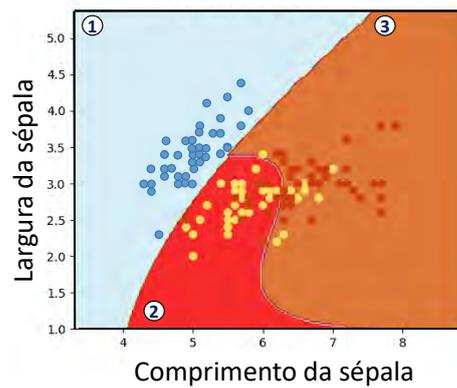
$$K(\mathbf{z}_i, \mathbf{z}_j) = \tanh(\gamma \mathbf{z}_i^T \mathbf{z}_j + r). \quad (3.64)$$

Os termos γ , d e r podem ser entendidos como hiper-parâmetros das funções de *Kernel*, que geralmente são ajustados pelo usuário de acordo as características do *dataset* trabalhado [105]. A constante $C \geq 0$, por sua vez, representa um parâmetro independente da função de núcleo, que está relacionada à compensação do impacto de termos de ordem superior em relação aos termos de ordem mais baixa [106].

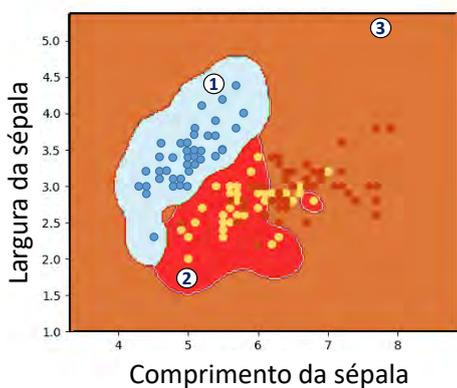
Com o objetivo de ilustrar a influência das funções de *Kernel* nos comportamentos dos modelos SVM, a Figura 3.9 apresenta alguns exemplos de regiões de decisão, que foram produzidas de acordo com o tipo de $K(\mathbf{z}_i, \mathbf{z}_j)$ selecionado para as simulações computacionais. Os quatro gráficos apresentados nos itens da referida figura, de 3.9a até 3.9d, atestam, de forma visual, alterações geométricas significativas nas regiões de decisão, de acordo com as não-linearidades características de suas respectivas funções de núcleo.



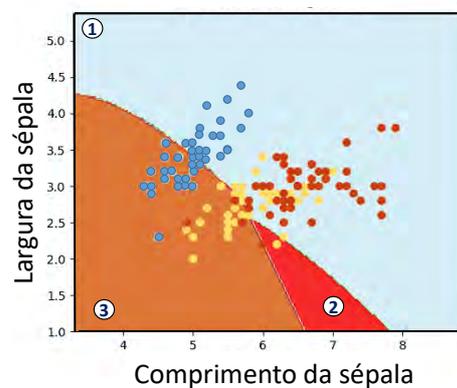
(a) *Kernel* linear.



(b) *Kernel* polinomial.



(c) *Kernel* RBF.



(d) *Kernel* Sigmoidal.

Figura 3.9: Regiões de decisão do algoritmo SVM para diferentes tipos de *Kernels*.

Para a realização das simulações computacionais apresentadas na Figura 3.9, foi utilizado o conjunto de dados *Iris Data Set* [107]. Este *dataset* possui apenas quatro características (comprimento e largura de sépalas e pétalas), com 50 amostras para cada uma das três espécies de íris (setosa, virgínica e versicolor). Para facilitar a visualização do problema (composição de gráfico bidimensional), apenas 2 parâmetros do conjunto de dados foram considerados para constituir os vetores de entrada dos modelos SVM - largura e comprimento das sépalas. Um exemplo de aplicação bem sucedida do algoritmo SVM para classificação multi-classes em configurações híbridas de modelos inteligentes é apresentada em [34].

3.2.4 Floresta Aleatória (*Random Forest*)

O algoritmo floresta aleatória (em inglês, *Random Forest* - RF) é definido como um classificador inteligente construído a partir de uma coleção de árvores de decisão [75]. É um conceito de árvores de regressão, induzido por um método de amostragem do tipo *bootstrap* [40] para um conjunto de dados de treinamento.

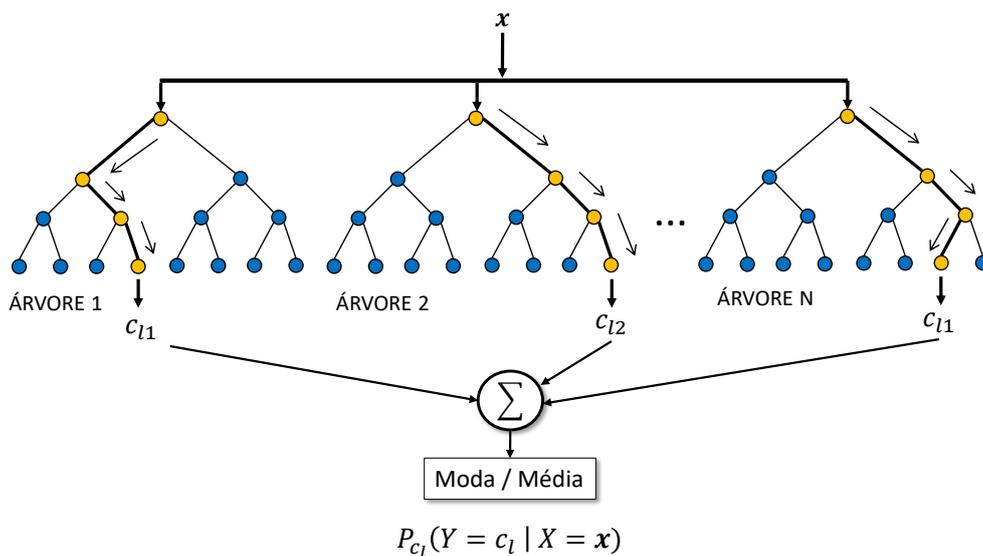


Figura 3.10: Representação simplificada de um conjunto de árvores de decisão que constituem o classificador *Random Forest*.

Nesta metodologia, usam-se descritores aleatórios selecionados no processo de indução de árvores. A floresta aleatória também pode ser entendida como um algoritmo de montagem ou de empacotamento (em inglês, *bagging*), utilizado para calcular a média de ruído e melhorar a redução da variância através da redução da

correlação entre árvores [108].

A principal característica do algoritmo floresta aleatória é que, para a k -ésima árvore em uma coleção, um vetor aleatório Θ_k é gerado, com independência estatística de outros vetores de árvores anteriores, $\Theta_1, \Theta_2, \dots, \Theta_{k-1}$. Todos possuem iguais distribuição de probabilidade. Uma árvore é gerada com o conjunto de treinamento e o vetor Θ_k , resultando em um classificador específico para um determinado vetor de entrada, \mathbf{x} .

Em um momento posterior à geração de um determinado número de árvores de decisão, conforme foi previamente descrito, calcula-se a classe ou valor mais adequado para um determinado vetor de entrada. Considerando o emprego deste algoritmo em tarefas de classificação, é comum realizar o cálculo da Moda, M_o , no conjunto de saídas das árvores de decisão [34]. Neste procedimento, é feito o somatório das saídas obtidas nestas árvores e determina-se a classe mais “popular” como a resposta global do classificador. Para tarefas de regressão, por sua vez, pode-se adotar, como resposta do modelo, o cálculo do valor médio dos dados de saída das árvores de decisão [67]. A Figura 3.10 ilustra, de forma simplificada, um conjunto de árvores de decisão e o processamento de sinal para a definição da classe/valor, $P_{c_l}(y = c_l | X = \mathbf{x})$, de uma determinada entrada.

3.2.5 Rede Bayesiana (*Bayes Net*)

A Rede Bayesiana, também chamada, em inglês, de *Bayesian Network* ou *Bayes Net*, pode ser entendida como um modelo que utiliza teoria dos grafos e distribuição de probabilidades para realizar inferências [41]. De forma mais específica, o algoritmo *Bayes Net* produz a probabilidade de que uma nova observação \mathbf{x} pertença a cada classe k ,

$$P(Y = k | \mathbf{x}) \tag{3.65}$$

Esta probabilidade *a posteriori* é calculada a partir do Teorema de Bayes,

$$P(Y = k | \mathbf{x}) = \frac{P(Y = k)P(\mathbf{x} | Y = k)}{P(\mathbf{x})}. \tag{3.66}$$

Todos estes termos são aprendidos com o conjunto de treino \mathbf{z} . O denominador é apenas um fator de escala. A variável $P(Y = k)$ representa a probabilidade *a priori* de que a hipótese é verdadeira com base no que é conhecido da população, enquanto

$P(\mathbf{x} | Y = k)$ é a probabilidade condicional, que determina o quão compatível a evidência é com a hipótese dada.

A Figura 3.11 apresenta um exemplo simplificado de grafo acíclico dirigido (conhecido como DAG - *Directed Acyclic Graph*) pertencente a uma rede bayesiana, que possui apenas quatro variáveis ou vértices - também intitulados como nós. Nesta imagem, as variáveis (A , B , C e D) possuem informações probabilísticas de cada evento associado e os arcos, de forma integrada, determinam a relação de dependências entre estes vértices. O nó também pode possuir independência condicional, quando é independente de seus predecessores, a exemplo de A e B , ou estar atribuído a uma probabilidade condicionada, como nos casos de C e D , que possuem relação causal com “nós pais” posicionados a montante das setas dos arcos.

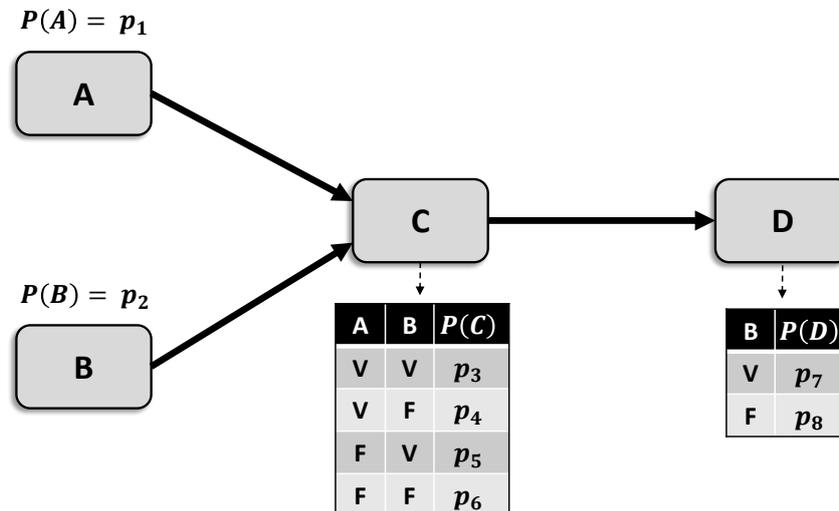


Figura 3.11: Exemplo simplificado de grafo acíclico dirigido (DAG) de uma rede bayesiana.

Para especificar os parâmetros da rede *bayesiana*, torna-se necessário a obtenção dos valores associados às probabilidades de cada evento, representados por $p_1 \dots p_8$, que compõem Tabela de Conjunção de Probabilidades (TCP) do universo do problema. Estas probabilidades, por sua vez, podem ser descobertas através das relações de dependência entre as variáveis, onde cada uma delas é dependente do nó de vértice pai. Uma vez definida a topologia da rede, basta especificar as probabilidades dos nós que participam em dependências diretas, e utilizar as mesmas para computar as demais probabilidades desejadas. No caso do DAG presente na Figura 3.11, temos

que a especificação dessa rede pode ser dada por:

$$P(A, B, C, D) = P(A)P(B)P(C|A, B)P(D|C), \quad (3.67)$$

onde $P(A)$ e $P(B)$ representam apenas um parâmetro de probabilidade. Enquanto $P(C|A, B)$, por sua vez, retrata quatro parâmetros, sendo eles: $P(C|A \wedge B)$, $P(C|A \wedge \neg B)$, $P(C|\neg A \wedge B)$ e $P(C|\neg A \wedge \neg B)$. E, por último, $P(D|C)$ depende de $P(D|C)$ e $P(D|\neg C)$. Em termos de aplicação, um estudo comparativo do desempenho de classificação da rede bayesiana em relação a outros algoritmos clássicos de aprendizado de máquina pode ser encontrado na referência [34].

3.2.6 Modelo Oculto de Markov (HMM)

O modelo Oculto de Markov (em inglês, *Hidden Markov Model* - HMM) é um modelo estocástico que permite a especificação da probabilidade de uma sequência de eventos que não é diretamente observável. De acordo com Gales e Young [42], o HMM consiste de dois processos fundamentais. O primeiro deles é chamado de cadeia de Markov, que é caracterizada pelos estados e pela probabilidades de transição. Os estados dessa cadeia não são externamente visíveis, portanto são denominados como “estados ocultos”. No segundo processo, são produzidas emissões sequenciais observáveis dependentes da distribuição de probabilidade, que estão associadas aos estados ocultos.

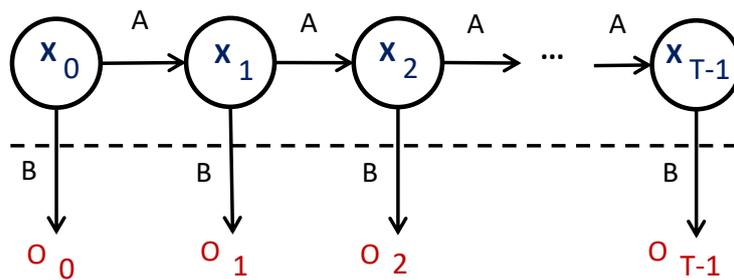


Figura 3.12: Representação genérica do Modelo Ocultos de Markov (HMMs).

Um HMM genérico pode ser visto na Figura 3.12. As notações presentes na figura são definidas como: Sequência Oculta de Estados da HMM (X_i), Matriz de Probabilidade de Transição de Estados (\mathbf{A}), Matriz de Probabilidade das Observações (\mathbf{B}), Sequência de Observações O e o Comprimento da Sequência de Observações

(T). O processo de Markov, que é oculto acima da linha pontilhada, é determinado pelo estado atual de \mathbf{A} . Neste modelo, somente é possível visualizar as variáveis de observação O_i , que são relacionadas aos estados ocultos de Markov por meio de \mathbf{B} . Segundo Bobulsk [109], existem dois problemas fundamentais que devem ser resolvidos para se construir um sistema de reconhecimento de padrões com HMMs:

- 1 Dada uma série de observações $O = (O_1, O_2, \dots, O_T)$ e o modelo $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, deseja-se calcular de maneira eficiente o valor de $P(O|\lambda)$.
- 2 Dada uma série de observações $O = (O_1, O_2, \dots, O_T)$, deseja-se estimar os parâmetros do modelo $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ que maximiza $P(O|\lambda)$.

Para resolução do problema 1, existe um algoritmo bastante conhecido, chamado de *Forward-Backward* [110]. Para o problema 2, pode-se empregar o algoritmo de Baum-Welch [111] e [112]. A HMM requer 3 medidas de probabilidades definidas \mathbf{A} , \mathbf{B} , e π , onde π é distribuição de probabilidades inicial dos estados e a notação $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ é comumente utilizada para indicar o conjunto de parâmetros do modelo [112]. Um exemplo de aplicação de cadeias ocultas de Markov direcionadas à classificação de sequências de observações de problemas multi-classes é apresentado em [32].

3.3 Processamento Digital de Imagens (PDI)

O Processamento Digital de Imagens (PDI) pode ser compreendido como um conjunto de técnicas computacionais direcionadas às tarefas de aquisição, representação e transformação de imagens digitais. De acordo com Gonzalez e Woods [113], o PDI consiste em uma modalidade de processamento digital de sinais cuja entrada é uma imagem, representada como uma função matemática bidimensional, e o resultado de saída é produzido de acordo com o escopo da aplicação trabalhada. Nestes casos, a saída desejada pode ser uma imagem alterada digitalmente [60] ou outro tipo de informação útil para o domínio do problema, a exemplo de regiões segmentadas [31] ou atributos associados às imagens [32, 56]. Atualmente, o processamento digital de imagens é considerado como uma área de investigação bastante diversificada e em amplo crescimento, situando-se em uma posição estratégica dentro dos campos de engenharia e de informática.

Considerando as principais aplicações inerentes à área de PDI, é possível propor duas categorias mais amplas para definir os estágios finais de processamento da informação [113], nomeadamente: (a) métodos cujas saídas são constituídas por imagens e (b) rotinas computacionais cujas saídas são constituídas por parâmetros/atributos que não compõem novas imagens. Estas duas abordagens viabilizam o emprego das técnicas de PDI em diferentes tipos aplicações, a exemplo de tarefas de melhoria e restauração de imagens [31] e de rotinas direcionadas à representação e reconhecimento de padrões [32].

A Figura 3.13, por sua vez, ilustra um diagrama típico que contempla as principais tarefas pertencentes à área processamento digital de imagens. Para fins de simplificação do fluxo de informação do referido diagrama, foi estabelecida uma sequência preferencial de atividades interconectadas, tendo início no *domínio do problema* investigado até a sua finalização na tarefa de *reconhecimento de objetos*. Vale salientar que podem ocorrer alterações na referida ordem e/ou nos tipos de métodos implementados, tendo em vista que as características inerentes às abordagens em PDI usualmente são dependentes de objetivos específicos de cada aplicação [60].

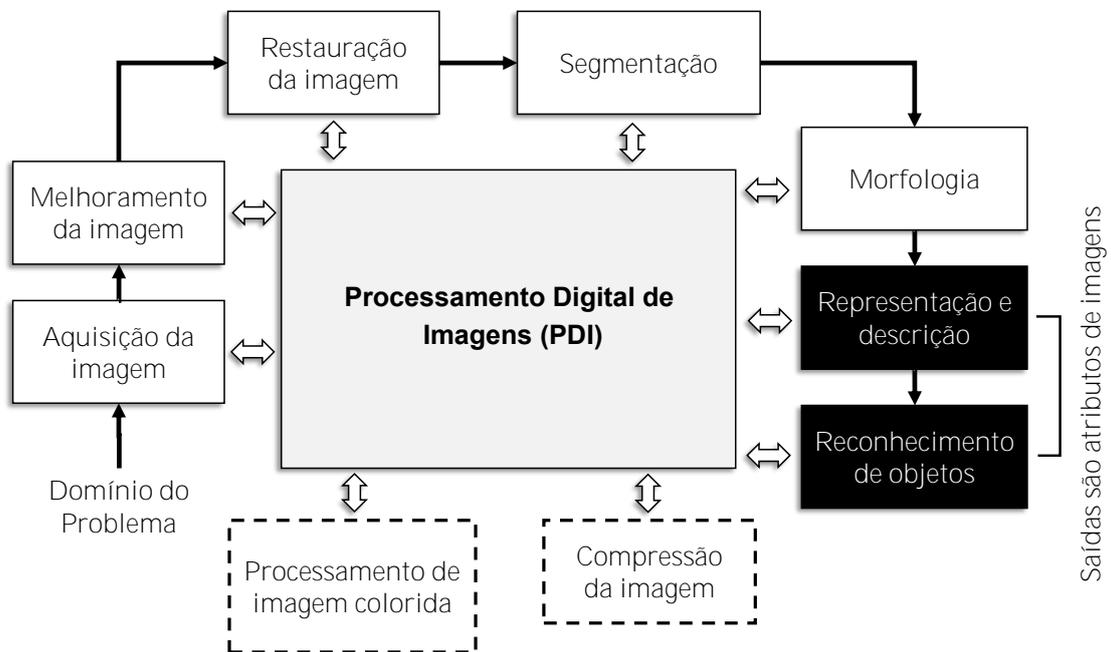


Figura 3.13: Tarefas fundamentais pertencentes a área de Processamento Digital de Imagens (PDI). Adaptado de [113].

3.3.1 Tarefas Fundamentais em PDI

Avaliando o extenso diagrama apresentado na Figura 3.13, pode-se inferir que os sistemas de processamento digital de imagens usualmente não são aplicações simples de serem implementadas. Atualmente, a área de PDI conta com um campo de atuação bastante abrangente e envolve um número considerável de métodos, cada um deles com uma finalidade específica e possíveis interconexões. Deste modo, com o intuito de facilitar a compreensão das técnicas de processamento de imagens implementadas como atividades complementares a esta pesquisa, serão apresentados, de forma simplificada, os principais métodos pertencentes a área de PDI:

Aquisição de imagem: Este estágio é constituído pelas tarefas de geração, coleta ou seleção de imagens digitais que possuem características visuais adequadas ao domínio do problema trabalhado. Considerando os dispositivos de aquisição de imagens, costuma-se empregar uma grande diversidade de elementos sensores, classificados de acordo com as respectivas sensibilidades aos diferentes tipos de fontes de energia. Para estes casos, as fontes mais usuais encontram-se no espectro eletromagnético visível e nas bandas de infravermelho, ultravioleta, raios X, dentre outras. Como estágios complementares às tarefas de aquisição, é possível incluir etapas de pré-processamento (a exemplo da rotina de redimensionamento de imagens), além de tarefas de compressão, armazenamento ou exibição de imagens.

Melhoramento da imagem: Método também conhecido como *realce de imagem*, pode ser interpretado como um conjunto de técnicas de manipulação de imagens digitais de modo que o resultado, se comparado com a imagem original, apresenta características mais adequadas para uma determinada modalidade de exibição ou para atender à demandas específicas de uma aplicação. Como exemplo de técnicas de melhoramento de imagens, é possível citar as rotinas de remoção de ruídos, alteração de brilho e realce de objetos. Os métodos inerentes a esta área de PDI apresentam características de desempenho tipicamente qualitativas, haja vista que os mesmos atendem a critérios de avaliação subjetivos impostos por um determinado observador.

Restauração da imagem: Este método também é empregado para obtenção de ganhos na qualidade visual de imagens digitais, sendo indicado principalmente para os casos em que as imagens estão corrompidas e/ou com a presença de ruídos. Contrastando com as aplicações de melhoramento de imagens, as técnicas de restauração possuem parâmetros de desempenho com características majoritariamente quantitativas, tendo em vista que estes métodos se baseiam em modelos matemáticos e probabilísticos que calculam o nível de degradação das imagens avaliadas [113].

Segmentação: É o processo de divisão de uma imagem digital em múltiplos sub-grupos ou regiões de interesse. De forma simplificada, a rotina de segmentação produz diferentes conjuntos ou aglomerados de *pixels*, também conhecidos como objetos de imagem. O objetivo dessa rotina reside na transformação de uma imagem digital em algo mais significativo ou simples de ser interpretado pelo computador [31]. As técnicas de segmentação serão abordadas com um maior nível de detalhamento na Subseção 3.8.2.

Morfologia: O termo morfologia foi originalmente associado a um dos ramos da biologia que trata das formas e estruturas de plantas e de animais. Na área de processamento digital de imagens, este termo é empregado no contexto de morfologia matemática e atua como uma ferramenta computacional direcionada à extração e análise de componentes de uma imagem digital. Tais informações são úteis para a representação e/ou descrição da forma de um determinado conjunto de *pixels* presentes nas imagens, a exemplo de esqueletos, fronteiras e fechos convexos (em inglês, *convex hull*) [113]. A linguagem matemática usualmente adotada para o processamento morfológico é a teoria de conjuntos, e as suas operações fundamentais consistem nas técnicas de erosão e dilatação, empregadas principalmente nas rotinas de alteração de componentes de imagens [114].

Representação e descrição: A técnica de representação de imagens digitais está usualmente relacionada ao resultados de uma rotina de segmentação e é dependente da forma como se deseja representar os objetos. Como características de mérito desta técnica, é possível obter a forma externa de objetos (informações de contorno) ou propriedades internas dos mesmos (informações de região), a exemplo das ca-

racterísticas de textura e esqueleto, dentro outras. No que se refere ao método de descrição de imagens, é comum selecionar características ou atributos presentes nos objetos que irão resultar em alguma informação quantitativa de interesse. Deste modo, de posse de informações que descrevem regiões específicas nas imagens, a exemplo de um conjunto de atributos geométricos, viabiliza-se o reconhecimento de padrões ou a discriminação de diferentes classes de objetos [32].

Reconhecimento de objetos: Este método pode ser definido como o processo que atribui um rótulo de classe a um determinado objeto de acordo com as informações obtidas nos estágio de descrição. De acordo com Gonzales e Woods [113], esta metodologia de reconhecimento de padrões pode ser dividida em duas categorias principais, denominadas como decisões teóricas e estruturais. Para o primeiro caso, o processo de classificação ocorre através do emprego de descritores quantitativos, a exemplo de parâmetros de comprimento, extensão e textura. Para o caso das decisões estruturais, abordam-se os padrões cuja descrição é obtida através de abordagens com características qualitativas, a exemplo dos descritores relacionais.

Compressão da imagem: Este processo pode ser definido como uma modalidade de compressão de dados aplicada à imagens digitais, cujos principais objetivos residem na redução de custos computacionais associados às tarefas de armazenamento e/ou transmissão de dados. Neste método, a imagem de saída frequentemente ocupa um espaço de armazenamento reduzido, mas com características visuais similares à versão original da imagem.

Processamento de imagem colorida: Trata das rotinas de processamento de imagens digitais coloridas e dos diferentes modelos de cores existentes, a exemplo dos sistemas RGB e CMYK [60]. Nas imagens digitais, a cor é um descritor poderoso que pode facilitar a extração características de um objeto presente na imagem. Estas tarefas podem ser associadas a duas áreas distintas, tidas como processamento imagens coloridas e de pseudo-cores. Para o primeiro caso, adota-se um conjunto de cores diversificado, provenientes de sensores de cores reais, a exemplo das câmeras digitais e *scanners*. No caso das pseudo-cores, o atributo avaliado é um tom de cor para uma determinada intensidade monocromática ou uma faixa de intensida-

des [113].

3.4 Aprendizagem Profunda

A aprendizagem profunda (do inglês, *deep learning*) pode ser entendida como uma subárea em ascensão do campo de inteligência computacional. Atualmente, existem diversas linhas de pensamento direcionadas a diferenciar os algoritmos de aprendizagem profunda em relação aos tradicionais modelos “canônicos” de redes neurais artificiais. Percebe-se que ainda não existe uma uniformidade nestas interpretações, contudo é possível citar características de mérito inerentes ao aprendizado profundo, a exemplo de: múltiplas camadas e diferentes estágios de transformação; modalidades mais complexas de conexão entre as camadas e, principalmente, a capacidade de aprender abstrações de alto nível nos dados utilizando arquiteturas organizadas de forma hierárquica [90, 78, 115].

No domínio da Inteligência Artificial (IA), os modelos computacionais de aprendizagem profunda vêm apresentando um papel de destaque em relação às técnicas convencionais de aprendizado de máquina. Nesta última década, por exemplo, a aprendizagem profunda proporcionou ganhos significativos de desempenho em diversas linhas de atuação, tais como: robótica [116, 117]; visão computacional [22, 24, 25, 26, 27]; processamento de voz e linguagem natural [118]; automação de processos (também conhecida como AutoML - *Automated Machine Learning*) [119]; transferência de conhecimento [120] e análise semântica [121]. Estes modelos também se tornaram protagonistas no desenvolvimento de aplicações inteligentes [33, 16] e de sistemas visuais avançados, a exemplo dos carros autônomos [28] e algoritmos de interpretação de imagens [29, 30].

As Figuras 3.14 e 3.15 apresentam gráficos com recortes das principais linhas de atuação presentes nos *Ciclos de Gartner*, para a área de inteligência artificial, referentes aos anos de 2019 e 2020, respectivamente [122, 123]. De forma simplificada, Gartner é uma empresa de pesquisa e consultoria que publica anualmente representações gráficas (ciclos de Gartner) que tratam dos níveis de maturidade de aplicações e tecnologias que fazem parte do fluxo de inovações e tendências da área de IA, considerando estudos conduzidos em ambientes acadêmicos e empresariais [124]. Estes

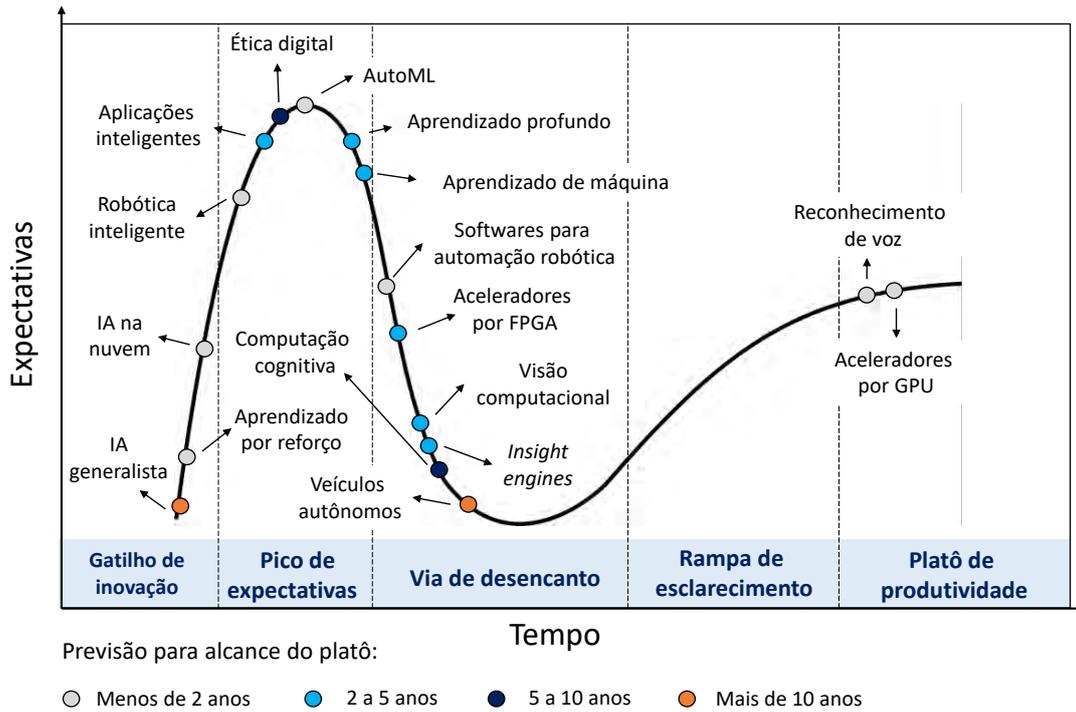


Figura 3.14: Ciclo de Gartner para Inteligência Artificial, 2019. Adaptado de [122].

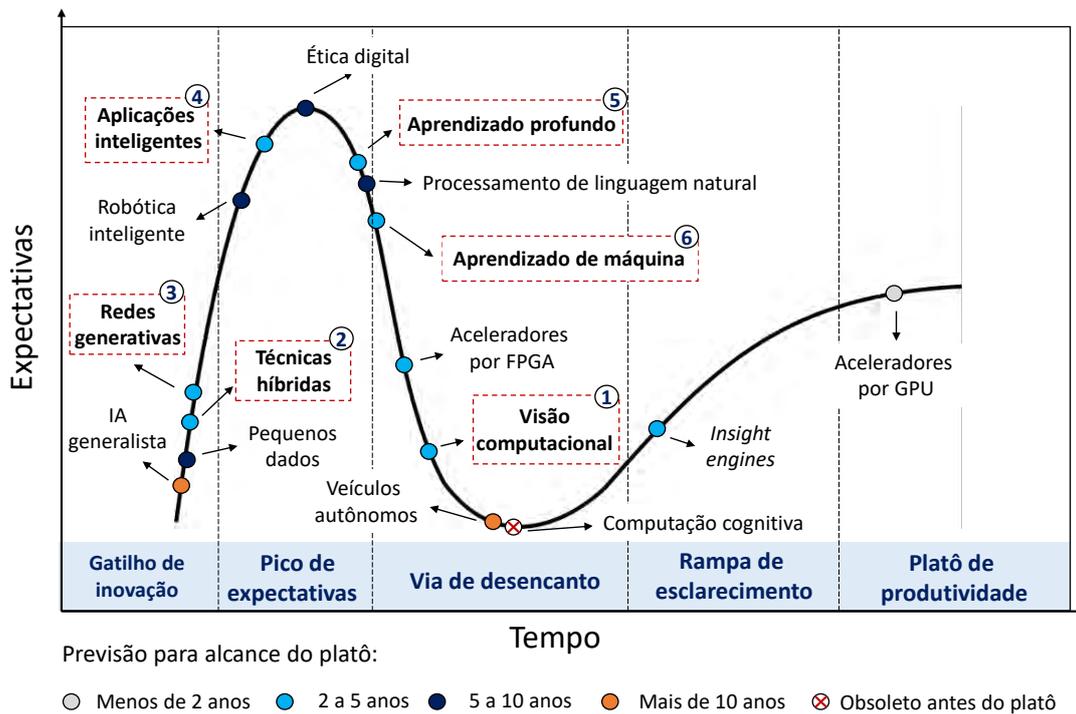


Figura 3.15: Ciclo de Gartner para Inteligência Artificial, 2020. Adaptado de [123].

gráficos, por sua vez, apresentam curvas não lineares que relacionam os níveis de expectativa da sociedade/mercado, para um determinado tipo de tecnologia, com o tempo transcorrido de desenvolvimento - desde sua prova de conceito (gatilho de

inovação) até o alcance de amplo nível de aplicabilidade e relevância no mercado (platô de produtividade) [125].

Os autores desta ferramenta de avaliação também apresentam previsões dos intervalos de tempo necessários para consolidação de cada tipo de tecnologia inteligente. Vale salientar que os ciclos de Gartner são estudos fundamentalmente empíricos e já demonstraram alguns tipos de incongruências em suas previsões [126], contudo este modelo vem se revelando, ao longo dos anos, como uma ferramenta satisfatória para planejamentos estratégicos, e também suscita um nível de interesse crescente por parte de profissionais em ambientes corporativos e, mais recentemente, de membros da comunidade científica.

A partir das informações apresentadas nos dois gráficos pertencentes ao Ciclo de Gartner, é possível afirmar que a área de inteligência artificial pode ser considerada, no momento atual, como um campo dinâmico, diversificado e com muitas potencialidades para criação de inovações. Em um curto intervalo de tempo, compreendido entre os anos de 2019 e 2020, novas tecnologias inteligentes emergiram, a exemplo das redes generativas [21, 35] e das técnicas híbridas [34], enquanto outras metodologias já estão sendo avaliadas como potencialmente obsoletas, antes mesmo do alcance do platô de produtividade, como é o caso da computação cognitiva [127].

Considerando algumas das principais metodologias inteligentes envolvidas nesta pesquisa de Doutorado (demarcadas com os índices de 1 a 6 na Figura 3.15), pode-se inferir também que existem atualmente altos níveis de expectativa da comunidade científica e empresarial quanto ao desenvolvimento de pesquisas e soluções de engenharia que implementam o aprendizado profundo para otimização de desempenho de potenciais sistemas inteligentes. Como informação adicional, as *aplicações inteligentes*, bem como as *técnica híbridas* e as tecnologias relacionadas ao campo de visão computacional, ainda são consideradas linhas de pesquisa pertencentes ao *estado da arte* da engenharia e estão evoluindo para alcançar seus respectivos platôs de produtividade num futuro próximo.

Além dos benefícios já mencionados, é possível citar outras duas importantes razões para o aumento da ocorrência de pesquisas e aplicações de engenharia relacionadas a aprendizagem profunda nos dias atuais [79, 90]:

1. *Grande quantidade de dados digitais brutos e datasets disponíveis para treina-*

mento de modelos inteligentes. O recente desenvolvimento de redes de sensoriamento e tecnologias de comunicação, a exemplo da internet, viabilizaram a coleta de grandes quantidades de dados digitais, também conhecidos como *big data* [128]. Atualmente, a *big data* proporciona grandes oportunidades e elevado potencial de transformação em diversos setores, incluindo o monitoramento urbano [129], bioinformática [130], comércio eletrônico, controle industrial, medicina inteligente e análise do comportamento humano [128, 131]. No entanto, ainda existem diversos desafios técnicos para o aproveitamento desses dados e análise das informações. À medida que os *datasets* de treinamento vão ficando maiores, a aprendizagem profunda vai proporcionalmente ocupando um papel de destaque no fornecimento de grandes soluções analíticas [129, 132].

2. *Aumento expressivo na capacidade de processamento e no nível de diversidade de hardware e software direcionados para treinamento e simulação de modelos inteligentes.* A implementação de algoritmos de aprendizagem profunda usualmente envolve uma grande quantidade de parâmetros ajustáveis e diferentes tipos de operações matemáticas, resultando em elevados custos computacionais de processamento, principalmente durante a etapa de treinamento dos modelos [133]. Deste modo, com foco na melhoria de desempenho e diversificação dos tipos de sistemas inteligentes, existe uma tendência crescente por parte da indústria de tecnologia no desenvolvimento e especialização de *hardware* que sejam mais adequados às especificidades de modelos de aprendizagem profunda [134]. Considerando que diversas operações comuns aos algoritmos inteligentes podem ser executadas de forma paralela, através de cálculo tensorial, o surgimento e popularização de *hardware* com alta capacidade de paralelismo, a exemplo das GPUs (*Graphics Processing Unit*) e, mais recentemente, das TPUs (*Tensor Processing Unit*), estão viabilizando o desenvolvimento de modelos de aprendizado profundo maiores e ainda mais sofisticados [135]. Em conjunto a isso, estão sendo realizados avanços contínuos no desenvolvimento de *frameworks*, bibliotecas e algoritmos capazes de realizar cálculos tensoriais complexos na área de IA [136]. Dois exemplos importantes de *frameworks* de aprendizado de máquina muito utilizados atualmente são o *TensorFlow* [137] e

o *PyTorch* [138]. No que diz respeito às bibliotecas de IA relativas às unidades de processamento, tais como a *cuDNN* (*Deep Neural Network library*) [139] e a *ROCm* (*Radeon Open Compute*) [140], pode-se dizer que elas estão viabilizando aumento de desempenho das GPUs na simulação e treinamento dos modelos de aprendizado profundo. No que diz respeito ao desenvolvimento de sistemas embarcados e aplicações móveis, diversos avanços recentes foram proporcionados através de *hardware* menos convencionais no campo de aprendizado de máquina, a exemplo das FPGAs (*Field Programmable Gate Arrays*) e dos ASICs (*Application Specific Integrated Circuits*) [141, 142, 143].

Por fim, pode-se observar, também, que o emprego em massa de técnicas de aprendizado profundo vem superando antigos paradigmas estabelecidos na área de inteligência computacional e estão demonstrando, ao longo desta última década, os melhores desempenhos em competições prestigiadas na área de visão computacional, como é o caso do “*ImageNet Large Scale Visual Recognition Challenge*” (ILSVRC) [91] e “*Microsoft COCO Challenge*” [144].

3.5 Rede Neural Convolutacional (RNC)

A Rede Neural Convolutacional (RNC), também conhecida como rede convolutiva, pode ser entendida como uma classe particular inerente aos métodos de aprendizagem profunda, que apresenta algumas semelhanças em relação às redes neurais artificiais convencionais [90]. De acordo com Haykin [77], uma rede neural convolutacional pode ser definida como uma classe especial de RNA Perceptron de Múltiplas Camadas (*Multilayer Perceptron - MLP*), que foi concebida especialmente para reconhecer padrões em *datasets* bidimensionais. Já para Goodfellow et al. [79], estes algoritmos podem ser entendidos simplesmente como redes neurais que utilizam a operação de convolução ao invés de multiplicação geral entre matrizes em pelo menos um dos seus estágios de processamento.

Conforme LeCun e Bengio [145] - ambos cientistas da computação que estão entre os pioneiros na condução de estudos sobre algoritmos de aprendizagem profunda e redes neurais convolutivas-, as RNCs apresentam, como características de mérito, alto grau de invariância em relação a translação, alteração de escala, in-

clinação e outras formas de distorção, tornando-se, por exemplo, uma importante ferramenta para tarefas de processamento de imagens. Além disso, de acordo com os mesmos autores, estes algoritmos também possuem propriedades adequadas para serem empregados em dados unidimensionais, a exemplo de sinais de voz.

As redes neurais convolucionais vêm demonstrando nos últimos anos excelente desempenho em diversos tipos de aplicações, especialmente em tarefas relacionadas à visão computacional [146, 147, 148]. Em uma rede convolutiva típica, a planta arquitetural do modelo geralmente apresenta uma série de estágios de processamento da informação, tais como: a *camada convolucional*; a *camada de agrupamento* ou de subamostragem (também chamada, em inglês, de *pooling layer*) e a camada totalmente conectada, comumente intitulada de *camada densa* [76]. Os referidos estágios de processamento desempenham papéis diferentes na RNC, e atuam de forma hierárquica [149]. Na Figura 3.16, é apresentada uma planta arquitetural de uma rede neural convolucional típica, projetada com o objetivo de realizar tarefas de classificação de imagens.

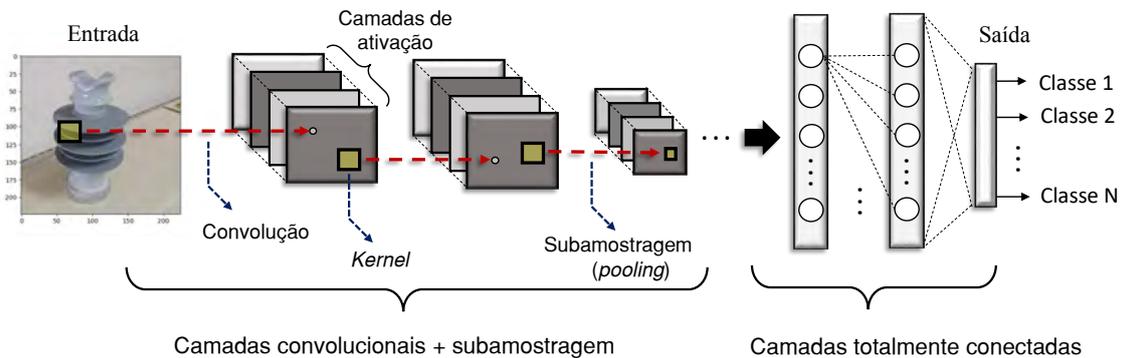


Figura 3.16: Arquitetura típica de uma Rede Neural Convolucional (RNC).

Considerando alguns dos fatores históricos que contribuíram para o desenvolvimento de redes convolutivas empregadas no campo de visão computacional, é possível inferir que estes algoritmos possuem um certo nível de inspiração biológica. O arcabouço teórico que dá lastro a esta afirmação advém especialmente de estudos relacionados com o funcionamento do córtex visual em humanos e em outros animais [150, 151]. Hubel e Wiesel [152, 153, 154], no período compreendido entre 1958 e 1968, apresentaram uma série de trabalhos científicos sobre os campos receptivos de neurônios individuais em gatos e primatas. Eles demonstraram que os padrões

de estímulos de luz mais eficazes para influenciar certas unidades do cérebro, em um determinado nível, podem não ser tão efetivos nos níveis seguintes. De certo modo, as referidas unidades operam de forma semelhante a pequenos filtros receptivos - características que também estão presentes em redes convolutivas [66]. Já em 1991, Felleman e Van Essen [151] apresentaram evidências sobre como o córtex visual pode estar hierarquicamente organizado nos cérebros de humanos e de primatas, o que significa que muitos módulos funcionais possuem ligações do tipo *alimentadas para frente (feedforward)* e *alimentadas para trás (backward)*, compatíveis com uma ordenação global de níveis inferiores para níveis superiores - atributos que também são predominantes em redes convolutivas e outros algoritmos de aprendizado profundo [21].

Retornando para a planta arquitetural presente na Figura 3.16, pode-se observar que existem algumas propriedades que são restritas às redes neurais convolucionais, a exemplo dos estágios de extração e mapeamento de características [145]. Para as rotinas de extração de características, por exemplo, verifica-se que cada neurônio obtém suas entradas sinápticas de um *campo receptivo local* (também chamado de filtro receptivo ou *kernel*) presente na camada anterior. Deste modo, são extraídas características locais dos dados bidimensionais presentes em cada camada da RNC. Uma vez de posse de uma destas características, a sua posição em relação às demais fica aproximadamente preservada.

No que diz respeito ao mapeamento de características, as redes convolutivas, durante o seu fluxo de processamento, produzem um conjunto de *mapas de ativação*, também chamados de mapas de características. Cada camada convolucional da RNC é comumente composta por um número expressivo destes mapas, sendo que cada um deles é gerado sob a forma de um plano, conforme ilustrado na Figura 3.16, e os neurônios são obrigados a dividir os mesmos pesos sinápticos, através de um núcleo retangular com pequenas dimensões. Estas características estruturais inerentes às RNCs vêm apresentando os seguintes efeitos benéficos [77]: redução do número de parâmetros variáveis do modelo, em decorrência da utilização de pesos partilhados, e invariância ao deslocamento da imagem, ocasionada pelo emprego de operações convolutivas ao longo de toda a extensão dos mapas de ativação.

Assim como ocorre com as redes neurais do tipo MLP, a RNC usualmente exe-

cuta o algoritmo de retropropagação de erro (*backpropagation*) durante a rotina de treinamento [136], de forma similar ao que foi apresentado na Subseção 3.2.1. O treinamento da rede convolutiva pode ser separado em dois estágios de processamento de dados. No primeiro estágio, o fluxo de dados da matriz ou tensor de entrada, com os parâmetros (pesos e *bias*) ainda não atualizados, se propaga *alimentado para frente* (*feedforward*), de forma sequencial, camada por camada ao longo de toda a rede. Em seguida, as saídas previstas pelo modelo são usadas para calcular uma função custo de acordo com as diferenças entre as respostas previstas e as respostas desejadas ou saídas alvo [79].

Após o cálculo da função custo, ocorre a segunda fase do treinamento, estágio em que a rede convolutiva é alimentada de forma reversa - rotina também conhecida como *backward*. Neste estágio, calculam-se os gradientes de cada variável da RNC de acordo com a regra da cadeia. Todos estes parâmetros são então atualizados com base nestes gradientes, e são preparados para o próximo cálculo de avanço. Após um determinado número de iterações das fases de *feedforward* e *backward*, a rotina de treinamento pode ser interrompida.

3.5.1 A operação de convolução

A convolução, em sua forma mais geral, pode ser entendida como um operador linear que atua sobre duas funções $f(t)$ e $g(t)$. Ela é o resultado da soma do produto entre essas funções, mas com uma delas revertida e deslocada sobre a outra. A função resultante $h(t)$, deste modo, dependerá da área gerada pela superposição de ambas as funções a partir do deslocamento existente entre elas. Sendo t a variável independente e a o deslocamento, a fórmula da convolução pode ser escrita como [155]:

$$h(t) = \int_{-\infty}^{+\infty} f(a)g(t - a)da. \quad (3.68)$$

Esta operação usualmente recebe as seguintes notações [156, 155]

$$h(t) = f(t) * g(t) \quad (3.69)$$

onde $h(t)$ é a função calculada e o asterisco (*) é utilizado como operador convolucional. Para casos particulares que envolvem redes neurais convolucionais, Goodfellow et al. [79], especifica que $g(t)$ necessita ser uma função de densidade de probabilidade

válida, ou a saída $h(t)$ não será uma média ponderada - característica necessária para a simulação dos modelos computacionais.

Normalmente, quando se trabalha com dados computacionais, o tempo deve ser discretizado em intervalos regulares. Deve-se amostrar a variável de tempo t em múltiplos inteiros de um período de amostragem T . Caso também seja assumido que $f[n]$ e $g[n]$ são definidos apenas para valores de nT , com n inteiro, pode-se definir a convolução discreta como:

$$h[n] = f[n] * g[n] = \sum_{a=-\infty}^{+\infty} f[a]g[n-a]. \quad (3.70)$$

Na terminologia empregada para redes neurais convolucionais, a sequência $f[n]$ é usualmente referida como a entrada. O segundo argumento (neste exemplo, a função $g[n]$) é denominado como filtro receptivo ou *kernel*. A saída da convolução $h[n]$ costuma ser apresentada como mapa de características [78].

Em aplicações de aprendizagem de máquina que envolvem redes convolutivas, a entrada é geralmente uma matriz multidimensional de dados \mathbf{F} , e o *kernel* também é uma matriz multidimensional de parâmetros \mathbf{G} que são adaptados pelo algoritmo de aprendizagem. É comum se referir a essas matrizes multidimensionais como tensores. A forma compacta para estes casos é dada por

$$\mathbf{H}[n] = \mathbf{F}[n] * \mathbf{G}[n], \quad (3.71)$$

onde \mathbf{H} é a matriz multidimensional calculada (apresentada previamente como mapas de características) e o operador $[\cdot] * [\cdot]$ é usado, nestes casos, como operador convolucional bidimensional para a linha x e coluna y .

Como os elementos da entrada e do *kernel* devem ser armazenados separadamente, geralmente se assume que essas sequências são iguais a zero em todos os lugares, menos no conjunto finito de pontos onde estes valores estão armazenados. Isto implica na viabilidade de implementação da convolução sobre um número finito de elementos da matriz [78].

Para melhor entendimento de operações convolucionais realizadas em matrizes, utiliza-se frequentemente convoluções sobre mais de um eixo de cada vez. Por exemplo, usando uma imagem bidimensional \mathbf{F} como entrada, provavelmente também será usado um *Kernel* bidimensional. Definindo-se J e K como dimensões do filtro

receptivo, a equação que representa a convolução será obtida por:

$$\mathbf{H}[x, y] = \mathbf{F}[x, y] * \mathbf{G}[x, y] = \sum_{j=-J}^J \sum_{k=-K}^K \mathbf{F}[j, k] \mathbf{G}[x - j, y - k]. \quad (3.72)$$

Neste caso, a convolução é uma operação comutativa, significa que podemos escrever de forma equivalente:

$$\mathbf{H}[x, y] = \sum_{j=-J}^J \sum_{k=-K}^K \mathbf{F}[x - j, y - k] \mathbf{G}[j, k]. \quad (3.73)$$

Esta última fórmula é mais simples de implementar em uma biblioteca de aprendizado de máquina, porque há menos variação na faixa de valores válidos de j e k , que representam os parâmetros do *kernel*. Como alteração adicional, muitas bibliotecas de redes neurais implementam uma função relacionada, chamada correlação espacial [113], que se assemelha à convolução, mas não reverte os filtros receptivos:

$$\mathbf{H}[x, y] = \sum_{j=-J}^J \sum_{k=-K}^K \mathbf{F}[x + j, y + k] \mathbf{G}[j, k], \quad (3.74)$$

Contudo, para evitar possíveis erros de interpretação do usuário, esta última informação é omitida e a operação (retratada pela equação 3.74) continua sendo chamada de convolução [79].

3.6 Arquitetura da Rede Neural Convolutiva

Para facilitar a compreensão dos estágios de processamento da informação pertencentes às redes convolutivas, a exemplo da representação simplificada presente na Figura 3.16, torna-se necessário investigar o comportamento hierárquico dos diferentes elementos presentes em suas plantas arquiteturas. Deste modo, com objetivo de fornecer o lastro teórico necessário para o entendimento desta classe de algoritmo de aprendizado profundo, serão definidas as principais camadas e operações matemáticas inerentes às redes convolutivas utilizadas no campo de visão computacional.

3.6.1 Camada Convolutiva

A camada convolutiva, que confere o nome à rede convolutiva, realiza a operação de convolução entre tensores de entrada e filtros receptivos do modelo computacional. Ela também é conhecida como a camada de extração de características da

rede convolucional [115]. Quanto à natureza dos seus dados de entrada, a camada convolucional pode processar tanto dados brutos quanto mapas de características provenientes de estágios convolutivos anteriores [149]. A partir da operação de convolução, serão produzidos *mapas de características* que farão parte da planta arquitetural da RNC.

A Figura 3.17 exemplifica a operação de convolução entre um tensor de entrada \mathbf{E} , de dimensões $4 \times 4 \times 3$, e um *kernel* do tipo $2 \times 2 \times 3$. A primeira operação matemática realizada pelo filtro pode ser visualizada nesta mesma imagem. O valor resultante desta operação, indicado como 63, representa o produto escalar entre um conjunto de matrizes de entrada (\mathbf{E}_1 , \mathbf{E}_2 e \mathbf{E}_3) e os respectivos filtros receptivos (\mathbf{F}_1 , \mathbf{F}_2 e \mathbf{F}_3). Considerando o movimento de translação dos filtros [157] (chamado, em inglês, de *stride*) igual a 1, obtém-se como resultado uma matriz de tamanho 3×3 .

Para o caso de uma imagem ou tensor de entrada, a RNC utiliza um ou mais filtros receptivos para convoluir toda a imagem, gerando, por conseguinte, um novo conjunto de matrizes. A matriz resultante é conhecida como mapa de características ou *Mapa de Ativação* (MA). O MA pode ser entendido como um plano bidimensional, contendo diversas características inerentes ao tensor de entrada, que foram filtradas pelo *kernel*. A Figura 3.18 mostra o MA resultante de uma operação de convolução entre um filtro convolutivo e uma imagem colorida - com representação simplificada (não fidedigna) de suas matrizes RGB. É possível perceber nesta imagem que determinadas regiões do tensor de entrada foram realçadas (áreas mais claras), em contraste com outras regiões mais escuras.

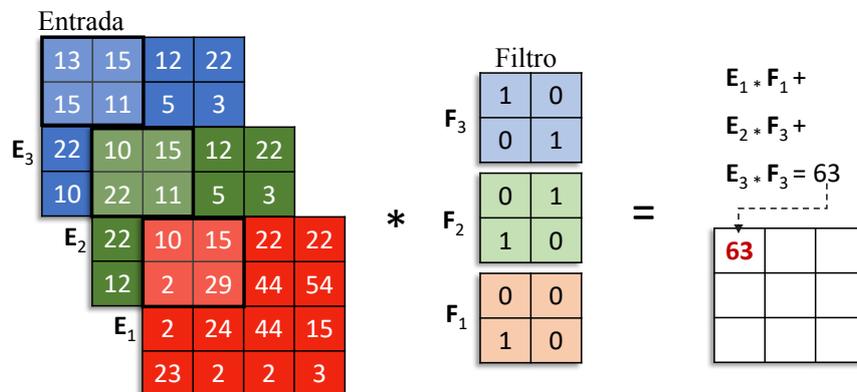


Figura 3.17: Exemplo de produto escalar entre filtros receptivos ($2 \times 2 \times 3$) e região do tensor de entrada de dimensões $4 \times 4 \times 3$.

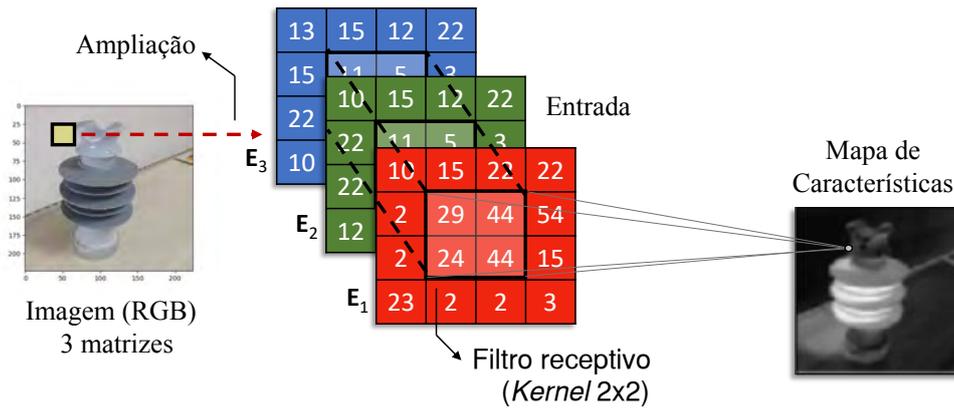


Figura 3.18: Filtros receptivos (*Kernel*) da rede neural convolucional.

De acordo com Zeiler [158], existem três vantagens principais proporcionadas pela operação convolucional: redução do número de parâmetros do modelo em decorrência do compartilhamento de pesos no mesmo mapa de características; a conectividade local aprende correlações entre *pixels* vizinhos e a invariância do método em relação à localização do objeto.

Mapas de Características: Para um conjunto de *kernels* pertencentes a um estágio convolutivo da RNC, a sua resposta será um conjunto de planos ou imagens bidimensionais. Como resultado desta operação, as imagens geradas pelos filtros receptivos comporão um novo tensor de mapas de características, que fará parte da arquitetura da rede convolucional. A partir do referido tensor, ocorrerão novas operações, a exemplo de diferentes estágios convolutivos ou de agrupamento, que serão realizados de forma sequencial no modelo computacional [90].

A Figura 3.19 ilustra o resultado de uma operação de convolução de 16 diferentes filtros receptivos, de dimensões $3 \times 3 \times 3$, em uma imagem colorida de entrada - com especificações dimensionais de $224 \times 224 \times 3$. O resultado dessa operação é um tensor do tipo $224 \times 224 \times 16$, contendo diferentes mapas de ativação. Para simplificar a visualização deste exemplo, os valores numéricos dos neurônios dos filtros receptivos foram substituídos por escalas de cor - quanto mais próximo do vermelho, maior é o valor do parâmetro.

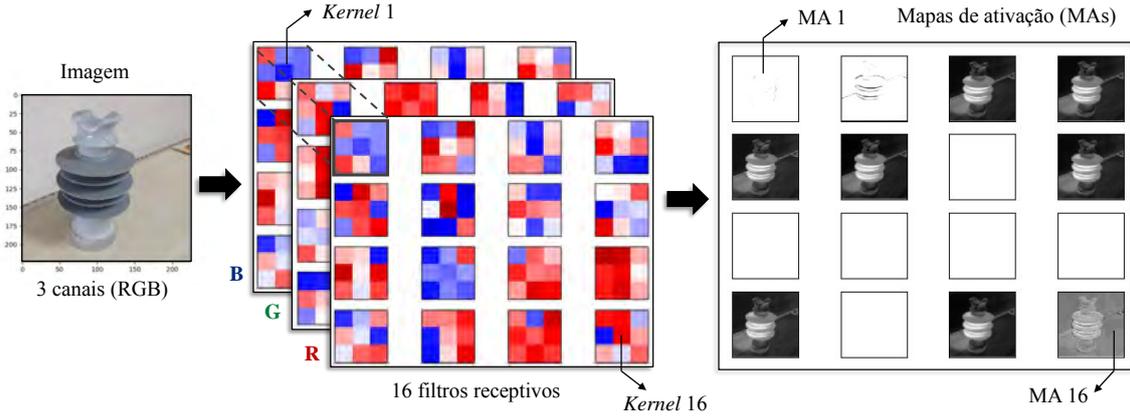


Figura 3.19: Exemplo de um conjunto de mapas de ativação (tensor $224 \times 224 \times 16$) gerados por um estágio convolutivo, com 16 *kernels* do tipo $3 \times 3 \times 3$.

3.6.2 Camada de Agrupamento (*Pooling*)

A Camada de Agrupamento (CA), também conhecida, em inglês, como *pooling layer*, tem como uma de suas principais funções a redução das dimensões dos mapas de características e do número de parâmetros por meio de uma subamostragem. De acordo com LeCun e Bengio [145], esta camada também possui a capacidade de reduzir a sensibilidade da saída do mapa de características a deslocamentos e outras formas de distorção. A CA também demonstra invariância em termos de translação, pois seus cálculos levam em conta a orientação dos *pixels* vizinhos. Nas redes convolutivas típicas, a camada de agrupamento vem geralmente após uma camada convolucional.

As modalidades de *agrupamento médio* (do inglês, *average pooling*) e do *agrupamento máximo* (do inglês, *max pooling*) são as estratégias mais utilizadas para redes convolutivas desenvolvidas para as tarefas de visão computacional.

Segundo Boureau et al. [159], a camada de agrupamento do tipo *max pooling*, $f_m(\mathbf{v})$, aplica uma função janela $u(x, y)$ à matriz ou *patch* de entrada \mathbf{v}_i , de $n \times n$ unidades, e calcula o valor máximo na vizinhança. Obtém-se a equação:

$$f_m(\mathbf{v}) = \max_{n \times n}[\mathbf{v}_i \cdot u(n, n)]. \quad (3.75)$$

Para a camada de agrupamento do tipo *average pooling*, $f_a(\mathbf{v})$, Scherer et al. [160], definem essa função como sendo uma média aritmética da janela de entrada, cuja

equação característica pode ser dada por

$$f_a(\mathbf{v}) = \frac{1}{n \times n} \sum_{i=1}^{n \times n} \mathbf{v}_i. \quad (3.76)$$

Em ambos os casos, o resultado é um mapa de características de menor resolução que a do tensor de entrada. No entanto, Scherer et al. [160] conduziram um estudo comparativo entre essas duas técnicas de agrupamento e constataram que o *max-pooling* pode apresentar algumas vantagens, a exemplo de: proporcionar uma convergência mais rápida na etapa de treinamento; selecionar características invariantes com desempenhos superiores e melhorar a generalização do modelo computacional.

A Figura 3.20 fornece um exemplo de subamostragem para ambos os casos apresentados. Utiliza-se um mapa de características, de dimensões 4×4 , com as operações de *max pooling* e *average pooling* com função janela do tipo 2×2 e *stride* 2.

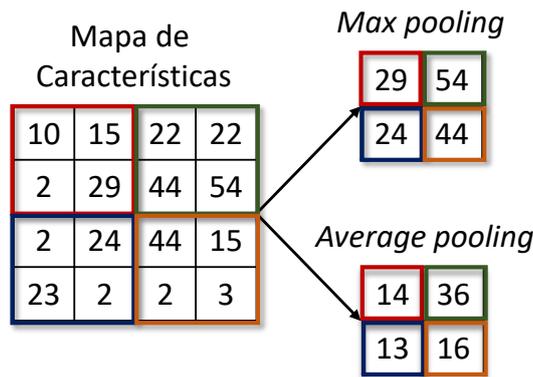


Figura 3.20: Camadas de subamostragem - *max pooling* e *average pooling*

Para casos poucos usuais, existem outras três formas de subamostragem presentes na literatura científica [161, 162, 163], cada uma com finalidades diferentes:

1. **Pooling piramidal:** Para eliminar a limitação de imagens de entrada com tamanho fixo em determinadas arquiteturas de RNCs, He et al. [161] utilizaram a arquitetura geral da RNC, mas substituíram a última camada de agrupamento por uma camada de *pooling* piramidal espacial. O agrupamento em pirâmide espacial pode extrair representações de comprimento fixo de imagens arbitrárias (ou regiões), gerando uma solução flexível para lidar com imagens de diferentes escalas e tamanhos.

2. **Pooling estocástico:** Com o objetivo de resolver eventuais problemas de *overfit*, Zeiler e Fergus [162] propuseram um *pooling* estocástico que substituiu a abordagem de subamostragem convencional, com operações de agrupamento determinísticas, por um procedimento estocástico, escolhendo aleatoriamente a ativação dentro de cada região de *pooling* de acordo com uma distribuição multinomial.
3. **Def-Pooling:** Para lidar com a deformação e variação de características geométricas de objetos em imagens de forma mais eficiente, Ouyang et al. [163] introduziram uma nova camada de agrupamento com restrições de deformação, chamada de camada de *def-pooling*, para otimizar a rede convolucional, através do aprendizado de características de deformação de padrões visuais.

3.6.3 Camada Densa

A Camada Densa (CD), também conhecida como *camada completamente conectada*, é comumente alocada nos estágios finais das redes convolutivas e corresponde, na maioria dos casos, à maior parte dos parâmetros ajustáveis do modelo - em torno de 90% [90]. A CD, quando presente na RNC, possui as mesmas características de funcionamento das camadas existentes em uma RNA tradicional, realizando o processamento de dados através de vetores unidimensionais com comprimentos pré-definidos [31].

É possível também configurar a última camada densa com certas categorias numéricas direcionadas à classificação de tensores, a exemplo de imagens coloridas [33], ou usá-la como uma extratora de características para processamento posterior através de outras técnicas inteligentes - abordagem híbrida [34]. A Figura 3.21a demonstra como é realizado o achatamento (conhecido, em inglês, como *flattening*) de uma matriz bidimensional (2D) para um vetor unidimensional (1D) que comporá parte da camada completamente conectada.

Considerando o que geralmente ocorre em uma rede convolucional empregada para classificação de imagens, a camada achatada vem logo após a última camada de agrupamento, composta por um tensor constituído de um conjunto de mapas de características, conforme pode ser visualizado na Figura 3.21b. Após o achatamento, a CD é interligada a um ou mais estágios contendo camadas totalmente conecta-

das, convertendo, deste modo, o tensor de mapas de características em vetores de características unidimensionais. Estas camadas densas, por sua vez, realizam transformações no volume de dados de entrada, que ocorrem em função do novo arranjo de parâmetros (pesos e *bias* dos neurônios) e das funções de ativação associadas [90].

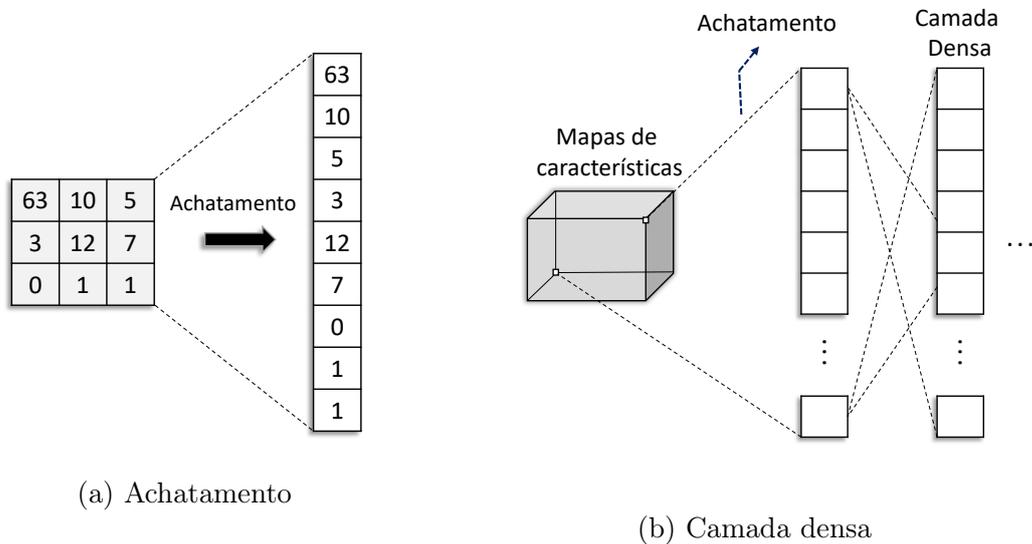


Figura 3.21: Representação das camadas achatadas da RNC.

Uma desvantagem conhecida das camadas densas é que elas usualmente possuem um número muito grande de parâmetros, o que resulta em um elevado custo computacional para treiná-las. Portanto, uma solução alternativa é remover parte dessas camadas ou diminuir as conexões com o emprego de determinadas alterações na arquitetura das redes convolutivas. Por exemplo, Szegedy et al. [164] projetou uma rede ampla e profunda, intitulada de *GoogLeNet*, alterando as camadas densas por arquiteturas esparsamente conectadas, reduzindo o custo computacional associado.

3.6.4 Camada de sobreamostragem (*Up-sampling*)

As camadas de sobreamostragem (em inglês, *up-sampling*) são rotinas típicas de algoritmos de aprendizagem profunda em que a saída desejada é uma nova imagem ou máscara de predição [165]. Este tipo de operação possui a particularidade de aumentar a dimensionalidade espacial (largura e altura) dos tensores de entrada, gerando novos mapas de características com resoluções ampliadas. Estas camadas de sobreamostragem costumam assumir um importante papel em redes convolucionais projetadas para segmentação de imagens [32].

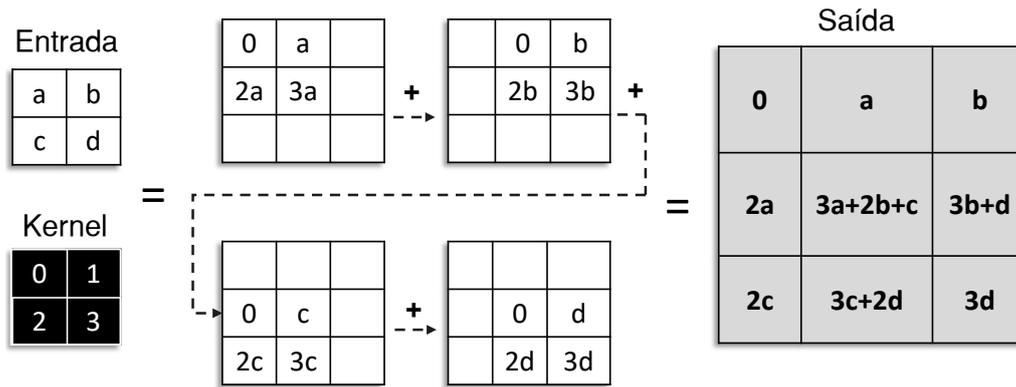


Figura 3.22: Operação de Convolução Transposta (CT) entre uma matriz de dimensões 2×2 e um *kernel* do tipo 2×2

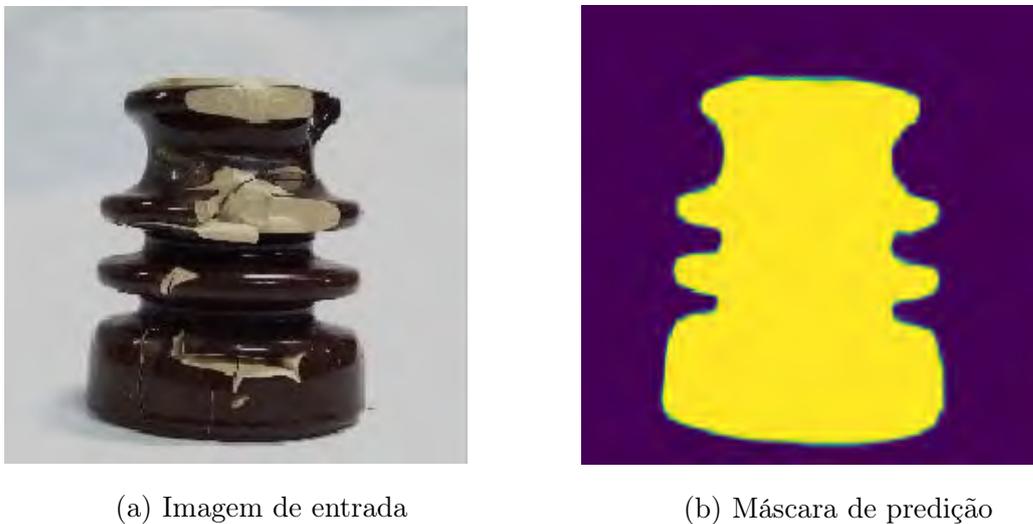


Figura 3.23: Exemplo de máscara de predição, para imagem de entrada, gerada por um algoritmo de aprendizagem profunda composto por camadas de sobreamostragem

Uma forma muito eficiente de sobreamostragem é conhecida como Convolução Transposta (CT) - também conhecida, em inglês, como *fractionally strided convolution*. Esta operação possui um comportamento semelhante à convolução tradicional, mas a sua funcionalidade principal é gerar mapas de características de saída com dimensionalidade espacial maior do que o tensor de entrada [166]. Da mesma forma como ocorre com as camadas convolutivas, as camadas compostas por CT também possuem filtros receptivos que são ajustados através das rotinas de treinamento por retropropagação de erro [165].

A Figura 3.22 ilustra, de forma simplificada, uma operação de convolução trans-

posta entre uma matriz do tipo 2×2 e um *kernel* com a mesma dimensionalidade. Após ocorrer a CT, uma nova matriz foi gerada, agora com dimensionalidade superior à da entrada, que neste caso foi de 3×3 . Dumoulin et al. [166] também detalham formas mais complexas de equacionamento da convolução transposta e alguns algoritmos direcionados a aplicações em aprendizado profundo.

É possível ilustrar um exemplo prático de emprego da referida técnica de sobre-amostragem. Nas imagens presentes na Figura 3.23, pode-se visualizar simultaneamente uma imagem de entrada (fotografia de isolador de distribuição defeituoso presente na Figura 3.23a) e, à direita, uma matriz bidimensional - máscara de predição (Figura 3.23b)-, produzida na saída de um algoritmo de aprendizagem profunda constituído por diversas camadas de convolução transposta em sua planta arquitetural [32].

3.6.5 Funções de Ativação

A Função de Ativação (FA), comumente representada matematicamente na literatura como $\varphi(v)$, define a saída de um neurônio ou filtro receptivo em função do campo local induzido v . Para que seja viável trabalhar com redes neurais de múltiplas camadas e com algoritmos de aprendizado baseados na retropropagação de erro, é necessário que a função $\varphi(v)$ seja contínua, não-linear e diferenciável em relação a v . De acordo com Haykin [77], a diferenciabilidade é a única exigência que a função de ativação deve satisfazer. A seguir, serão detalhadas cinco funções de ativação muito utilizadas no campo de aprendizagem profunda:

1. **Função sigmoide:** Esta função não-linear $\varphi_S(v)$, cujo gráfico possui a forma semelhante à letra s (ver Figura 3.24a), é uma das modalidades mais convencionais para construção de camadas de redes neurais do tipo perceptron [31]. Ela é uma função monotônica crescente e exibe um balanceamento entre o comportamento linear e não-linear [77]. Um exemplo de $\varphi_S(v)$ é a função logística, definida por

$$\varphi_S(v) = \frac{1}{1 + e^{-av}}, \quad (3.77)$$

para $a > 0$ e com campo local induzido definido como $-\infty < v < \infty$. O parâmetro a determina a inclinação da função. A saída de $\varphi_S(v)$ é não-linear

e se encontra dentro do intervalo de $[0, 1]$. Derivando esta função em relação a v , obtemos também uma função contínua (eq. 3.78)

$$\varphi'_S(v) = a\varphi_S(v)[1 - \varphi_S(v)]. \quad (3.78)$$

As características mencionadas facilitam o processo de treinamento de algoritmos de aprendizagem.

2. **Função Tangente Hiperbólica:** Esta função, expressa matematicamente como $\varphi_T(v)$, é um exemplo conhecido de não-linearidade anti-simétrica, cujos valores de saída pertencem ao primeiro e ao terceiro quadrantes do eixo cartesiano [167]. Ela pode ser definida pela equação

$$\varphi_T(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}, \quad (3.79)$$

podendo assumir valores na faixa de $[-1, 1]$, conforme mostrado na Figura 3.24b. A função $\varphi_T(v)$ também é diferenciável, inserindo um termo quadrático na função de ativação, como pode ser visto na equação

$$\varphi'_T(v) = 1 - \varphi_T(v)^2. \quad (3.80)$$

3. **Função Relu:** A função do tipo retificadora (em inglês, *Rectified Linear Units* - ReLU) vem produzindo desempenho satisfatórios quando empregadas em algoritmos de aprendizado profundo [33]. A curva da ReLU, ($\varphi_R(v)$), pode ser visualizada na Figura 3.24c. A sua equação é dada por

$$\varphi_R(v) = \max(0, v), \quad (3.81)$$

e sua derivada se assemelha a uma função binária:

$$\varphi'_R(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (3.82)$$

De acordo com Witten et al. [78], como a função $\varphi_R(v)$ anula os valores de argumento negativos, algumas unidades no modelo produzirão ativações que são nulas, dando uma propriedade de “esparsidade” que é útil em muitos contextos.

4. **Função Leaky ReLU:** Esta função é muito semelhante à ReLU, tendo em vista que, para valores positivos de v , o comportamento é idêntico ao $\varphi_R(v)$,

contudo a *Leaky ReLU*, expressa como $\varphi'_L(v)$, apresenta uma alteração para os valores negativos de v - uma pequena inclinação no terceiro quadrante, definido por uma constante $b \ll 1$. A Figura 3.24d mostra o gráfico dessa função e sua equação pode ser obtida como

$$\varphi_L(v) = \begin{cases} v, & \text{se } v \geq 0 \\ bv, & \text{se } v < 0 \end{cases} \quad (3.83)$$

A derivada de sua parte positiva é igual 1. Para valores negativos de v , $\varphi'_L(v)$ corresponde a uma pequeno valor definido pela constante b .

$$\varphi'_L(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ b, & \text{se } v < 0 \end{cases} \quad (3.84)$$

Uma vantagem da utilização de $\varphi'_L(v)$ é a menor dependência da rede neural a uma etapa de inicialização dos pesos. No caso ReLU, por exemplo, parte do modelo computacional pode ter dificuldades para aprender se os neurônios não forem ativados no início. No caso da *Leaky Relu*, estes valores não serão iguais a zero e a rede tenderá a convergir em algum momento futuro [168].

5. **Função Softmax:** Ela é utilizada em problemas de classificação do tipo *um-vs-todos* [82]. Essa função força a saída de uma rede neural a representar a probabilidade da informação ser de uma das classes definidas. Sem ela, as saídas dos neurônios são simplesmente valores numéricos, onde o maior indica a classe vencedora. A sua equação pode ser visualizada abaixo:

$$\varphi(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.85)$$

Nessa equação, K representa o número de classes do classificador e i é o índice do neurônio de saída. O termo $\mathbf{z} = \{z_1, \dots, z_K\}$ designa o vetor de neurônios de saída. Um exemplo de aplicação prática de $\varphi_i(\mathbf{z})$ direcionada à tarefa de classificação pode ser encontrado na referência [31].

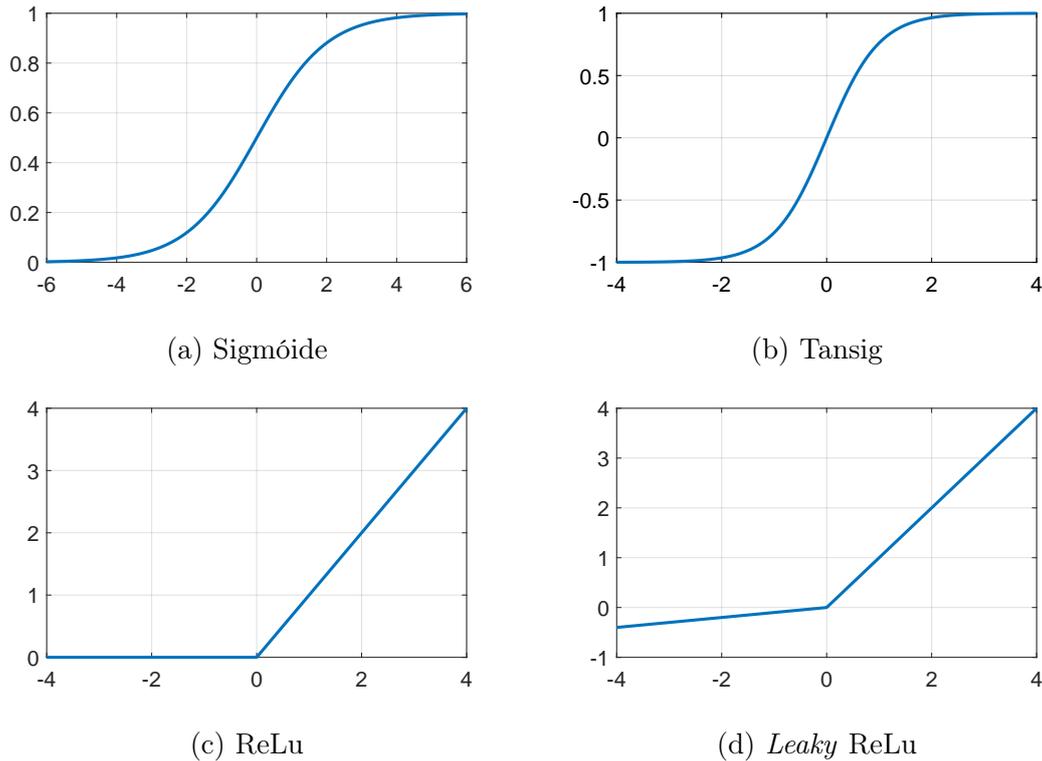


Figura 3.24: Gráficos gerados pelas principais funções de ativação implementadas em algoritmos de aprendizagem profunda.

3.7 Estratégias de treinamento

As técnicas de aprendizado profundo possuem vantagens estratégicas em relação aos algoritmos inteligentes tradicionais, a exemplo da viabilidade de construir arquiteturas profundas e, conseqüentemente, aprender informações com maiores níveis de abstração em relação às tarefas a serem executadas [145]. No entanto, a grande quantidade de parâmetros disponíveis nestes modelos computacionais pode ocasionar problemas de desempenho de treinamento, tais como sobreajuste, subajuste e *mode-collapse* [169, 86]. Para o caso do sobreajuste (em inglês, *overfitting*), os modelos computacionais podem se adaptar de forma demasiada ao *dataset* de treinamento, mas não conseguem extrapolar ou generalizar este mesmo desempenho para outros tipos de base de dados que fazem parte do mesmo problema trabalhado. Esta condição costuma gerar diferenças elevadas entre os resultados previstos nos *datasets* de treino e de teste utilizados nos classificadores. Para este tipo de sensibilidade a flutuações no conjunto de treino é dado o nome de variância do modelo [170].

No caso da subamostragem (em inglês, *underfitting*), o modelo pode apresentar

um determinado viés (*bias*) que deriva de um desajuste entre a classe do modelo e a distribuição de dados subjacentes [169]. Modelos simplificados demais para um determinado *dataset* podem não ser capazes de realizar o treinamento corretamente, gerando *bias* elevados. Deste modo, para o correto ajuste dos modelos inteligentes, faz-se necessário tomar medidas para que os algoritmos trabalhados atinjam o melhor compromisso entre viés e a variância [170].

Outro fator importante que impacta negativamente no desempenho de treinamento de redes convolutivas empregadas especificamente para as tarefas de geração de imagens é o *mode-collapse*. Esta condição ocorre quando apenas pequenos subconjuntos de distribuição de probabilidades da saída são representados pelo algoritmo gerador de imagens [86]. De acordo com Salimans et al. [171], o *mode-collapse* é um problema muito presente no estágio de treinamento de redes convolutivas adversárias condicionais, também conhecidas como cGANs (*Conditional Generative Adversarial Networks*) [172].

Ao longo dos últimos anos, vários métodos de regularização surgiram para prevenir ou mitigar os efeitos dos referidos problemas de treinamento dos algoritmos de aprendizagem profunda. Nesta seção, serão introduzidas estratégias de treinamento que podem influenciar positivamente o desempenho de aprendizado dos modelos computacionais.

3.7.1 Penalidades na Função Objetivo

Grande parte das estratégias de regularização empregadas em algoritmos de aprendizagem profunda são direcionadas a estabelecer limitações específicas no ajuste de parâmetros variáveis dos modelos computacionais [79]. Com foco na redução de problemas de desempenho associados a altas variâncias de treinamento, como é o caso da sobreamostragem, revela-se como uma abordagem bem sucedida a implementação de *mecanismos de penalização* da função custo, $\mathcal{L}(\cdot)$ - também conhecida, no campo de aprendizagem de máquina, como *função objetivo*. Com este propósito, adiciona-se um *termo de regularização*, $\lambda\mathcal{L}_R(\mathbf{W})$, alterando a função objetivo convencional, obtendo a equação 3.86 [78]:

$$\tilde{\mathcal{L}}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \lambda_R\mathcal{L}_R(\mathbf{W}), \quad (3.86)$$

com $\tilde{\mathcal{L}}(\cdot)$ denominado de *função objetivo regularizada*. A função de regularização é expressa como $\mathcal{L}_R(\cdot)$ e λ_R é um hiper-parâmetro que pondera a contribuição relativa do termo de regularização. Segundo Haykin [77], $\mathcal{L}_R(\cdot)$ também é conhecido como termo de *punição da complexidade* e depende unicamente dos parâmetros variáveis do modelo. Para o caso de λ_R , quanto maior for seu valor, maior será a importância proporcionada pela regularização durante a etapa de aprendizagem do modelo computacional.

De acordo com Yang et al. [86], existem outras formas de regularização direcionadas para modelos inteligentes mais exóticos, a exemplo das *redes generativas adversárias* (GANs) [21]. Para estes casos, é possível empregar um *termo de diversidade*, $\lambda_f \mathcal{L}_z(G)$, resultando na seguinte equação de aprendizagem para as GANs:

$$\min_G \max_D \tilde{\mathcal{L}}_{GANs}(D, G) = \min_G \max_D \mathcal{L}(D, G) - \lambda_f \mathcal{L}_z(G), \quad (3.87)$$

onde $\mathcal{L}_z(\cdot)$ corresponde ao *fator de diversidade* de $\tilde{\mathcal{L}}_{GANs}(\cdot)$ e λ_f é a *taxa de controle da regularização*. A equação 3.87 será abordada com maiores detalhes na Subseção 3.11. A seguir serão tratadas as principais formas de penalidade da função objetivo empregadas para elevar a capacidade de generalização dos modelos de aprendizagem profunda [115].

Regularização L^1 : Esta modalidade de regularização sobre os parâmetros do modelo, \mathbf{W} , também conhecida como *decaimento de pesos L^1* , pode ser definida da seguinte forma [173]:

$$\mathcal{L}_R(\mathbf{W}) = \|\mathbf{W}\|_1 = \sum_i |w_i|. \quad (3.88)$$

Através desta abordagem, penaliza-se $\tilde{\mathcal{L}}(\cdot)$ com base na soma dos valores absolutos dos parâmetros de \mathbf{W} . O decaimento de pesos L^1 controlará a força do nível de penalização $\mathcal{L}_R(\mathbf{W})$ através do ajuste do hiper-parâmetro positivo λ_R . Dessa forma, a função objetivo regularizada será dada por:

$$\tilde{\mathcal{L}}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \lambda_R \|\mathbf{W}\|_1. \quad (3.89)$$

Através do cálculo dos gradientes $\partial \tilde{\mathcal{L}} / \partial \mathbf{W}$, comum ao algoritmo de retropropagação de erros (*backpropagation*), explorado na Subseção 3.2.1, obtém-se [79]:

$$\nabla_{\mathbf{W}} \tilde{\mathcal{L}}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \nabla_{\mathbf{W}} \mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \lambda_R \cdot \text{sign}(\mathbf{W}), \quad (3.90)$$

onde $\text{sign}(\mathbf{W})$ é o sinal de \mathbf{W} , ou seja, $+1$ se w for positivo e -1 se w for negativo. Empregando em seguida as equações 3.22 e 3.27, altera-se o algoritmo de retropropagação de erros, obtendo-se a seguinte equação:

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \eta\lambda_R \cdot \text{sign}[\mathbf{W}(n)] - \eta\nabla_{\mathbf{W}}\mathcal{L}[f(\mathbf{x}; \mathbf{W}(n)), \hat{\mathbf{p}}], \quad (3.91)$$

onde n é o passo de treinamento e η é a taxa de aprendizado. É possível observar que os pesos diminuem em uma quantidade constante até alcançarem o valor 0. Por conseguinte, a regularização L^1 tende a concentrar a influência dos parâmetros do modelo em um número relativamente pequeno de conexões de alta importância, enquanto os outros pesos são direcionados para zero, gerando, dessa forma, matrizes com configurações esparsas [173].

Regularização L^2 : Nesta modalidade de decaimento de pesos, também chamada de *Regressão Ride* ou *Regularização Tikhonov* [79], a estratégia consiste no ajuste dos pesos \mathbf{W} para valores próximos de zero, a partir da adição de um termo $\frac{1}{2}\|\mathbf{W}\|_2^2$ à função objetivo [146]. A função custo regularizada será dada por:

$$\tilde{\mathcal{L}}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \frac{\lambda_R}{2}\mathbf{W}^T\mathbf{W}. \quad (3.92)$$

Através do cálculo dos gradientes $\partial\tilde{\mathcal{L}}/\partial\mathbf{W}$:

$$\nabla_{\mathbf{W}}\tilde{\mathcal{L}}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] = \nabla_{\mathbf{W}}\mathcal{L}[f(\mathbf{x}; \mathbf{W}), \hat{\mathbf{p}}] + \lambda_R\mathbf{W} \quad (3.93)$$

Seguindo o mesmo procedimento adotado para a equação 3.91, encontra-se a relação de ajuste dos parâmetros de \mathbf{W} :

$$\mathbf{W}(n+1) = (1 - \eta\lambda_R)\mathbf{W}(n) - \eta\nabla_{\mathbf{W}}\mathcal{L}[f(\mathbf{x}; \mathbf{W}(n)), \hat{\mathbf{p}}]. \quad (3.94)$$

Nesta abordagem de regularização, os pesos são reduzidos de forma proporcional a \mathbf{W} . Também é possível afirmar que a adição deste termo de decaimento modificou a regra de aprendizagem de forma a reduzir multiplicativamente o vetor de pesos por um fator contante a cada passo, imediatamente antes de realizar a atualização convencional dos gradientes [79]. O resultado esperado após a implementação desta técnica é a obtenção de um modelo com pesos sinápticos mais próximos de zero, promovendo uma maior capacidade de generalização entre *datasets* de treino e de testes.

Fator de Diversidade: Nesta modalidade de regularização, direcionada às GANs [21], o *modelo gerador* (G) sofre penalização em sua função objetivo de modo a reduzir a incidência do comportamento de *mode collapse* [174]. Nestes casos, implementa-se na rotina de aprendizagem das GANs o objetivo de maximização da função $\mathcal{L}_z(G)$, resultando na equação 3.95 [86]:

$$\max_G \mathcal{L}_z(G) = E_{z_1, z_2} \left[\min \left(\frac{\|G(\mathbf{x}, \mathbf{z}_1) - G(\mathbf{x}, \mathbf{z}_2)\|}{\|\mathbf{z}_1 - \mathbf{z}_2\|}, \tau \right) \right], \quad (3.95)$$

onde $\|\cdot\|$ indica a norma e τ é um valor limite para assegurar a estabilidade numérica. A intuição por trás desta técnica de regularização é baseada na seguinte constatação: quando o modelo gerador colapsa num modo único (*mode collapse*) e produz resultados determinísticos baseados apenas na variável condicionante \mathbf{x} , a equação 3.95 aproxima-se do seu mínimo. Dessa forma, a regularização explícita no modelo gerador, no melhor cenário, permite controlar um equilíbrio entre qualidade das amostras geradas e seus respectivos níveis de diversidade [174]. As GANs serão abordadas com maior profundidade na Subseção 3.11.

3.7.2 Técnicas de *Data Augmentation*

Para os algoritmos de aprendizagem profunda empregados no campo de visão computacional, a rotina de aumento de dados, também conhecida como *data augmentation*, se mostrou como uma ferramenta satisfatória para gerar informações adicionais ao modelo sem ter a necessidade de ampliar manualmente os *datasets* [33, 78]. Dependendo das características da base de dados avaliada, os algoritmos de *data augmentation* podem realizar alterações visuais sintéticas nas imagens, impactando positivamente no desempenho de generalização do modelo. A Figura 3.25 ilustra exemplos dos principais tipos de alterações visuais realizadas através dessa técnica (indicados pelos itens de a até h), enumerados como: (a) redução e (b) ampliação dimensional ; (c) movimentos de translação e (d) rotação; (e) reflexão (*flip*) horizontal; (f) alterações de cor e (g) brilho e (h) inclusão de ruído gaussiano [175]. Estas transformações são comumente realizadas de forma aleatória durante a execução das rotinas de treinamento dos modelos computacionais [90].

O processamento digital de imagens através da implementação de *data augmentation* pode ser empregado ao longo do treinamento da RNC, como uma estratégia

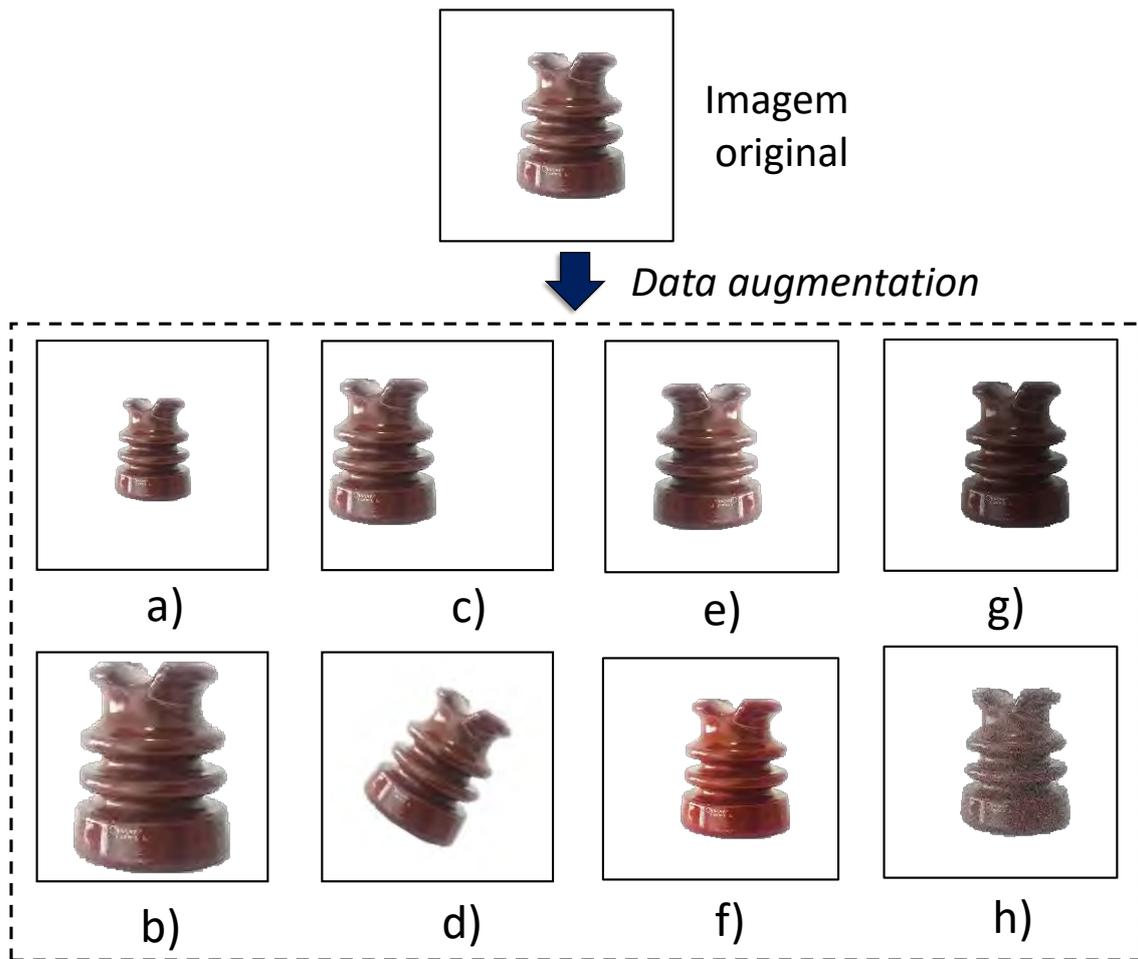


Figura 3.25: Imagens geradas por *data augmentation*.

de regularização, ou durante a fase de testes, para avaliar a robustez dos classificadores - etapa conhecida como *stress-test*. Um exemplo prático de implementação das técnicas de *data augmentation* e *stress-test* pode ser visto na referência [33].

3.7.3 Transferência de Conhecimento

A transferência de conhecimento (do inglês, *transfer learning*), pode ser entendida como uma técnica de aprendizagem de máquina em que um algoritmo inteligente, previamente treinado para uma tarefa específica, sofre uma forma de ajuste fino para uma segunda tarefa relacionada [79]. Para o caso dos algoritmos de aprendizagem profunda, a transferência de conhecimento pode ser conduzida através de um conjunto de atividades, a saber:

1. Realiza-se o pré-treinamento ou ajuste dos parâmetros do modelo computacional para estarem calibrados com um *dataset* chamado de *tarefa base*. Por

exemplo, o acervo de imagens do ImageNet - contendo cerca de 14 milhões de fotografias e 1.000 classes de saída-, é bastante utilizado para este propósito [91, 34].

2. Após a etapa de pré-treinamento, ajustam-se determinados conjuntos de parâmetros do modelo computacional (presentes nas camadas convolucionais e/ou densas) com as novas informações do conjunto de dados da tarefa de destino [33].
3. Como abordagem alternativa, utiliza-se novamente o modelo computacional previamente treinado com o *dataset* base, para se comportar como um extrator de características para o treinamento de diferentes algoritmos inteligentes [34]. Estas abordagens são conhecidas como configurações híbridas ou do tipo *ensemble* [176, 177].

O emprego da técnica de transferência de conhecimento apresenta diversos benefícios associados às tarefas de classificação de imagens, mas três deles devem ser enfatizados: otimização no desempenho de classificação para o conjunto de dados da tarefa de destino, redução do custo computacional de treinamento dos modelos computacionais e redução da tendência dos algoritmos à sobreamostragem quando são utilizados *datasets* de treinamento com baixo conjunto de dados.

3.7.4 *Dropout*

A técnica de *dropout* é uma rotina que compõe o algoritmo de treinamento de redes neurais artificiais. Ela foi apresentada por Hinton et al. [178], no ano de 2012. Em sua forma mais simples, esta rotina de aprendizado exclui os parâmetros ajustáveis do modelo a partir de uma determinada probabilidade definida pelo usuário, e os demais parâmetros são treinados através do algoritmo de retropropagação de erro. O *dropout* também pode ser entendido como um algoritmo de regularização, tendo em vista que ele tende a reduzir a sobreamostragem quando atua de forma a evitar co-adaptações complexas nos dados de treinamento e, ao mesmo tempo, eleva a capacidade de generalização dos modelos.

Baldi e Sadowski [179] explicaram essa ferramenta com maior profundidade, detalhando o comportamento do *droupout* desde redes lineares até algoritmos de

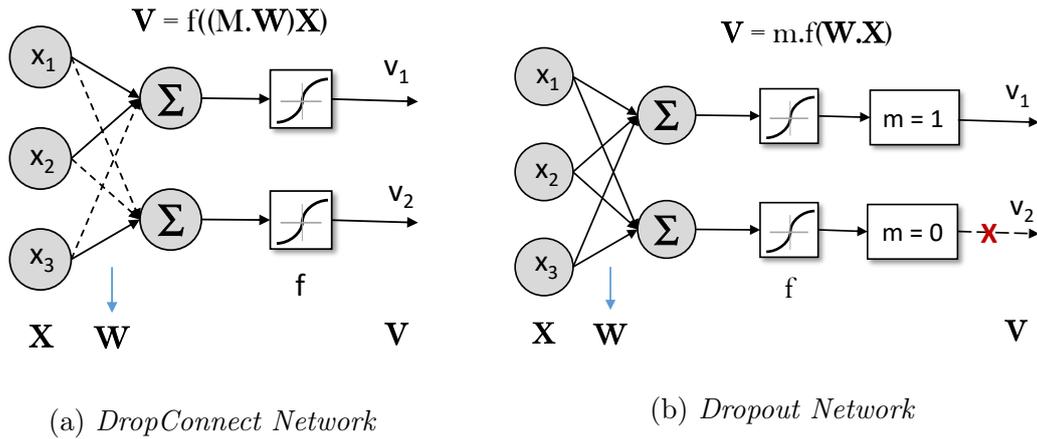


Figura 3.26: Representação das rotinas de *dropout*.

aprendizado profundo. Basicamente, existem dois tipos conhecidos de rotinas *dropout* [90], apresentadas na Figura 3.29. A primeira delas, conhecida como *DropConnect Network*, exclui um percentual dos parâmetros dos neurônios ou filtros receptivos (ver Figura 3.26a), enquanto o segundo modo atua na saída da função de ativação, ajustando o valor de $m = 0$, conforme pode ser visualizado na Figura 3.26b.

3.8 Aplicações das Redes Neurais Convolucionais

As redes neurais convolucionais apresentam uma ampla faixa de generalização entre tarefas (alta capacidade de adaptação para diferentes tipos aplicações [21, 78, 146]), e vêm trazendo contribuições relevantes nas áreas científica, médica, agrícola, industrial e de entretenimento. Estes algoritmos já comprovaram ser ferramentas adequadas para tarefas que utilizam imagens, vídeos e matrizes multidimensionais, obtendo desempenhos satisfatórios em aplicações relacionadas ao campo de visão computacional, tais como: classificação de imagens [33, 16]; segmentação semântica [24, 32] e instanciada (2D e 3D) [23, 25]; identificação de objetos [26]; extração de atributos [34] e geração de imagens sintéticas [21, 180]. A seguir, serão apresentadas algumas das arquiteturas de redes convolutivas que possuem um papel relevante na condução desta pesquisa, conforme cada tipo de tarefa desempenhada.

3.8.1 Classificação de Imagens

De acordo com a ilustração apresentada no início deste capítulo, presente na Figura 3.16, é possível projetar redes neurais convolucionais direcionadas às tarefas de classificação de imagens. Para estes casos, usam-se imagens como tensores de entrada e, para a camada de saída, um ou mais vetores unidimensionais de rótulos que definirão a classe predita. O sucesso destes algoritmos vai depender de diversos fatores, a exemplo da arquitetura de RNC desenvolvida, das características do *dataset* trabalhado e das estratégias de treinamento implementadas pelo usuário. Nesta subseção, serão avaliados alguns tipos de redes convolutivas que vêm se destacando na tarefa de classificação de imagens.

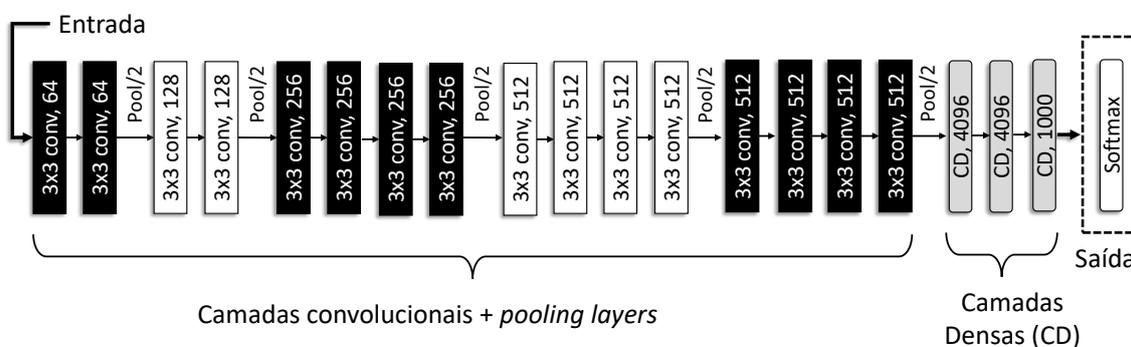


Figura 3.27: Planta arquitetural da RNC VGG-19.

VGG: As redes convolutivas do tipo VGG são consideradas atualmente como algoritmos bastante eficientes para tarefas de classificação de imagens. Esta arquitetura de RNC foi desenvolvida pelo grupo de geometria visual (em inglês, *Visual Geometry Group* - VGG) do departamento de Ciências da Engenharia da Universidade de Oxford, grupo este que confere o nome a esta rede convolutiva. Duas variações da VGG conquistaram a primeira e segunda posições na competição de classificação de imagens do ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*), que ocorreu no ano de 2014. Estes algoritmos detinham profundidades inovadoras para a época em que foram concebidos, com 16 e 19 camadas com parâmetros treináveis (VGG-16 e VGG-19, respectivamente) [181].

A VGG-19 é constituída pela intercalação de camadas convolucionais e de agrupamento, finalizando com um conjunto de camadas completamente conectadas. Para

a camada densa de saída, costuma-se utilizar uma função de ativação do tipo Softmax para as tarefas de classificação multi-classes e sigmoide para classificação binária [33]. A Figura 3.27 ilustra a planta arquitetural da VGG-19.

ResNet: A Rede Neural Residual, também conhecida como ResNet, foi introduzida por Kaiming [182], junto ao centro de pesquisas da Microsoft Research. A Figura 3.28a ilustra, de forma simplificada, a planta arquitetural de uma das variações da rede neural residual, intitulada de ResNet-50 [34]. Este tipo de RNC possui Blocos Residuais (ver Figura 3.28b) que facilitam o treinamento de redes convolutivas muito profundas. Uma variação deste modelo, com 152 camadas, foi a vencedora do Concurso ILSVRC do ano de 2015.

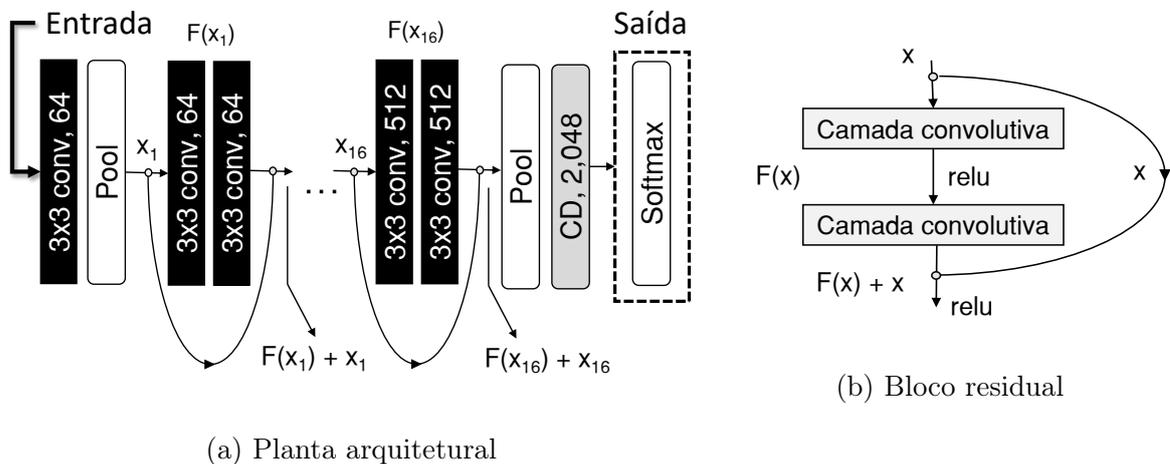


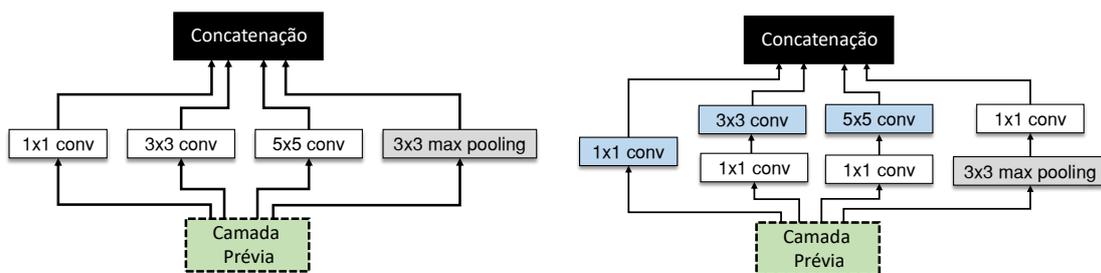
Figura 3.28: Representação simplificada da ResNet-50.

Para a arquitetura característica da ResNet, o mapeamento da camada subjacente $H(x)$, ao invés de aprender de forma direta (com poucos estágios enfileirados), se organiza em um mapeamento residual. Em outras palavras, camadas não-lineares podem ser configuradas da forma $F(x) = H(x) - x$. Deste modo, um novo tipo de mapeamento é criado, que pode ser expresso por $H(x) = F(x) + x$, onde $F(x)$ e x representam as camadas não-lineares empilhadas e a função de identidade, respectivamente. Uma revisão atual sobre esta arquitetura de rede convolutiva está presente na referência [183].

Inception: A arquitetura *Inception* pode ser explicada como uma variação de rede convolutiva cuidadosamente desenvolvida com o propósito de apresentar menor

custo computacional que os algoritmos do tipo VGG, mas que ainda assim demonstrasse desempenhos satisfatórios em tarefas de classificação de imagens [164]. Como particularidades importantes, este algoritmo emprega, ao longo da sua estrutura complexa e heterogênea, dois tipos de módulos muito similares entre si - intitulados de módulos *inception*.

O primeiro módulo *inception* é conhecido como versão *naive* (ingênua, na tradução literal para o português) e pode ser visualizado na Figura 3.29a. Ele realiza convoluções com filtros receptivos de formatos diferentes, se adaptando melhor a possíveis variações das regiões de interesse das imagens. A inclusão de módulos com estas características tornam a rede mais ramificada (com diversas camadas paralelas), mas, em contrapartida, a deixa menos profunda e menos susceptível a ocorrência de *overfitting* [164].



(a) Módulo *inception* (arranjo *naive*) (b) Módulo *inception* - redutor dimensional

Figura 3.29: Representação simplificada dos módulos *inception*.

Para o segundo tipo de módulo, apresentado na Figura 3.29b, o principal objetivo é reduzir o custo computacional associado à grande quantidade de parâmetros presentes nas camadas paralelas. Deste modo, foram inseridas camadas convolucionais do tipo 1×1 para tratar o sinal de entrada e reduzir os mapas de ativação que serão processados nas camadas 3×3 e 5×5 . O mesmo procedimento é realizado após a camada de *max pooling*.

3.8.2 Segmentação

No campo de visão computacional, a tarefa de segmentação se refere ao processo de dividir uma imagem ou vídeo em diferentes regiões para facilitar a análise de suas informações [146]. Os algoritmos de segmentação costumam ser definidos de

duas formas distintas. Na primeira modalidade, conhecida como segmentação semântica, as regiões de interesse são separadas nas imagens, mas nenhum rótulo é gerado [24, 165]. No segundo caso, intitulado de segmentação instanciada, as regiões são segmentadas e rotuladas de forma simultânea [23, 25].

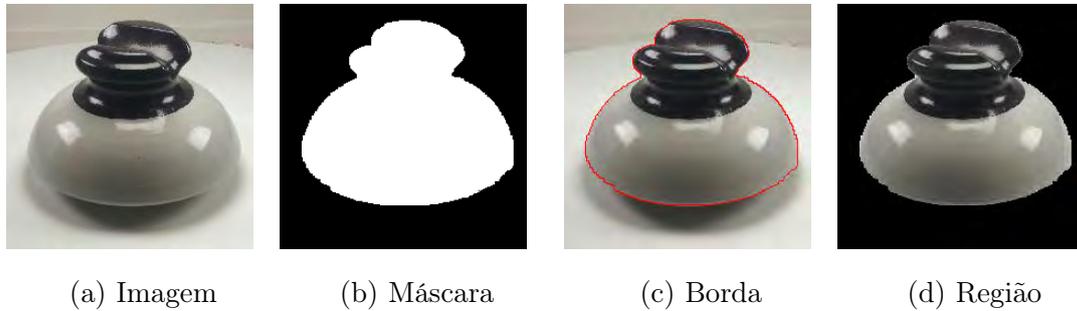


Figura 3.30: Exemplos de respostas produzidas por algoritmos de segmentação.

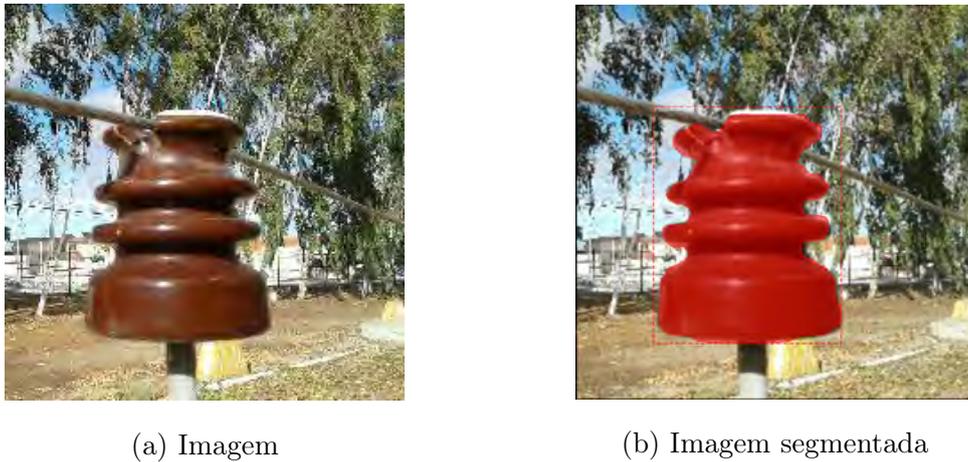


Figura 3.31: Exemplo de segmentação realizada com o algoritmo Mask R-CNN.

Para as duas modalidades de segmentação, os modelos inteligentes estão obtendo atualmente diversas vantagens em relação a outras metodologias, a exemplo de técnicas de *processamento digital de imagens*. De acordo com Garcia et al. [184] as técnicas de aprendizagem profunda de segmentação vêm superando as metodologias tradicionais por uma ampla margem em termos de precisão e de eficiência.

A Figura 3.30 exemplifica algumas das respostas associadas aos modelos computacionais projetados para segmentação de imagens. A partir da imagem bruta, presente na Figura 3.30a, a rotina de segmentação possibilita criar novas imagens, com diferentes características, a exemplo da geração de máscaras de predição (matrizes bidimensionais) indicando a localização e formato da região de interesse (ver

Figura 3.30b), da produção de contornos ou bordas sobrepostas a imagem original, conforme ilustrado na Figura 3.30c e geração de nova imagem possuindo somente a informação da região desejada – componente com fundo preto mostrado na Figura 3.30d.

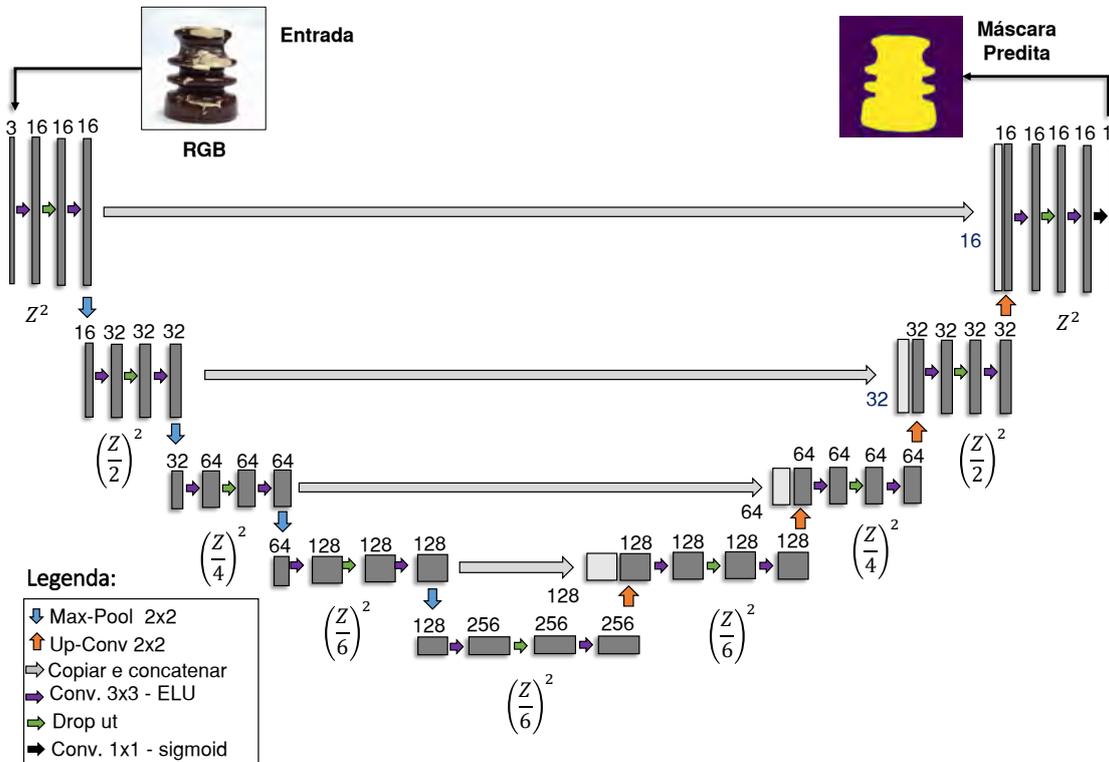


Figura 3.32: Exemplo de algoritmo de segmentação com configuração inspirada na U-Net.

U-Net: A U-Net pode ser entendida como uma rede convolutiva que foi inicialmente desenvolvida para segmentação de imagens biomédicas, de autoria de um grupo de pesquisa pertencente à Universidade de Freiburg, na Alemanha [165]. Mesmo sendo concebida para um campo de pesquisa limitado, a grande potencialidade demonstrada por este modelo computacional acabou viabilizando seu estudo e aplicação em diversas outras áreas.

A ideia principal deste algoritmo consiste em reconfigurar uma rede convolutiva convencional, composta de camadas convolucionais e de agrupamento, através da adição de uma nova arquitetura de camadas sucessivas, em que as operações de agrupamento são substituídas por operadores de sobreamostragem, a exemplo das convoluções transpostas. Deste modo, ao longo da segunda metade da RNC, as

camadas de sobreamostragem passam a aumentar a resolução dos mapas de características da saída de cada estágio. Para dar continuidade ao aprendizado nesta região, também são inseridos estágios convolucionais sucessivos para que a rede convolucional tenha a capacidade de aprender e formar uma matriz de predição de saída adequada, baseada na máscara utilizada para o treinamento.

Esta RNC é constituída por um trecho de contração e outro caminho de expansão, característica que lhe confere uma arquitetura em formato de “u”. Durante os estágios de contração, a informação espacial é reduzida enquanto as informações de características são aumentadas. No caminho expansivo, por outro lado, é realizada a combinação das informações espaciais e de características através de uma sequência de sobreamostragens e concatenações com os mapas de características de alta resolução - provenientes dos estágios de contração que possuem as mesmas dimensionalidades. A Figura 3.32 ilustra um algoritmo de segmentação adaptado da arquitetura U-Net, cuja principal alteração foi a adição de camadas de *dropout* ao longo de toda rede convolucional.

A U-Net usualmente possui uma planta arquitetural mais simples e de menor custo computacional para treinamento e simulação do que outros algoritmos de segmentação de aprendizagem profunda [146]. Contudo, ela já foi superada em desempenho por outras redes convolutivas maiores e mais complexas, a exemplo da Mask-RCNN [185] - empregada em tarefas de segmentação instanciada. Ainda assim, a U-Net ainda se apresenta como uma ferramenta muito útil para tarefas de segmentação que não possuem planos de fundo muito desafiadores, como é o caso da coleta de imagens realizadas em ambientes controlados [32]. A Figura 3.30b ilustra um exemplo de segmentação realizada pela U-Net, considerando uma imagem de isolador distribuição contendo poucas informações de plano de fundo. A Figura 3.31, por sua vez, apresenta uma imagem segmentada produzida pelo algoritmo Mask-RCNN para uma imagem de isolador com plano de fundo realista. Exemplos de aplicações recentes para as redes convolucionais U-Nets podem ser encontradas em [32, 35]

3.8.3 Identificação de Objetos

Os algoritmos de aprendizagem profunda projetados para *identificação de objetos* em imagens ou vídeos realizam duas tarefas simultâneas: a classificação do objeto e a localização da sua posição na imagem – através da geração de janela de demarcação (em inglês, *bounding box*) [27].

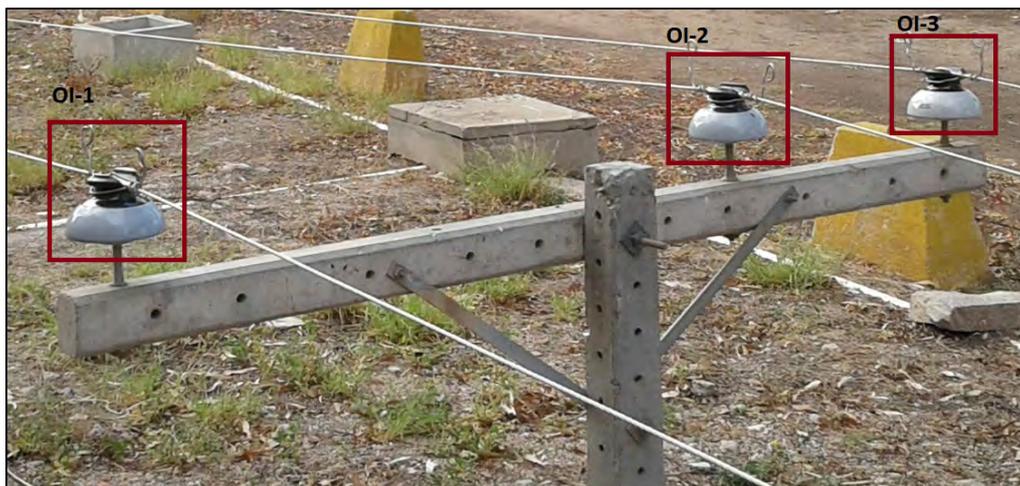


Figura 3.33: Exemplificação de imagem com regiões demarcadas por uma RNC desenvolvida para identificação de objetos.

Essa janela de demarcação, como o próprio nome sugere, consiste em localizar o objeto na imagem, enquanto a tarefa de classificação é voltada para diferenciar a sua classe, de acordo com as especificações de treinamento realizadas pelo usuário. De acordo com Li et al. [186] existem diversos algoritmos desenvolvidos para tarefas de identificação de objetos, podendo ser mencionadas as seguintes redes convolutivas: SSD, YOLO, R-CNN e *Faster R-CNN* [186, 26].

Deste modo, os modelos computacionais direcionados para identificação de objetos se diferem dos algoritmos de classificação, principalmente, por dois motivos. Primeiro, enquanto um classificador produz uma única categoria por imagem, um detector de objetos deve ser capaz de reconhecer vários objetos em uma única imagem. Em segundo lugar, enquanto os classificadores só precisam gerar probabilidades sobre classes, os detectores de objetos devem gerar probabilidades de associação a classes e também às coordenadas que identificam a localização dos objetos classificados [146]. A Figura 3.33 ilustra a resposta típica de um algoritmo de identificação de objetos (neste caso, *Single Shot MultiBox Detector* - SSD) treinado para identificar

isoladores cerâmicos de distribuição.

3.9 *Multi-Task Learning* (MTL)

No caso de algoritmos de aprendizagem profunda empregados para classificação de múltiplas tarefas de saída de forma simultânea - operação conhecida como *Multi-Task Learning* (MTL) [187]-, a função custo resultante, $\mathcal{L}'(\mathbf{y}, \hat{\mathbf{p}})$, combina as funções custo de todas as tarefas [188]. Por exemplo, para o caso de duas tarefas simultâneas, apresentado na Figura 3.34, considera-se $\mathcal{L}^{(a)}(\mathbf{y}, \hat{\mathbf{p}})$ para a tarefa 1 e $\mathcal{L}^{(b)}(\mathbf{y}, \hat{\mathbf{p}})$ para tarefa 2. O resultado de \mathcal{L}' é uma simples soma não ponderada

$$\mathcal{L}'(\mathbf{y}, \hat{\mathbf{p}}) = \mathcal{L}^{(a)}(\mathbf{y}, \hat{\mathbf{p}}) + \mathcal{L}^{(b)}(\mathbf{y}, \hat{\mathbf{p}}), \quad (3.96)$$

onde \mathbf{y} é o resultado calculado para o vetor de saída e $\hat{\mathbf{p}}$ é o valor presente no vetor de rótulos.

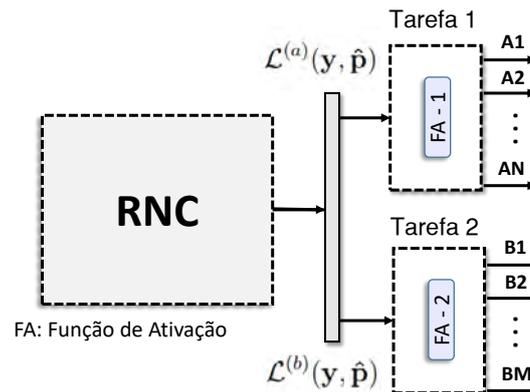


Figura 3.34: RNC empregada para operação de *Multi-Tasking Learning* (MTL).

A configuração *Multi-Tasking* utiliza a relação intrínseca entre as tarefas de aprendizagem individuais para melhorar o desempenho da classificação do modelo computacional. Existem na literatura casos de sucesso em que esta configuração proporcionou ganhos de desempenho na classificação de parte ou de todo o conjunto de tarefas avaliadas [33, 189].

3.10 *Multi-View*

A configuração de algoritmos inteligentes em múltiplas visões (em inglês, *Multi-View*) pode se tornar uma alternativa vantajosa para os casos em que seja necessário

uma visão tridimensional do objeto, antes do algoritmo desempenhar as tarefas de classificação ou segmentação [190]. Para este caso, todas as vistas são processadas simultaneamente no modelo computacional nos estágios de processamento de sinal da rede convolucional. Sun et al [191] demonstrou uma forma eficiente de reconhecimento de formatos tridimensionais através do emprego de técnicas *Multi-View*. A Figura 3.35 ilustra, de forma simplificada, como as múltiplas vistas seriam inseridas na rede convolucional para desempenhar uma tarefa de classificação.

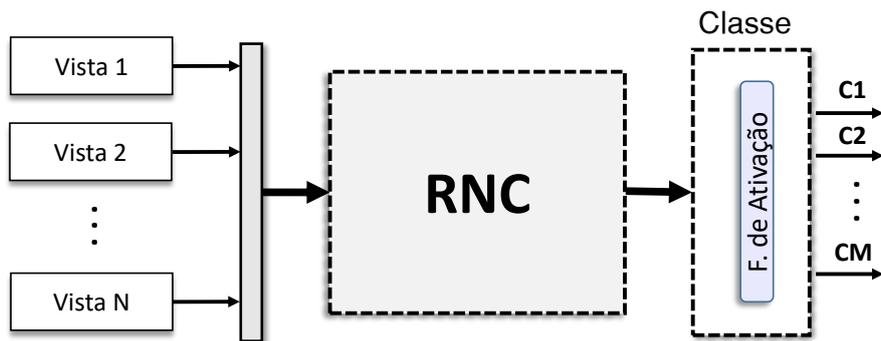


Figura 3.35: RNC empregada para configuração de *Multi-View*.

3.11 Redes Generativas Adversárias (GANs)

As *redes generativas adversárias* (do inglês, *Generative Adversarial Networks* - GANs) podem ser entendidas como um arranjo de algoritmos de aprendizado de máquina desenvolvidos com o propósito de gerar novos dados com estatísticas similares à distribuição de probabilidades do *dataset* de treinamento. De acordo com seus criadores (Goodfellow et al. [21]), esta estrutura de aprendizagem consiste em uma composição de dois modelos inteligentes que concorrem entre si - sob a forma de um *jogo de soma zero* (*minimax two-player game*), em que o ganho de um dos *agentes* corresponde à perda do outro. Vale mencionar também que estes algoritmos sofreram adaptações ao longo dos últimos anos e novos arranjos e/ou arquiteturas foram surgindo para o desenvolvimento de diferentes tipos de aplicações, a exemplo de técnicas para traslação de imagens (*image-to-image translation*) [180], transferência de estilo (*style transfer*) [192, 193] e invariância a planos de fundo (*background invariance*) [35].

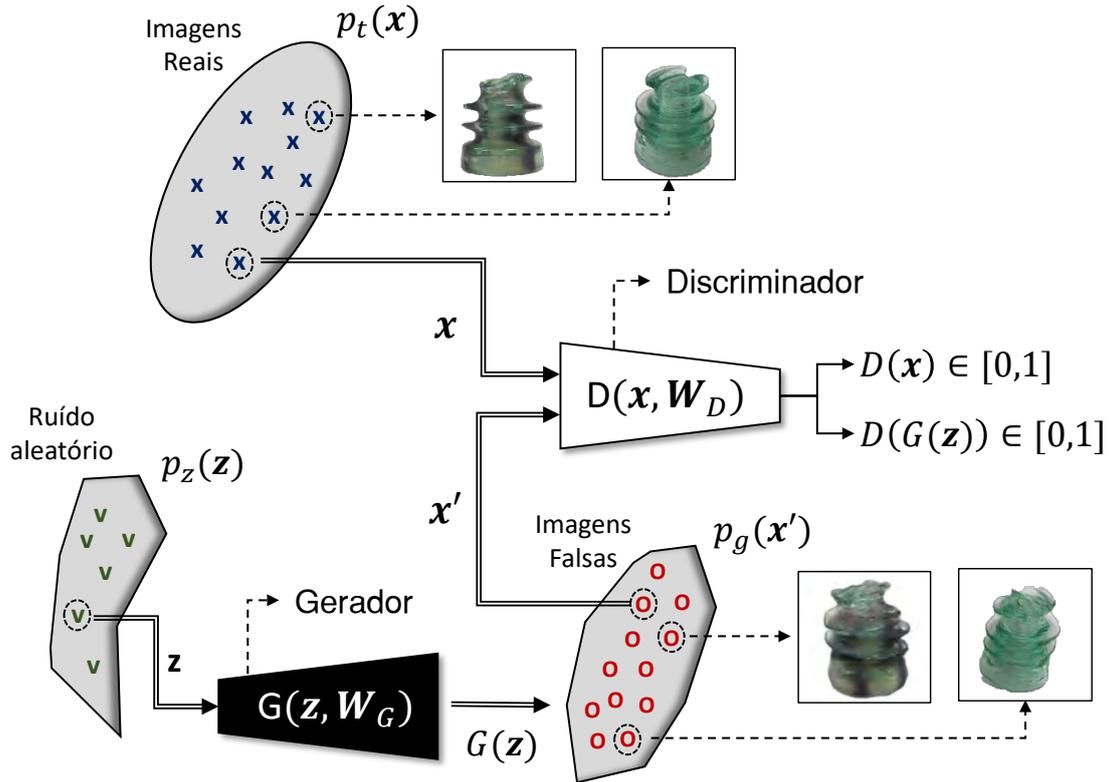


Figura 3.36: Diagrama de funcionamento das *redes generativas adversárias* (GANs).

As GANs, em sua configuração tradicional, são constituídas por dois algoritmos principais: um *modelo generativo* (G) e um *modelo discriminativo* (D) [174]. Quanto ao modelo generativo, é possível afirmar que o mesmo é uma aplicação diferenciável e possui a finalidade de reproduzir a função de distribuição pertencente ao *dataset* de treinamento, $p_t(\mathbf{x})$, de modo a se tornar um algoritmo gerador de novos dados. O modelo discriminador, por sua vez, é geralmente constituído por um classificador binário, que estima a probabilidade de uma amostra de entrada ser oriunda da base de dados de treinamento (exemplo verdadeiro) ou ter sido produzida pelo modelo gerador (dado sintético - exemplo falso) [79]. Por conseguinte, a rotina de treinamento de G se destina a modificar a função de distribuição inerente aos dados sintéticos, $p_g(\mathbf{x}')$, de forma a maximizar as chances de D cometer um erro de classificação.

Como condição ideal para o treinamento das GANs, objetiva-se equiparar as distribuições de probabilidades pertencentes ao *dataset* de treinamento com os exemplos sintéticos produzidos pelo modelo generativo, aproximando-se do patamar desejado: $p_g(\mathbf{x}') \cong p_t(\mathbf{x})$. A Figura 3.36 ilustra um diagrama simplificado de funcionamento

de redes generativas adversárias direcionadas à tarefa de geração de imagens. Neste exemplo, é possível visualizar o espaço latente, $p_z(\mathbf{z})$, que serve como origem dos vetores ruído \mathbf{z} utilizados na entrada do modelo gerador de imagens, $G(\mathbf{z}, \mathbf{W}_G)$. No caso do modelo discriminador, representado como $D(\mathbf{x}, \mathbf{W}_D)$, o mesmo recebe os tensores de entrada (imagens RGB) \mathbf{x} e \mathbf{x}' , pertencentes às distribuições de probabilidades $p_d(\mathbf{x})$ e $p_g(\mathbf{x}')$. Os Termos \mathbf{W}_G e \mathbf{W}_D , por sua vez, correspondem aos parâmetros ajustáveis de G e D , respectivamente.

Tendo em vista que as GANs são geralmente constituídas por redes neurais artificiais, o processo convencional de treinamento é dado através do algoritmo de retropropagação de erro [174]. Para o cálculo das funções custo pertencentes à referida composição de modelos inteligentes, adotam-se, na saída de D , a função de entropia binária cruzada (em inglês, *binary cross-entropy*), dada por:

$$\mathcal{L}_B(y, \hat{p}) = [\hat{p} \log y + (1 - \hat{p}) \log[(1 - y)]] \quad (3.97)$$

onde y corresponde ao valor de saída do modelo discriminador e \hat{p} representa o seu respectivo rótulo. Como estratégia de treinamento de D , adota-se as igualdades de \hat{p} em função de y de acordo com as variáveis de entrada \mathbf{x} e \mathbf{x}' . Obtém-se então a seguinte relação:

$$\hat{p} = \begin{cases} 1, & \text{se } y = D(\mathbf{x}) \\ 0, & \text{se } y = D[G(\mathbf{z})] \end{cases} \quad (3.98)$$

A equação 3.98 condiciona os valores de \hat{p} para a rotina de treinamento do modelo discriminador. Definir-se-á $\hat{p} = 1$ para os exemplos pertencentes a $p_t(\mathbf{x})$ e $\hat{p} = 0$ para os dados provenientes de $p_g(\mathbf{x}')$. Deste modo, obtêm-se as seguintes equações resultantes para $\mathcal{L}_B(y, \hat{p})$:

$$\mathcal{L}_B[D(\mathbf{x}), 1] = \log[D(x)], \quad (3.99)$$

$$\mathcal{L}_B[D(G(\mathbf{z})), 0] = \log[1 - D(G(\mathbf{z}))]. \quad (3.100)$$

As equações 3.99 e 3.100 definem os comportamentos da função custo para cada um dos dois tipos de variável de entrada. Como informação complementar, a Fi-

gura 3.37 apresenta as curvas de $\mathcal{L}_B[D(\mathbf{x}), 1]$ e $\mathcal{L}_B[D(G(\mathbf{z}), 0)]$ em função das variáveis $D(\mathbf{x})$ e $D[G(\mathbf{z})]$, respectivamente. Deste modo, para que D apresente um comportamento condizente com as condições estabelecidas na equação 3.98, o modelo discriminador deve maximizar ambos os termos apresentados para $\mathcal{L}_B(y, \hat{p})$.

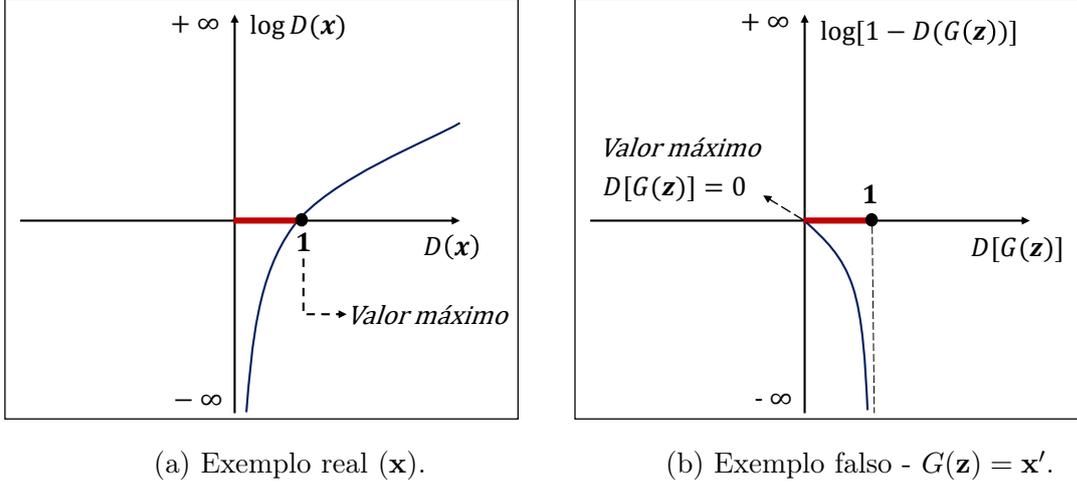


Figura 3.37: Comportamento de $\mathcal{L}_B(y, \hat{p})$ em função do tipo de variável de entrada do modelo discriminador (D).

No caso da rotina de treinamento do modelo gerador, objetiva-se que $D[G(\mathbf{z})]$ apresente respostas semelhantes às obtidas por $D(\mathbf{x})$, adotando-se, deste modo, $\hat{p} = 1$ para todos os valores de $D[G(\mathbf{z})]$. Isto só reforça o argumento de G ter sido desenvolvido para gerar dados com estatísticas semelhantes aos exemplos de treinamento e, no melhor cenário, ter condições de produzir exemplos falsos que sejam classificados como verdadeiros pelo modelo discriminador. Reavaliando-se o gráfico apresentado na Figura 3.37b, pode-se inferir então que $\mathcal{L}_B(y, \hat{p})$ deve ser minimizado para este caso, pois é necessário que os valores de $D[G(\mathbf{z})]$ se aproximem da assíntota vertical, tendendo a $D[G(\mathbf{z})] = 1$. Para dar lastro às proposições apresentadas sobre o processo de aprendizagem das GANs, deve-se pontuar que Goodfellow et al. [79] sugeriram que a maneira mais simples de formular a rotina de aprendizagem das redes generativas adversárias é através do jogo de soma zero, que resulta na conhecida fórmula concebida pelos autores:

$$\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_t(\mathbf{x})}[\log D(x)] + E_{\mathbf{z} \sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \quad (3.101)$$

Na equação 3.101, a função $V(D, G)$ determina o nível de compensação (*payoff*)

do discriminador. No caso do gerador, o mesmo recebe o valor negativo de $V(D, G)$ como sua própria compensação [79]. Ao longo da etapa de treinamento, cada modelo tenta maximizar o seu próprio *payoff* e, caso seja atingida a convergência, as amostras produzidas pelo gerador (\mathbf{x}') se tornarão indistinguíveis dos exemplos reais (\mathbf{x}). Neste caso ideal, D produzirá a saída de $1/2$ para ambos os tipos de dados de entrada. E ao final do processo, o discriminador poderá ser desprezado.

3.11.1 Redes Generativas Adversárias Condicionais (cGANs)

As *redes generativas adversárias condicionais* (cGANs) foram inicialmente apresentadas por Mirza e Osindero [172], e podem ser classificadas como uma forma particular de adaptação das GANs tradicionais, propostas por Goodfellow et al. [21]. Neste arranjo de algoritmos inteligentes, o modelo gerador (G) e o modelo discriminador (D) são condicionados a algum tipo de informação extra, expressa como r . De acordo com Gui et al. [174] a função objetivo original, apontada na equação 3.101, é parcialmente modificada com a inclusão de um termo condicional, resultando na equação 3.102:

$$\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_t(\mathbf{x})} [\log D(x|r)] + E_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(z|r)))]. \quad (3.102)$$

Dentre as configurações disponíveis atualmente para compor as plantas arquiteturas das cGANs, pode-se afirmar que existe uma grande diversidade de meios e estratégias para inserção de novas entradas e/ou saídas tanto em G quanto em D , a depender do tipo de aplicação almejada [174]. Essas variáveis extras também podem conter diferentes tipos de informações, a exemplo de rótulos de classes [194], contornos de imagens [180], pontos-chave (*keypoints*) [195] e textos [196].

De forma complementar ao que foi apresentado na Figura 3.36, a Figura 3.38 ilustra uma representação simplificada de um arranjo específico de cGANs empregado para tarefas de geração *multi-classes*. Neste arranjo em particular - constituído por duas redes convolutivas ($G(\mathbf{z}, \mathbf{r})$ e $D(\mathbf{x})$) -, é possível observar que a configuração de entrada de G é do tipo de múltiplas variáveis (*multi-input*), possuindo, além do vetor aleatório \mathbf{z} , um termo \mathbf{r} que delimita a classe de \mathbf{x}' . No caso do modelo discriminador, o mesmo recebe apenas uma tipo de entrada por vez (\mathbf{x} ou \mathbf{x}'), contudo esta RNC apresenta a configuração de múltiplas saídas (*multi-task learning - MTL*)

- arquitetura descrita na Subseção 3.9. Isto implica que o modelo discriminador produzirá simultaneamente dois tipos de saídas (\mathbf{d} e \mathbf{r}') para cada um dos exemplos de entrada x , conseqüentemente, desempenhará um comportamento semelhante de cálculo das funções custos à condição apresentada na equação 3.96.

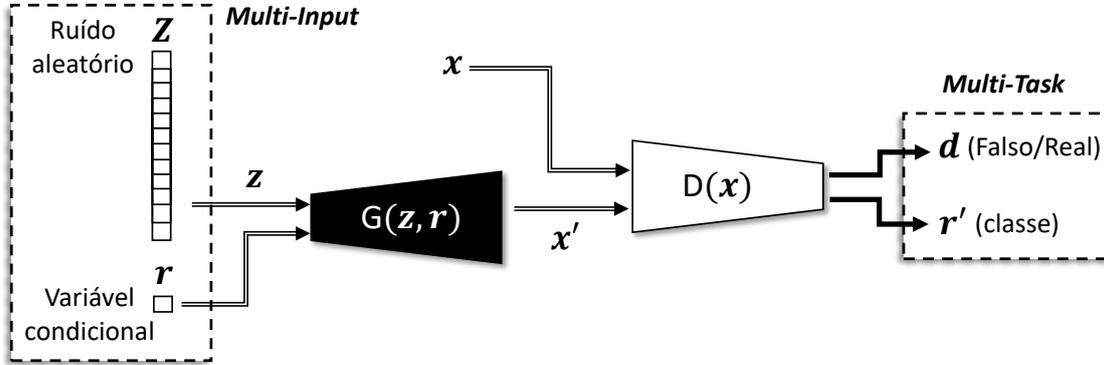


Figura 3.38: Representação simplificada de um arranjo de *redes generativas adversárias condicionais* (cGANs).

Como forma de otimizar o desempenho de treinamento das cGANs e elevar o nível de diversidade dos dados produzidos pelos modelos geradores, Yang et al. [86], propuseram a implementação de técnicas de regularização em G , estratégia também conhecida como *fator de diversidade*. Esta metodologia pode forçar o gerador a apresentar um nível de diversidade maior ao longo do processo de aprendizagem, reduzindo as chances de ocorrer o *mode collapse*. A equação resultante é apresentada a seguir:

$$\min_G \max_D \mathcal{L}_{cGAN}(D, G) - \lambda_f \mathcal{L}_z(G). \quad (3.103)$$

Os termos $\mathcal{L}_z(\cdot)$ e λ_f correspondem, respectivamente, ao *fator de diversidade* e à *taxa de controle* da regularização. A estrutura de treinamento do arranjo de cGANs apresentado nesta Subseção está presente no Algoritmo 01.

Algoritmo 01: Treinamento de *redes generativas adversárias condicionais* (cGANs) através do algoritmo de retropropagação de erros e emprego de técnica para cálculo de fator de diversidade com comparação de mini-lotes.

- 1: **Definir os parâmetros:** tamanho do mini-lote (m); número de épocas de treinamento (N); número de iterações em cada época (K); funções custos $\mathcal{L}_A(\cdot)$ e $\mathcal{L}_B(\cdot)$; *função de otimização* (ex: Adam); *fator de diversidade* ($\mathcal{L}_z(\cdot)$); *taxa de controle* da regularização ($\lambda_f(\cdot)$).
 - 2: **for** $[i] \leftarrow 0$ *to* N **do**
 - 3: **for** $[j] \leftarrow 0$ *to* K **do**
 - 4: Gerar mini-lotes: $\mathbf{Z}_A = \{\mathbf{z}_A^{(1)}, \dots, \mathbf{z}_A^{(m)}\}$ e $\mathbf{Z}_B = \{\mathbf{z}_B^{(1)}, \dots, \mathbf{z}_B^{(m)}\}$ de $p_g(\mathbf{z})$.
 - 5: Gerar mini-lote: de exemplos reais $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ e respectivos rótulos $\mathbf{r} = \{r^{(1)}, \dots, r^{(m)}\}$ de $p_t(\mathbf{x})$.
 - 6: $\mathbf{X}'_A \leftarrow G(\mathbf{Z}_A, \mathbf{r})$
 - 7: $\mathbf{X}'_B \leftarrow G(\mathbf{Z}_B, \mathbf{r})$
 - 8: Computar os *fatores de diversidade*: $\mathcal{L}_z(\mathbf{X}'_A, \mathbf{X}'_B)$
 - 9: Computar a *taxa de aprendizado*: $\lambda_f(\mathbf{X}'_A, \mathbf{X}'_B)$
 - 10: $\mathbf{d}_{\text{Real}}, \mathbf{r}'_{\text{Real}} \leftarrow D(\mathbf{X})$
 - 11: $\mathbf{d}_{\text{Falso}}, \mathbf{r}'_{\text{Falso}} \leftarrow D(\mathbf{X}'_A)$
 - 12: $\mathbf{d}_{\text{loss}} \leftarrow \mathcal{L}_A(\mathbf{d}_{\text{Real}}, 1) + \mathcal{L}_A(\mathbf{d}_{\text{Falso}}, 0) + \mathcal{L}_B(\mathbf{r}'_{\text{Real}}, \mathbf{r})$
 - 13: Otimizar D
 - 14: Gerar mini-lote: $\mathbf{Z}_C = \{\mathbf{z}_C^{(1)}, \dots, \mathbf{z}_C^{(m)}\}$ de $p_g(\mathbf{z})$.
 - 15: $\mathbf{X}'_C \leftarrow G(\mathbf{Z}_C, \mathbf{r})$
 - 16: $\mathbf{d}_{\text{Falso2}}, \mathbf{r}'_{\text{Falso2}} \leftarrow D(\mathbf{X}'_C)$
 - 17: $\mathbf{g}_{\text{loss}} \leftarrow \mathcal{L}_A(\mathbf{d}_{\text{Falso2}}, 1) + \mathcal{L}_B(\mathbf{r}'_{\text{Falso2}}, \mathbf{r}) - \lambda_f \mathcal{L}_z(\mathbf{X}'_A, \mathbf{X}'_B)$
 - 18: Otimizar G
 - 19: **end for**
 - 20: **end for**
-

Capítulo 4

Definição do Problema e Metodologia

Para compreensão do objeto de estudo, foi conduzida uma pesquisa detalhada sobre as *Redes Primárias de Distribuição Aéreas de Energia Elétrica* (RPDA) e dos principais procedimentos de *Inspeção Visual* (IV) empregados atualmente pelas concessionárias de energia elétrica. Outro ponto considerado relevante e também explorado nesta pesquisa são os *Sistemas Inteligentes de Inspeção Visual* (SIV), que foram definidos como um conjunto de potenciais tecnologias direcionadas à automação de tarefas apresentadas nos procedimentos operacionais de inspeção visual de RPDA. A Figura 4.1 ilustra um fluxograma simplificado contendo as principais atividades desenvolvidas neste trabalho. Para facilitar a compreensão, este fluxograma pode ser dividido em três grandes blocos de atividades, nomeadamente: (1) *levantamento teórico*, (2) *abordagem experimental* e (3) *análise de resultados*.

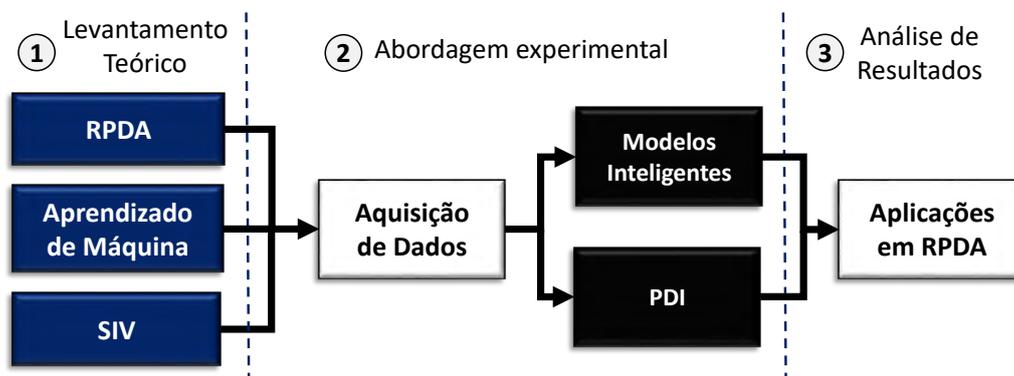


Figura 4.1: Fluxograma macro de atividades desenvolvidas para a pesquisa de Doutorado.

No que diz respeito ao primeiro bloco de atividades apresentado na Figura 4.1, referente à etapa de levantamento teórico, é correto afirmar que ela proporciona um arcabouço conceitual para as principais temáticas trabalhadas nesta pesquisa. De forma simplificada, pode-se enumerar os temas mais significativos para se atingir os objetivos propostos neste estudo, exemplificadamente: **(a)** sistemas inteligentes de classificação; **(b)** técnicas clássicas de aprendizado de máquina; **(c)** algoritmos de aprendizagem profunda; **(d)** redes primárias de distribuição aéreas de energia elétrica e, finalmente, **(e)** procedimentos relativos às atividades de inspeção visual e **(f)** demandas tecnológicas relevantes para o desenvolvimento de sistemas inteligentes de inspeção de RPDA.

Considerando a *abordagem experimental* (item 2), este estágio foi iniciado com a etapa de *aquisição de dados*. Para esta demanda, foram elaboradas duas metodologias de coleta de imagens. A primeira delas teve enfoque direcionado no controle apurado de parâmetros de iluminação, posição e interferências de plano de fundo das fotografias coletadas. Para a segunda opção, em contraponto com à primeira, foi projetado e construído um *trecho de rede de distribuição não energizada*, com foco na reprodução de ambientes com características realistas e não controladas (condições aleatórias de iluminação e planos de fundo). Estes *datasets* originais foram então empregados em diferentes estágios de processamento da informação, tais como: extração de características; treinamento, validação e testes dos modelos computacionais propostos.

A Figura 4.2 apresenta um diagrama detalhado das atividades realizadas durante a etapa experimental. É possível verificar nesta figura que, além do estágio de aquisição de imagens (descrito no parágrafo anterior), estão presentes as principais técnicas empregadas para o processamento das informações. Vale mencionar que os modelos computacionais propostos foram elaborados a partir do emprego de técnicas de *Processamento Digital de Imagens* (PDI) e de *aprendizado de máquina*. As abordagens relativas ao PDI foram adotadas, principalmente, para as tarefas de extração de atributos e de geração artificial de imagens. Para o campo de aprendizado de máquina, as principais metodologias adotadas podem ser enumeradas da seguinte forma: **(1)** técnicas clássicas; **(2)** aprendizado profundo; **(3)** técnicas híbridas; **(4)** técnicas de imunidade a planos de fundo (*background invariance*) e

(5) geração de imagens sintéticas. Outras abordagens também foram empregadas de forma complementar, a exemplo das técnicas de transferência de conhecimento (*transfer learning*) e de aumento de dados (*data augmentation*).

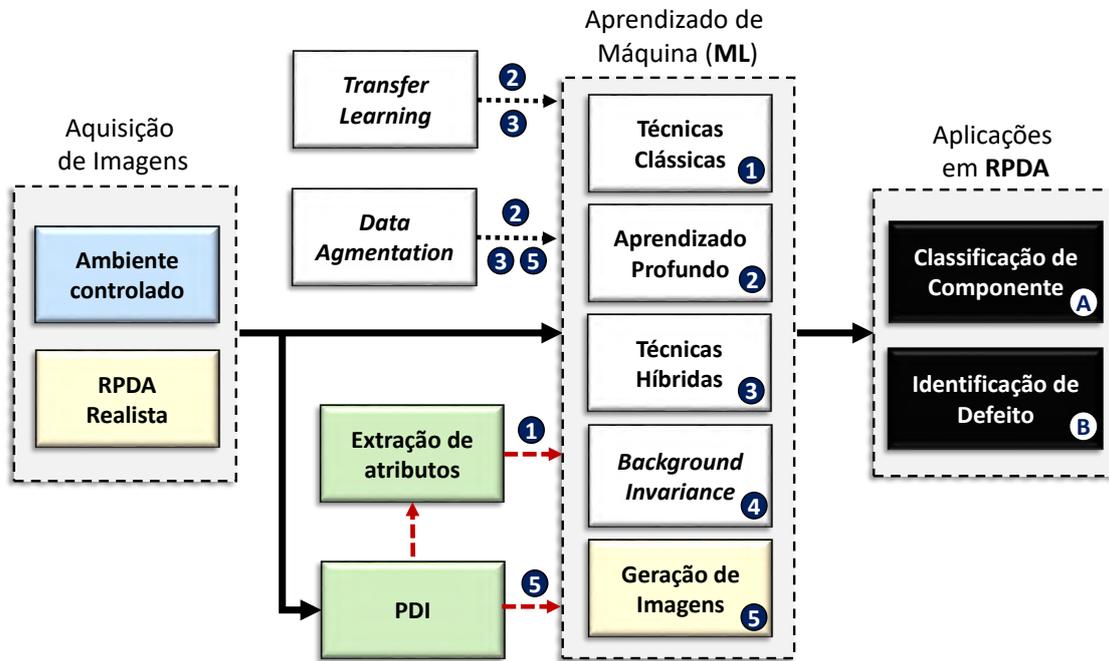


Figura 4.2: Diagrama de atividades desenvolvidas para a etapa experimental.

O último bloco de atividades presente no fluxograma da Figura 4.2, intitulado como *aplicações em RPDA*, apresenta o principal objetivo investigado nesta pesquisa. As duas aplicações contidas neste bloco são denominadas: (A) classificação de componentes e (B) identificação de defeitos. Para a primeira aplicação, objetiva-se realizar tarefas de classificação de componentes isoladores de distribuição, definindo o tipo de material e *design*, independente de interferências de plano de fundo das imagens, e da condição de conformidade destes dispositivos. Já para as aplicações direcionadas à identificação de defeitos, foi elaborado um conjunto de técnicas voltadas à melhoria de desempenho de classificação dos modelos de aprendizagem profunda empregados nas simulações.

Como atividades complementares a esta pesquisa, foi desenvolvido um conjunto de metodologias direcionadas a propor soluções para mitigar problemas particulares do objeto de estudo, que impactam no desempenho dos modelos computacionais. Duas soluções complementares podem ser mencionadas. A primeira consiste no de-

envolvimento de algoritmos para geração de imagens artificiais, com o objetivo de gerar *datasets* realistas e diversificados. A segunda estratégia, por sua vez, foi concebida para reduzir o nível de interferência apresentado por planos de fundo complexos de imagens de RPDA. Neste caso, foram desenvolvidas diferentes técnicas direcionadas a tarefas de *background invariance*. Conforme mencionado no Capítulo 2, as metodologias trabalhadas nesta pesquisa podem contribuir para o desenvolvimento de sistemas inteligentes de inspeção visual de redes primárias de distribuição aérea.

4.1 Aquisição de Imagens

Para a fase de aquisição de dados, foi criada uma coleção de imagens contendo informações sobre os componentes, defeitos e possíveis fontes de interferência. Como medida adicional, foi avaliada a flexibilidade para atualizar o conjunto de dados, considerando inovações tecnológicas e mudanças nos padrões de construção. No caso dos isoladores de distribuição, por exemplo, ocorrem mudanças contínuas de *design* e do tipo de material construtivo. Isto pode dificultar o desenvolvimento de um “*dataset* universal” que possa ser utilizado em todas configurações de redes primárias de distribuição aéreas.

Em termos práticos, é possível propor dois procedimentos para a aquisição de imagens de componentes de RPDA, sendo eles: **(a)** coleta *in loco*, realizada em RPDA reais ou em estruturas adaptadas construídas com a finalidade de facilitar a coleta de dados e **(b)** coleta em estúdios fotográficos, executadas em ambientes com iluminação e planos de fundo controlados. Para a primeira alternativa, é possível obter imagens que refletem características presentes nas redes elétricas distribuição. No entanto, podem ser mencionadas algumas desvantagens inerentes à realização deste tipo de procedimento:

1. Riscos de contato acidental do sistema de aquisição de imagens com condutores energizados ou estruturas de sustentação da rede elétrica.
2. Obstáculos relacionados com a geração de *datasets* balanceados e representativos para todas as classes de componentes e possíveis tipos de inconformidades. As RPDA reais são normalmente constituídas por um número muito superior de componentes íntegros em relação à condição defeituosa, e também existem

determinados tipos de defeitos que possuem menor incidência em relação a toda a gama de inconformidades possíveis.

3. dificuldades na atualização dos *datasets* para novos tipos de componentes instalados nas RPDA;
4. custos elevados para a construção de trechos de redes não energizadas (contendo estruturas e componentes realistas) empregadas com o objetivo de coleta de imagens.

Para o procedimento de aquisição de imagens em estúdio fotográfico, existe a vantagem dessa atividade ocorrer em um ambiente seguro e com condições externas controladas. No entanto, os *datasets* coletados carecem de informações relevantes que estão presentes nos ambientes reais, a exemplo de:

1. Fontes de interferência nas imagens, como vegetação, edificações ou objetos encontrados em planos de fundo de ambientes urbanos e rurais;
2. presença e/ou sobreposição de condutores e componentes complementares, tais como: a amarração metálica superior [50], chaves seccionadoras dentre outros;
3. existência de objetos estranhos, tais como: animais de pequeno porte, ninhos de pássaros e objetos atirados por vândalos;
4. variações de condições climáticas, brilho, dimensões e ângulos dos componentes.

Para esta fase de aquisição de imagens, o primeiro passo foi a seleção de quatro tipos de isoladores de distribuição que operam na faixa de 15 kV [37, 36], nomeadamente: **Isolador Cerâmico de Pino (ICP)**, **Isolador Cerâmico Bicolor (ICB)**, **Isolador Polimérico Cinza (IPC)**, **Isolador de Vidro Verde (IVV)**. A seleção dos referidos componentes, dentro do universo de isoladores possíveis, está relacionada, principalmente, à amostragem de três tipos de materiais construtivos distintos (cerâmico, polimérico e vítreo), e também em decorrência da alta incidência destes componentes nas redes elétricas de distribuição existentes no Estado da Bahia - tendo maior presença os isoladores cerâmicos ICP e ICB. As imagens representativas desses componentes podem ser vistas na Figura 2.3 (presentes na Subseção 2.1).

Numa etapa posterior, foram projetadas e construídas duas modalidades distintas para coleta de imagens:

1. Desenvolvimento de uma **Estação de Coleta de Imagens** (ECI), com iluminação e cor de plano de fundo controlados.
2. Construção de uma **Rede de Distribuição de Ensaio** (RDE), não energizada e adaptada para facilitar a coleta de imagens.

4.1.1 Estação de Coleta de Imagens (ECI)

A *Estação de Coleta de Imagens* (ECI) é constituída pelos seguintes componentes principais: mesa giratória para ajustar o ângulo ϕ no intervalo de $[0, 360]$ graus e um eixo móvel para acoplamento de câmera, com uma inclinação de $\theta \in [0, 90]$ graus, conforme pode ser visualizado nas Figuras 4.3 e 4.4. Este aparato foi desenvolvido pelo aluno de Doutorado André Marotta, e encontra-se atualmente nas dependências da UFBA [53].

Para a etapa de aquisição de dados na ECI, foram coletados conjuntos de imagens de componentes isoladores de distribuição, íntegros e defeituosos. As características da base de dados gerada, tais como tipos de componentes, estado de conformidade, número de imagens e intervalos dos ângulos de ϕ e θ , foram estabelecidas de modo a representar, da forma mais aproximada possível, o nível de diversidade e estatísticas de imagens de componentes isoladores presentes em RPDA reais. A Figura 2.3 ilustra as referidas imagens coletadas para componentes íntegros (com etapa posterior de segmentação) para três ângulos θ distintos. As Figuras 4.5 e 4.6, por sua vez, exemplificam as imagens de componentes inconformes coletadas na ECI e os principais tipos de defeitos presentes neste *dataset*.

Considerando as bases de dados coletada para cada tipo de isolador, elas foram separadas em imagens de componentes íntegros, que compõe o *dataset* intitulado **DATA-INT**, e imagens de componentes defeituosos, com denominação de **DATA-DEF**, sendo que os tipos de defeitos foram apresentados na Seção 2.2 e estão presentes nas Figuras 4.5 e 4.6. Para compor essa base de dados, foram coletadas 2400 fotos no total (600 de cada componente) - 400 imagens para o isolador íntegro e 200 exemplos de componentes defeituosos. O modelo de câmera empregado nesta pes-



Figura 4.3: Estação de Coleta de Imagens (ECI).

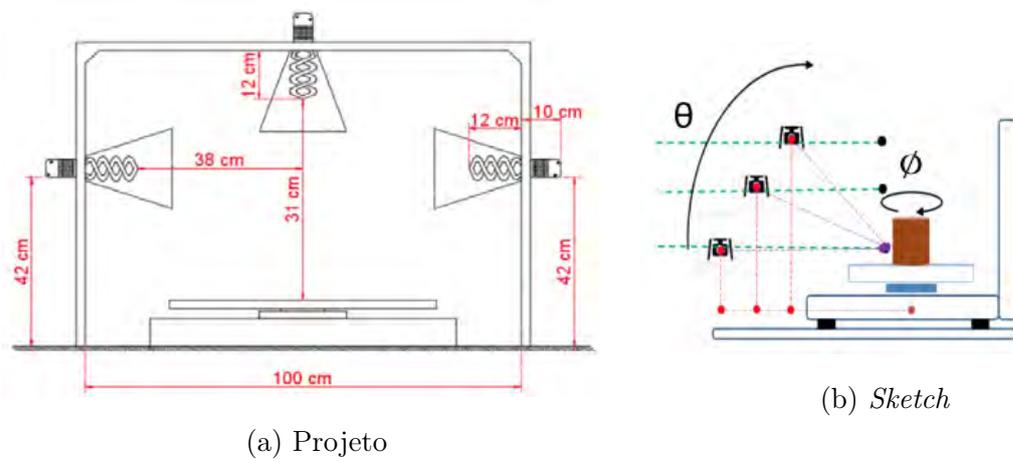


Figura 4.4: *Layout* da estação de coleta de imagens.

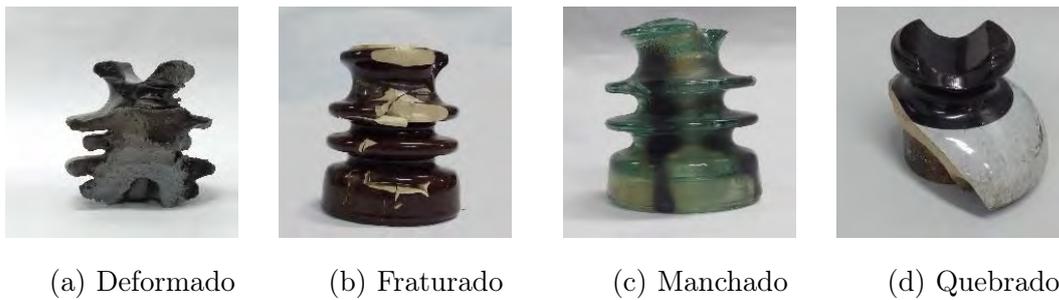


Figura 4.5: Exemplos de tipos de inconformidades coletadas na ECI.

quisa foi o K53b36, com dimensões das imagens de 4.632 x 2.608 pixels e resolução de 72 dpi.



Figura 4.6: Exemplos de componentes defeituosos coletados na ECI: (a) ICP; (b) IPC; (c) ICB; (d) IVV.

4.1.2 Rede de Distribuição de Ensaio (RDE)

No caso da *Rede de Distribuição de Ensaio* (RDE), o objetivo foi reproduzir condições semelhantes às encontradas nas redes primárias de distribuição aéreas reais. A RDE foi construída em ambiente externo e possui um comprimento de 30 m, sendo composta por três postes, cada um com 2,2 m de altura. A configuração da estrutura central é do tipo N1 e sua posição é equidistante de ambas as estruturas de fim de linha (tipo N3) [48]. Os componentes isoladores, a rede de distribuição de ensaios, o poste central e o design das estruturas podem ser visualizados nas Figuras 4.7, 4.8, 4.9 e 4.10 [33].

O procedimento de coleta de imagens adotado para a RDE teve como foco a reprodução de condições realistas. Durante a etapa de aquisição de dados, foram coletadas imagens com distâncias, ângulos, iluminações e planos de fundos não controlados. Para cada tipo de isolador, foram obtidas 520 fotografias, totalizando 2080 imagens. Das 520 imagens de cada componente, 400 são de componentes intactos e 120 de componentes defeituosos. Dessa forma, também foram

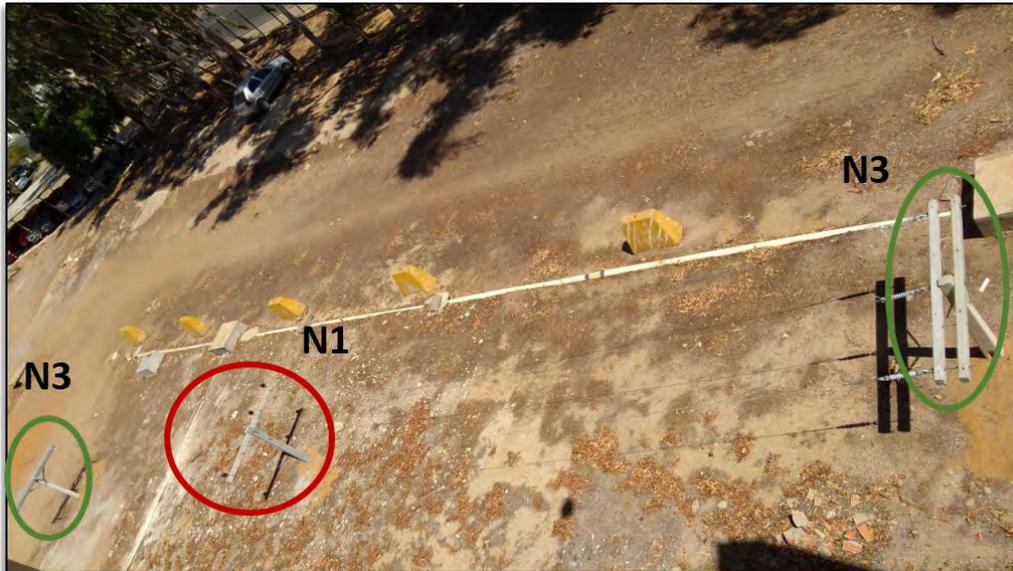
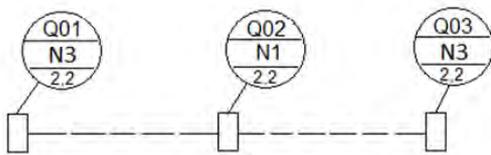


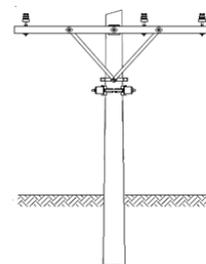
Figura 4.7: Fotografia (vista superior) da Rede de Distribuição de Ensaio (RDE).



Figura 4.8: Fotografia do poste central da RDE.



(a) Projeto do trecho de RDE.



(b) Ilustração do poste central.

Figura 4.9: Esquemático da Rede de Distribuição de Ensaio (RDE).

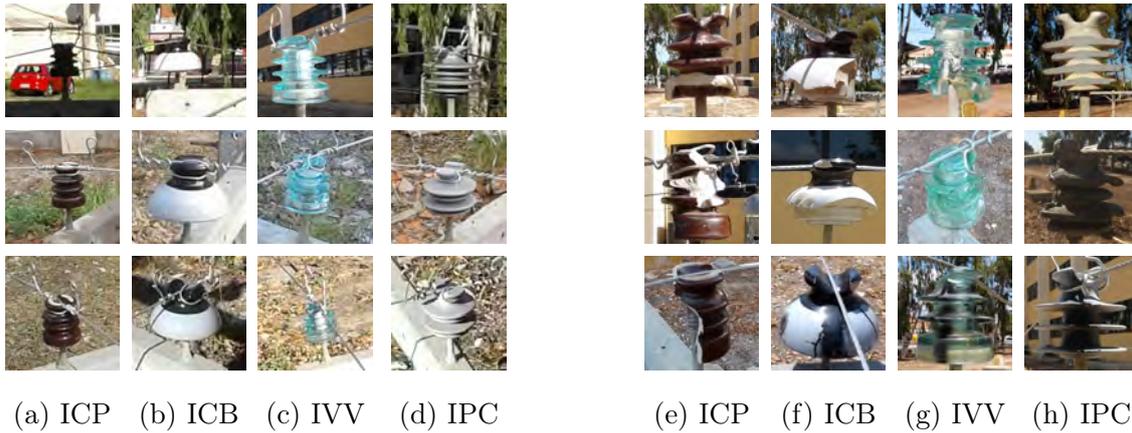


Figura 4.10: Imagens de Isoladores íntegros (a, b, c e d) e defeituosos (e, f, g e h) coletadas em diferentes ângulos.

produzidos os *datasets* do tipo DATA-INT e DATA-DEF. A Figura 4.10 exemplifica imagens dos *datasets* de componentes íntegros e defeituosos coletados na RDE. A Tabela 4.1 apresenta os quantitativos de imagens coletadas para cada classe de componente, sendo que 69,4% das amostras retratam componentes íntegros (INT) e 30,6% apresentam algum tipo de defeito (DEF). As classes de defeitos identificadas estão representadas nas Figuras 4.5 e 4.6. Os conjuntos de dados mencionados, coletados tanto na ECI quanto na RDD, estão disponíveis em http://www.dee.eng.ufba.br/dslab/index.php/opdl_dataset/.

Tabela 4.1: Tabela de Imagens Coletadas para os isoladores de distribuição íntegros (INT) e defeituosos (DEF).

Componentes	ECI - INT	ECI - DEF	RDE - INT	RDE - DEF
ICP	400	200	400	120
ICB	400	200	400	120
IPL	400	200	400	120
IVV	400	200	400	120

4.2 Gerador de Imagens Artificiais (GIA)

Com a finalidade de reduzir a complexidade do *dataset* de treinamento e facilitar as etapas de coleta de imagens, foi proposta uma nova metodologia direcionada à criação de imagens sintéticas a partir de *datasets* adquiridos em estúdios fotográficos. Objetivou-se desenvolver um Gerador de Imagens Artificiais (GIA) que seja alimentado com fotografias provenientes da ECI e produza imagens com características realistas, inerentes às redes de distribuição aéreas. Esta etapa não só envolveu as transformações usuais de *data augmentation*, como rotação, translação e alteração de brilho, mas também incluiu tarefas adicionais ao fluxo padrão. Foram implementados estágios de segmentação, adição de planos de fundo e sobreposição de detalhes realísticos, tais como amarrações metálicas e condutores livres. Um diagrama do GIA é apresentado na Figura 4.11.

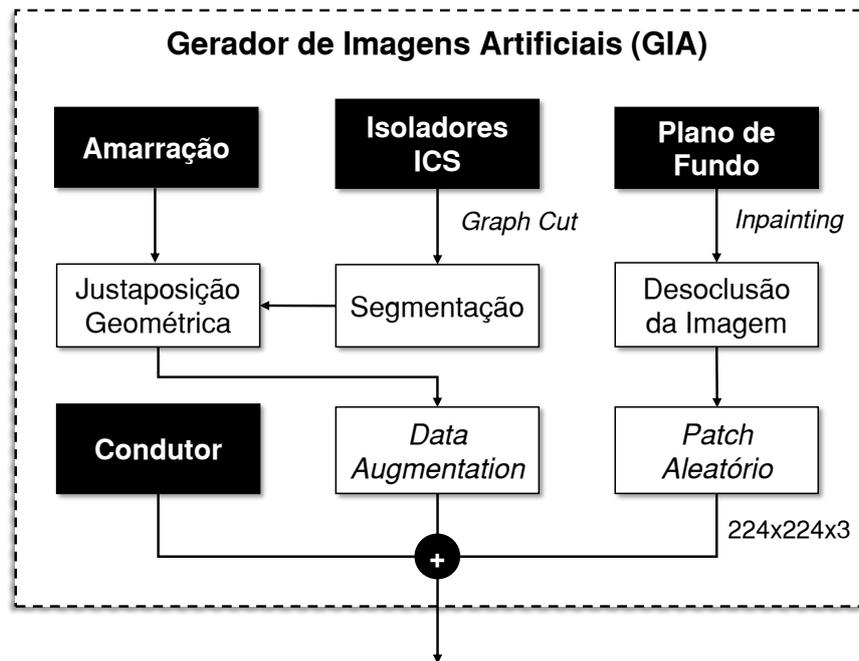


Figura 4.11: Diagrama de fluxo da informação do Gerador de Imagens Artificiais (GIA).

Segmentação interativa: Considerando a etapa de aquisição de imagens na ECI, os componentes da RDA são fotografados de forma isolada, com iluminação e planos de fundo controlados, conforme pode ser visualizado nas Figuras 2.3, 4.4a, 4.5

e 4.6. Este procedimento se mostrou problemático, tendo em vista que imagens de RPDA reais podem apresentar sobreposição de componentes menores (amarrações e condutores), além de possuir grandes variações de planos de fundo, provenientes de paisagens urbanas e rurais. Esta situação dificulta a extrapolação do desempenho do conjunto de testes para avaliar ambientes mais realistas.

A fim de superar esses problemas, torna-se necessário viabilizar a introdução artificial de detalhes realísticos a partir da inclusão de diferentes tipos de planos de fundo. Com este propósito, foi criado um novo estágio para segmentação das imagens da ECI. Isoladores, amarrações e condutores livres foram segmentados com a finalidade de isolar somente as informações de interesse nas imagens. Esta etapa de segmentação foi realizada por meio do algoritmo iterativo *Graph Cut* [197]. A Figura 4.12 ilustra exemplos de imagens de isoladores coletadas na ECI e respectivas máscaras geradas e a Figura 4.13, por sua vez, mostra exemplos de imagens já segmentadas de amarrações e condutores livres.



Figura 4.12: Imagem do dataset ECI (parte superior) e máscaras produzidas com a técnica *Graph Cut* (parte inferior).

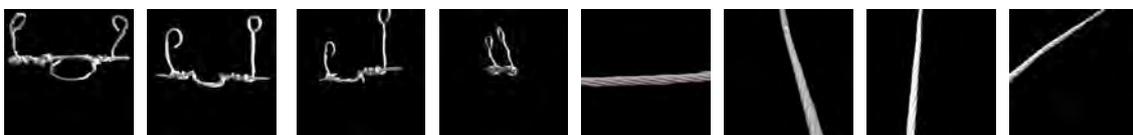


Figura 4.13: Exemplos de imagens de amarrações metálicas e condutores livres segmentadas pela técnica *Graph Cut*.

Desocclusão de Imagens: Para melhorar o desempenho do gerador de imagens, foi avaliada uma forma de inclusão de fragmentos de imagens (em inglês, *patches*) de planos de fundo a partir de imagens com estruturas de RPDA reais. Essas estruturas possuem características visuais importantes (a exemplo de cor, formato

e textura) que estão sempre presentes em ambientes reais. Contudo, podem ocorrer casos em que o fragmento de imagem, extraído de forma aleatória, crie duplicidades de componentes no *dataset* de treino. Um exemplo possível é a existência de dois ou mais isoladores na mesma imagem criada pelo GIA, sendo que os componentes indesejados estavam inicialmente presentes no plano de fundo.

Dessa forma, foi utilizada uma técnica de desocclusão de imagens, também conhecida como *inpainting* [198, 199], que possibilita a utilização de imagens de RPDA para geração de planos de fundo extraíndo as informações indesejadas. O método utilizado foi o *Fast Marching Method* (FMM), proposto por Alexandru Telea [200]. A Figura 4.14 ilustra o processo de desocclusão das imagens de isoladores presentes em uma fotografia utilizada para extração de planos de fundo. A partir da imagem real e respectiva máscara (Figuras 4.14a e 4.14b), é possível gerar uma nova imagem com informações que apresentam menor nível de interferência no processo de decisão dos classificadores utilizados (ver Figura 4.14c).

Detalhamento realístico e *data augmentation*: Após o procedimento de segmentação e desocclusão de imagens, foi implementado um processo de *data augmentation* e detalhamento realístico, constituído pelas seguintes subetapas:

1. Para 50% das imagens dos isoladores, exemplificados na Figura 4.12, foi introduzida uma imagem aleatória de amarração, posicionando-a, dentro do retângulo periférico *bounding box* do componente segmentado, entre o centro horizontal e um terço da parte superior da delimitação vertical.
2. Transformações convencionais de *data augmentation* foram então aplicadas nas imagens dos isoladores segmentados (com ou sem a presença de amarrações), tendo sido considerados seis parâmetros de transformação modificados de forma aleatórias, tendo como valores máximos: 15° de rotação, 15% de movimentos de translação, 15% de brilho, 30% de cisalhamento, 50% de inversão horizontal e 15% de mudanças de escala (ampliação e redução).
3. Seguindo uma rotina probabilística, condutores elétricos foram introduzidos em 50% das imagens. Neste estágio, 12,5% das imagens combinam condutores e amarração metálica, e nos 37,5% restantes, ocorre a inclusão de condutores (máximo de 2) nas imagens que não possuem amarração. Estes valores foram



(a) Imagem de cenário realista.



(b) Máscara de regiões da imagem.



(c) Imagem alterada.

Figura 4.14: Imagem de cenário alterada pela técnica de Inpainting.

determinados de forma empírica, levando em consideração o desempenho de treinamento dos modelos computacionais e algumas características dos ambientes realistas, tais como os diferentes ângulos e posições possíveis para as tarefas de coleta de imagens por VANTs.

4. Fragmentos de imagens de cenários, retirados de forma aleatória de fotografias ao ar livre, foram então introduzidos como planos de fundo para composição final de todas as imagens.

Esta metodologia de geração de imagens é inserida dinamicamente na rotina de aprendizagem do modelo computacional e se utiliza de parâmetros que são alterados de forma aleatória ao longo do processamento da informação. Deste modo, ela é repetida iterativamente em cada uma das épocas utilizadas para o treinamento dos

algoritmos. Por exemplo, para 100 épocas de simulação, de um número de 480 imagens do *dataset* real, é gerado um total de 48.000 imagens modificadas para fins de treinamento.

4.2.1 Segmentação Semântica com emprego de GIA

O algoritmo de segmentação implementado é uma adaptação da arquitetura de aprendizado profundo, intitulada de U-Net. Ela foi apresentada por Ronneberger et al [165], no ano de 2015. Uma representação simplificada dessa rede convolutiva pode ser vista na Figura 4.15. Este modelo computacional é constituído das etapas de subamostragem, que é composta por camadas do tipo *max-pooling* e de sobre-amostragem - estágios de convoluções transpostas. Na etapa de subamostragem, o algoritmo possui cinco blocos convolucionais adicionados a estágios de *dropout*. Cada bloco possui duas camadas convolucionais com filtros 3×3 e *stride* 1. O detalhamento completo deste algoritmo de segmentação foi apresentado na Seção 3.8.

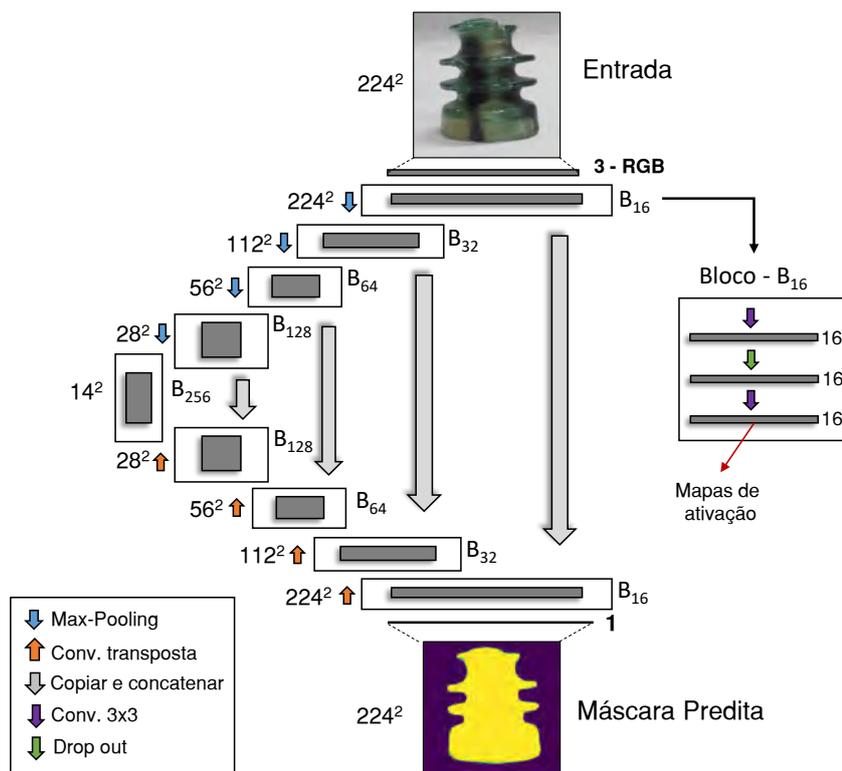


Figura 4.15: Algoritmo de segmentação semântica U-Net adaptado.

Através desta arquitetura de rede convolutiva, foi elaborado um conjunto de estratégias com a implementação do gerador de imagens artificiais com a finalidade de

otimizar a etapa de treinamento da rede convolutiva e, conseqüentemente, os resultados de segmentação dos componentes isoladores de RPDA, a partir das imagens coletados em estúdio fotográfico (ECI). O diagrama da metodologia implementada é apresentado na Figura 4.16.

No fluxograma supracitado, é possível identificar que o Gerador de Imagens Artificial foi empregado para alterar digitalmente as imagens de entrada (\mathbf{X}_{im}) do modelo computacional ao longo da rotina de treinamento. Para este algoritmo de segmentação, o GIA foi utilizado de forma limitada, realizando somente as etapas de *data augmentation* de \mathbf{X}_{im} e inserindo digitalmente os fragmentos de imagens de planos de fundo. O objetivo principal é deixar o algoritmo de segmentação menos susceptível a interferências de contexto, proporcionando um certo nível de imunidade aos ruídos de plano de fundo.

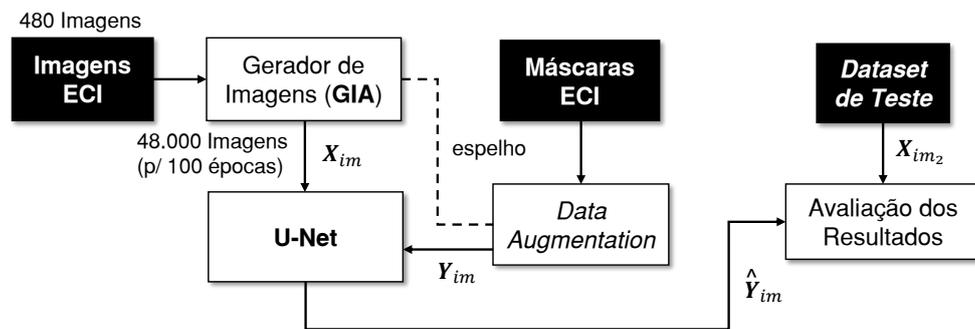


Figura 4.16: Diagrama da metodologia proposta de segmentação semântica através do emprego do Gerador de Imagens Artificiais (GIA).

Para este caso de treinamento supervisionado com emprego de máscaras, foi adicionada uma nova etapa de *data augmentation* no interior da rotina de treinamento. Esta nova tarefa possui a função de alterar as máscaras usadas como rótulo, especificadas como \mathbf{Y}_{im} . Vale ressaltar que a rotina de *data augmentation* aplicada em \mathbf{Y}_{im} é um “espelho” (possui simetria) em relação à mesma alteração realizada pelo GIA, ou seja, todas as modificações de posição, escala ou formato, realizadas nos componentes em \mathbf{X}_{im} , devem ser igualmente aplicadas nos rótulos de saída \mathbf{Y}_{im} (ver Figura 4.17). Na etapa de testes, as máscaras previstas pelo modelo ($\hat{\mathbf{Y}}_{im}$) são avaliadas para imagens provenientes de um novo dataset de testes (\mathbf{X}_{rd}). Os algoritmos propostos para o gerador artificial de imagens estão disponíveis em http://www.dee.eng.ufba.br/dslab/index.php/opdl_dataset/.

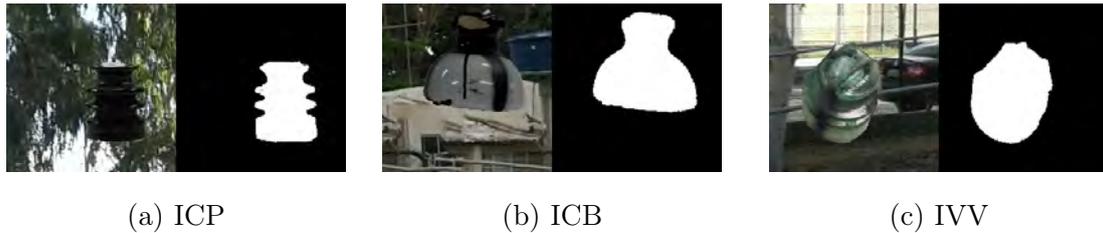


Figura 4.17: Exemplos de três pares de treinamento gerados para o algoritmo de segmentação semântica: imagem de entrada (\mathbf{X}_{im} - imagem à esquerda) e máscara de saída para o *label* (\mathbf{Y}_{im} - imagem à direita).

4.3 Aplicações em RPDA

Dentre as atividades que envolvem a etapa experimental, foi elaborado um conjunto de técnicas inteligentes com o propósito de compor **sistemas de reconhecimento de padrões**. As principais atividades realizadas podem ser enumeradas como: (1) pré-processamento da informação; (2) extração de atributos (geométricos e paramétricos), segmentação semântica, (3) geração artificial de imagens e (4) classificação de padrões. Em suma, estas atividades combinam aplicações de Processamento Digital de Imagem (PDI) e de aprendizado de máquina para o desenvolvimento de técnicas inteligentes de classificação de imagens de componentes isoladores de redes de distribuição de energia elétrica.

A partir dessa perspectiva, buscou-se analisar o desempenho de diferentes algoritmos inteligentes supervisionados e técnicas de PDI. Para este propósito, foram avaliados exemplares de modelos inteligentes, a exemplo de classificadores clássicos, de algoritmos de aprendizagem profunda e de composições híbridas que correspondem à combinação entre mais de uma técnica inteligente. Nas subseções seguintes, serão apresentadas as principais abordagens experimentais implementadas nesta pesquisa. Como forma de roteirizar esta etapa, será seguida uma cadência de atividades bem definida, que podem ser apresentadas como:

1. **Avaliação de desempenho de algoritmos inteligentes tradicionais** para classificação de imagens de componentes isoladores a partir de conjuntos de atributos geométricos [31, 32, 53].
2. **Implementação de técnicas híbridas para classificação de imagens** de isoladores a partir da combinação de redes convolutivas e de algoritmos

tradicionais [34].

3. **Desenvolvimento de técnicas de aprendizagem profunda** para classificação do modelo de isolador e identificação do seu respectivo *status* de conformidade [33].

Em conjunto com as técnicas citadas, foram empregadas diferentes estratégias direcionadas à otimização do processo de aprendizagem dos modelos computacionais propostos, a exemplo da configuração de *multi-task learning* e das técnicas de *transfer learning*, *data augmentation* e *background invariance*, esta última empregada com o auxílio de algoritmos de geração de imagens artificiais.

4.3.1 Classificação por Atributos Geométricos

Os isoladores de distribuição de energia elétrica, a depender do fabricante e do tipo de material construtivo, usualmente apresentam características visuais distintas, a exemplo de tamanho e *design*, conforme pode ser observado nas Figuras 2.3, 4.5 e 4.6. A partir destas particularidades visuais, pode-se inferir que existem potenciais linhas de investigação para aplicações inteligentes que utilizam *assinaturas* geométricas, inerentes às imagens destes dispositivos. Deste modo, uma das abordagens experimentais conduzidas nesta pesquisa consistiu na análise de *parâmetros geométricos* de imagens bidimensionais de componentes isoladores, com alterações de posição e *status* de conformidade dos dispositivos. O ponto de mérito desta investigação está na resposta da hipótese levantada: se é viável que um conjunto de informações geométricas de componentes isoladores seja utilizado para o desenvolvimento de sistemas de classificação inteligentes supervisionados. A Figura 4.18 apresenta, de forma esquemática, o diagrama de procedimentos realizados para este estágio da pesquisa.

O fluxograma apresentado se inicia com a etapa de seleção da base de dados de estudo e depois segue a seguinte cadência de atividades: **(1)** segmentação das imagens dos componentes; **(2)** extração de atributos geométricos; **(3)** cálculo de vetor de parâmetros (\mathbf{x}_g); **(4)** pré-processamento das informações e **(5)** desenvolvimento/implementação de diferentes tipos de classificadores inteligentes. Como síntese deste estágio experimental, os atributos geométricos, originados de máscaras

produzidas por duas técnicas de segmentação distintas, são utilizados para compor um vetor de atributos que servirá como entrada para os modelos computacionais propostos. No que tange à otimização de desempenho dos modelos computacionais, foi realizado um estudo comparativo entre técnicas inteligentes comumente empregadas em sistemas de classificação, tais como *Hidden Markov Model* (HMM) [109], *Multi-Layer Perceptron* (MLP), *Support Vector Machine* (SVM) [98], *Random Forest* [108] e *k*-Nearest Neighbor (*k*NN) [97].

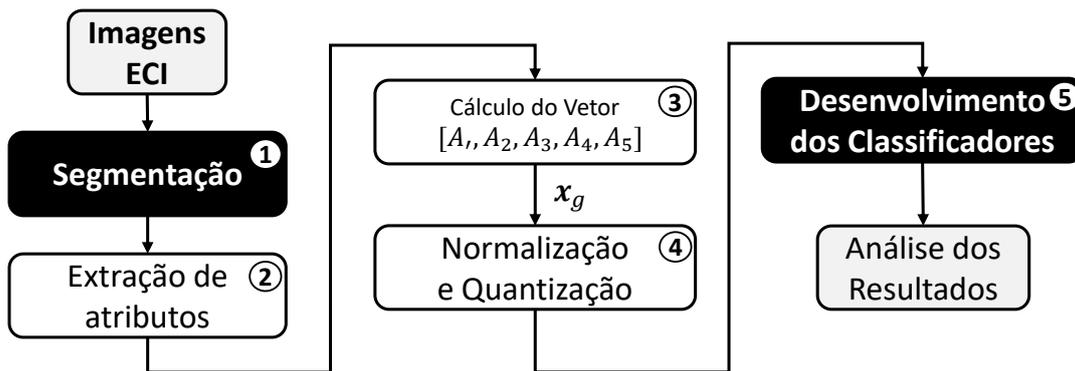


Figura 4.18: Diagrama da metodologia proposta.

Na etapa de seleção da base de dados de treinamento e teste dos modelos computacionais, foram utilizados os *datasets* DATA-INT e DATA-DEF, adquiridos com a ECI, referentes a três tipos de componentes isoladores que apresentam materiais e configurações visuais distintas, sendo eles: Isolador Cerâmico Bicolor (ICB); Isolador de Vidro Verde (IVV) e Isolador Polimérico Cinza (IPC). Para a etapa de segmentação de imagens, foram avaliadas duas técnicas distintas. A primeira delas é uma abordagem clássica de PDI, intitulada de processo de limiarização múltipla de Otsu [201]. A segunda abordagem é uma adaptação do algoritmo U-Net [165], com emprego do Gerador de Imagens Artificiais (GIA), conforme apresentado na Subseção 4.2.1. Apesar de U-Net ser empregada para segmentação de imagens com controle de iluminação e planos de fundo, foi identificado que o desempenho de segmentação apresenta ganhos caso sejam inseridos contextos (planos de fundo) com informações não controladas - a exemplo de imagens provenientes de ambientes externos. A técnica de segmentação que se utiliza de processamento digital de imagens será apresentada a seguir.

Segmentação por limiarização múltipla de Otsu: Pode ser entendida como uma técnica clássica de segmentação pertencente ao campo de PDI, e apresenta um baixo custo computacional para implementação [113]. Ela ainda é empregada em casos onde os planos de fundo não são muito desafiadores [31]. Esta técnica utiliza um conjunto de estágios de pré-processamento, filtragem e segmentação. As imagens coletadas são convertidas para níveis de cinza e, para redução do nível de ruído, realiza-se também um processo de filtragem. Como etapa seguinte, um algoritmo adaptativo de filtro de Wiener é aplicado. Este algoritmo implementa uma abordagem *pixel-a-pixel* baseada em estatísticas estimadas em vizinhanças. O filtro local é, na verdade, um filtro passa-baixas variante no espaço bidimensional (coordenadas u e v), cuja transformada de Fourier é dada por [202]:

$$H(u, v) = \frac{S_f(u, v)}{S_f(u, v) + S_\eta(u, v)} \quad (4.1)$$

onde $S_f(u, v)$ e $S_\eta(u, v)$ são a densidade espectral de potência do ruído e da imagem degradada, respectivamente. Dentro da região local (vizinhança), a imagem degradada (filtrada) é assumida como estacionária, com média m_f e variância σ_f^2 . O algoritmo adaptativo então estima a média e variância locais, definindo uma vizinhança Δ de dimensões $N \times M$ ao redor de cada pixel na imagem degradada, chamada de $F(n_1, n_2)$, dadas por:

$$m_f = \frac{1}{NM} \sum_{n_1, n_2 \in \Delta} F(n_1, n_2) \quad (4.2)$$

$$\sigma_f^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \Delta} F^2(n_1, n_2) - m_f \quad (4.3)$$

onde n_1 e n_2 são os índices de cada *pixel* da imagem transformada. A imagem restaurada \hat{F} é obtida das regiões restauradas localmente e dada por:

$$\hat{F}(n_1, n_2) = m_f + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_\eta^2} (F(n_1, n_2) - m_f) \quad (4.4)$$

A variância do ruído é estimada a partir da média de todas as variâncias σ_f^2 . Utiliza-se neste trabalho uma vizinhança de tamanho 5×5 [113].

Após a filtragem de Wiener, aplica-se um algoritmo de segmentação baseado no processo de limiarização múltipla de Otsu [201]. Este algoritmo pode ser apresentado como uma técnica de limiarização, que objetiva determinar um valor ideal de

limiar (*threshold*) para separar os elementos de uma imagens em dois aglomerados (*clusters*), atribuindo a cor preta ou branca para cada um deles. Para o caso deste estudo, são obtidos dois limiares de segmentação para as imagens dos componentes isoladores, uma vez que elas apresentam contraste relativamente alto, mas cores muito próximas entre si. A partir dos dois limiares, usa-se o de menor valor, tendo em vista que componentes apresentam níveis de cinza menores que os planos de fundo considerados.

Extração de atributos geométricos: Após a etapa de segmentação, foi extraído um conjunto de atributos geométricos das regiões destacadas presentes nas máscaras. Estes atributos são listados a seguir: Área (A_r), Perímetro (P), Raio Menor (R_{menor}), Raio Maior (R_{maior}), Diâmetro Equivalente (D_e), Excentricidade (E) e Extensão (E_x). Os referidos atributos, ilustrados na Figura 4.19, foram selecionados em função das *assinaturas geométricas* pertencentes aos objetos avaliados, tendo em vista que foram identificadas alterações quantitativas perceptíveis nos valores destes sete parâmetros em função da classe do componente isolador. Logo em seguida, foram calculados 5 parâmetros para compor um vetor de atributos, $\mathbf{G}_a = [A_1, A_2, A_3, A_4, A_5]$. As equações adotadas para o cálculo deste vetor podem ser visualizadas nas equações a seguir:

$$A_1 = (P)^2 / (100 \cdot A_r) \quad (4.5)$$

$$A_2 = E \quad (4.6)$$

$$A_3 = D_e / P \quad (4.7)$$

$$A_4 = R_{menor} / R_{maior} \quad (4.8)$$

$$A_5 = E_x \quad (4.9)$$

Os termos apresentados nas equações 4.5 a 4.9 também foram estabelecidos a partir da avaliações experimentais dos desempenhos de classificação dos modelos computacionais. As informações obtidas para os vetores \mathbf{G}_a foram então normalizadas para que os valores de cada atributo ficassem distribuídos dentro de um intervalo pré-determinado entre $[0,1]$. A transformação linear ocorreu através da

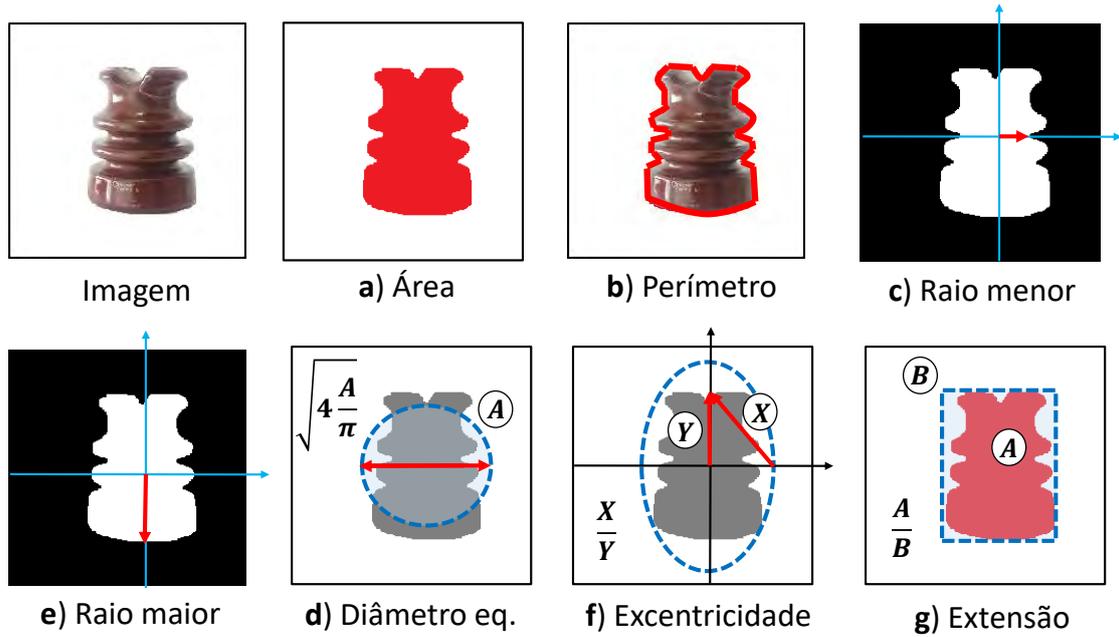


Figura 4.19: Atributos geométricos da imagem (máscara) de isolador de distribuição do tipo ICP.

seguinte equação:

$$X_{i,[0,1]} = \frac{X_i - X_{min}}{X_{max} - X_{min}}, \quad (4.10)$$

onde X_i é o valor original do i -ésimo dado, X_{min} é o menor valor observado, X_{max} é o maior valor observado na base de dados e $X_{i,[0,1]}$ define o valor do dado X_i normalizado no intervalo $[0, 1]$.

Classificação por cadeia de HMMs: Como etapa preliminar para implementação desta técnica, o vetor de atributos normalizado é quantizado para uma faixa de valores inteiros, contendo um número finito, definido em $Z = 1, 2, \dots, N_m$, onde N_m é um número inteiro que pode ser definido pelo usuário. Obtém-se, então, um Vetor de Parâmetros Quantizado $\mathbf{G}_{Qa} = [L_1, L_2, L_3, L_4, L_5]$. Como uma estratégia não convencional de classificação de imagens, proposta nesta pesquisa, foi realizado o treinamento de três Modelos Ocultos de Markov unidimensionais, configurados em paralelo, conforme ilustrado na Figura 4.20, para realizar a classificação de vetores de atributos geométricos, \mathbf{G}_{Qa} , do mesmo modo como ocorre com um sinal do tipo sequencial [32].

Considerando os dois problemas fundamentais para resolução de HMMs, descri-

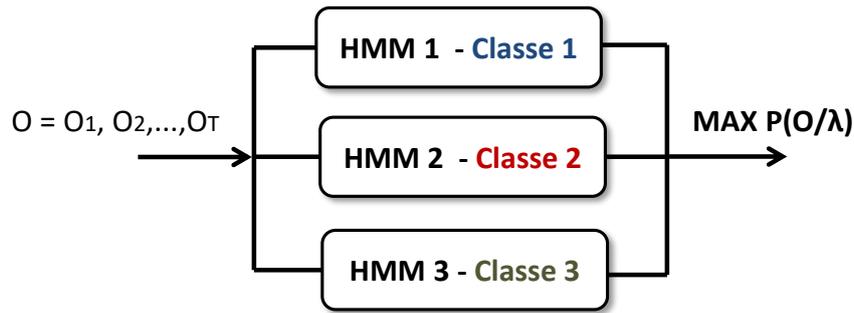


Figura 4.20: Arranjo dos 3 Modelos Ocultos de Markov (HMMs) empregados para classificação de parâmetros geométricos de isoladores de distribuição.

tos na Subseção 3.2.6, o algoritmo *Forward-Backward* foi empregado para resolução do problema 1 (cálculo do valor de $P(O|\lambda)$). Para o problema 2, relativo à estimação dos parâmetros do modelo ($\lambda = (\mathbf{A}, \mathbf{B}, \pi)$), foi aplicado o algoritmo de Baum-Welch. Então, de posse da base de dados dos vetores de parâmetros quantizados, foram realizados os treinamentos das três HMMs, cada uma delas representando uma das três classes dos isoladores avaliados.

A classe de isolador é escolhida conforme o valor de $P(O|\lambda)$ de cada HMM. O modelo que obtém o maior valor de probabilidade $Max_{P(O|\lambda)}$, para uma dada sequência O , é definido como a classe de saída. Os números de Estados Ocultos (N_e) e de níveis de quantização das Variáveis de Observação (N_V) podem ser manipulados pelo usuário, possibilitando a verificação e escolha de um valor otimizado de cada HMM para aplicação.

4.3.2 Classificação por Técnicas Híbridas

Os isoladores de distribuição com defeitos visíveis, conforme exemplificado na Figura 4.6, podem apresentar severas alterações em suas características visuais, a exemplo de cor, formato e/ou textura, que podem reduzir a acurácia de algoritmos de classificação de imagens – principalmente quando as técnicas de identificação atuam de forma restrita à avaliação de parâmetros geométricos. Deste modo, com a finalidade de explorar a aprendizagem profunda para o desenvolvimento de um estágio alternativo de extração de características, foi conduzido um estudo pautado na composição de diferentes *configurações híbridas* de algoritmos inteligentes de classificação. Um ponto de investigação específico que será explorado neste estágio da

pesquisa consiste em avaliar se um algoritmo inteligente treinado com componentes íntegros é capaz de classificar corretamente conjuntos de imagens pertencentes a componentes defeituosos.

Outro objetivo importante assumido nesta pesquisa foi identificar se duas técnicas inteligentes combinadas podem obter melhores desempenhos de classificação de imagens de isoladores do que um único tipo de abordagem empregada individualmente. Um dos pontos de mérito consiste na viabilidade de uso de técnicas clássicas de aprendizagem supervisionada sem que haja necessidade de uma etapa prévia de segmentação. Em síntese, este processo pode ser entendido como uma abordagem alternativa à extração de parâmetros geométricos, apresentada na subseção anterior, com adição de alguns atributos que podem ser relevantes para esta pesquisa, a exemplo da sensibilidade da rede convolutiva a informações de cor e textura [203]. As referidas características visuais das imagens possuem potencial de contribuir nos estágios de classificação dos modelos computacionais. Como tarefa adicional, foram implementadas técnicas de transferência de conhecimento com o propósito de ampliar a capacidade de generalização dos classificadores [91]. A Figura 4.21 apresenta diagrama de procedimentos propostos neste estágio da pesquisa.

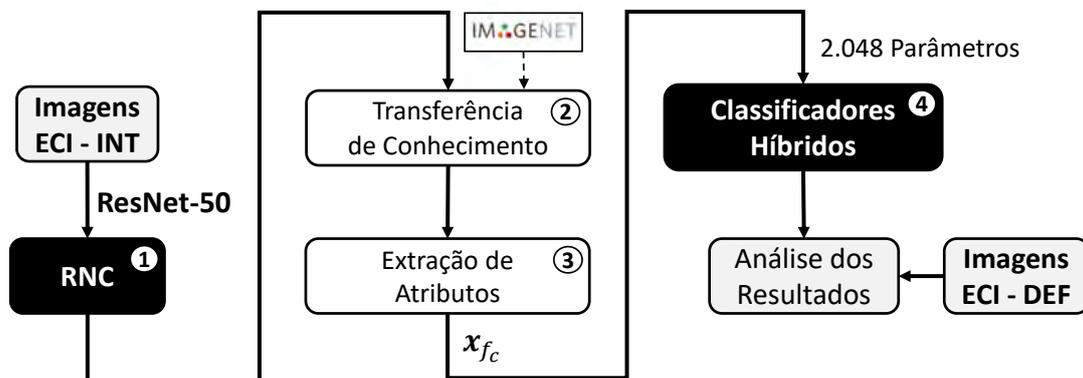
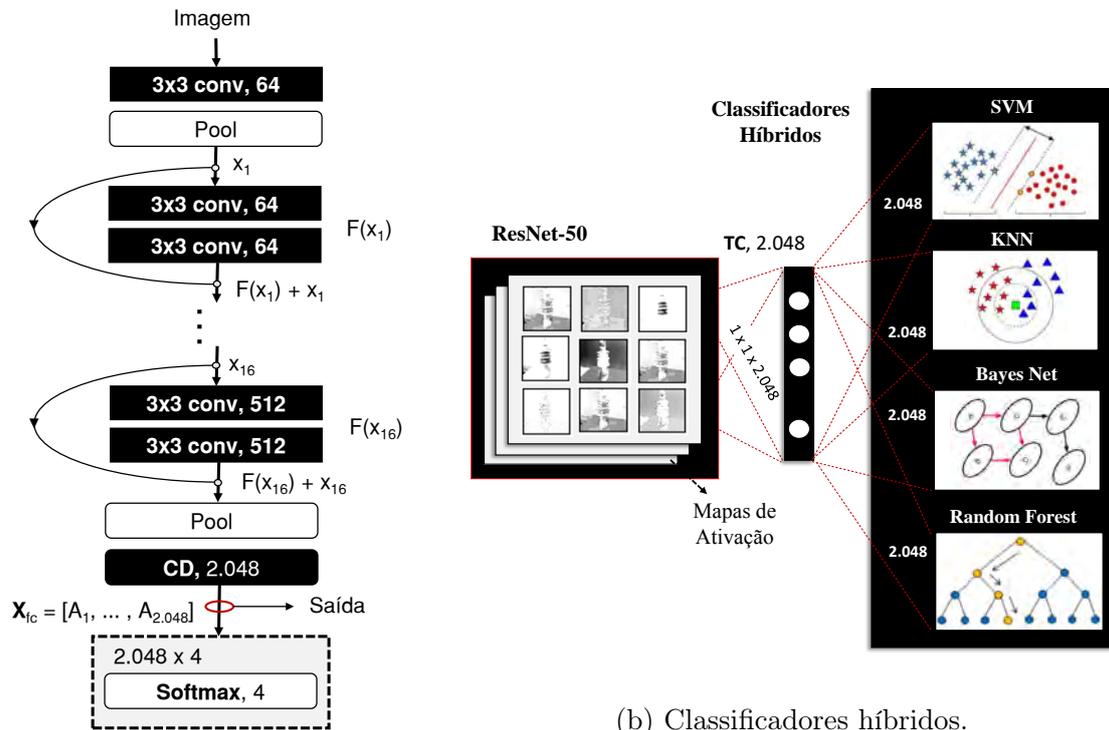


Figura 4.21: Diagrama da metodologia proposta.

O emprego de técnicas híbridas foi focado no ganho potencial de desempenho proporcionado pela combinação de duas ou mais técnicas de aprendizado de máquina [204, 177, 176]. Como primeiro estágio, utiliza-se uma rede neural convolucional para fornecer atributos para o treinamento de outros algoritmos inteligentes. Com este propósito, foi realizada uma sequência de atividades, nomeadamente:

(1) implementação de classificador de aprendizagem profunda; (2) emprego de técnica de transferência de conhecimento; (3) extração de vetor de atributos de treinamento e (4) implementação de um segundo estágio de classificação composto por algoritmos tradicionais - *Support Vector Machine (SVM)*, *k-Nearest Neighbors (kNN)*, *Random Forest* e *Bayes Net*.

A arquitetura de aprendizagem profunda escolhida para a etapa de extração de atributos foi a ResNet-50 [183]. Esta rede convolutiva serviu como modelo base para os experimentos. A Figura 4.22a ilustra, de forma simplificada, a arquitetura desta RNC, juntamente com o vetor atributos \mathbf{X}_{fc} , que possui a utilidade de atuar como um gerador de parâmetros para o desenvolvimento de um novo estágio de treinamento. Cada imagem inserida na rede convolutiva compõe um novo conjunto de 2.048 atributos de rede da Camada Densa (CD), que são empregados como um dos vetores de entrada para os classificadores presentes no segundo estágio de treinamento, conforme ilustrado na Figura 4.22b.



(a) ResNet-50.

(b) Classificadores híbridos.

Figura 4.22: Arquitetura simplificada da ResNet-50 (a) e configuração dos classificadores híbridos (b).

Além disso, a última camada desta RNC foi substituída para representar as classes pertencentes aos conjuntos de dados estudados. A função de ativação escolhida para a referida camada foi do tipo softmax [205]. Para a fase experimental, foram treinados modelos utilizando conjuntos de dados de componentes intactos e seus respectivos testes foram realizados para um banco de dados composto apenas de componentes defeituosos - a partir do *dataset* proveniente da ECI. Levando em consideração que a rede convolucional também é sensível a alterações de cor e textura [203], também foram adicionadas nas simulações as imagens de componentes isoladores de materiais distintos que possuem o mesmo formato, a exemplo dos isoladores cerâmicos de pino (ICP) e dos isoladores de vidro verde (IVV). Ao final dos experimentos, foi avaliado um conjunto de métricas de desempenho e sugerido qual o modelo híbrido mais adequado para atender às demandas desta pesquisa.

4.3.3 Diagnóstico de defeitos por aprendizagem profunda

A aprendizagem profunda já se consolidou como uma das principais técnicas de aprendizado de máquina empregadas atualmente no campo de visão computacional, principalmente no que se refere à tarefas relacionadas à classificação de imagens [91]. Contudo, os algoritmos derivados destas metodologias ainda se mostram suscetíveis a fontes de interferência de planos de fundo, conforme foi demonstrado por Cruz et al. [35], quando abordou o tema vinculado ao *background invariance*.

Dentre as dificuldades relatadas no Capítulo 2 para o desenvolvimento de *Sistemas Inteligentes de Inspeção Visual* (SIV) de RPDA, dois pontos ficaram em evidência. O primeiro deles reside nos ambientes realistas onde se realizam as *inspeções visuais* (urbanos e rurais), que usualmente se mostram ricos em interferências de planos de fundo. A segunda condição indesejada se apresenta na dificuldade de geração de *datasets* de RPDA realistas e com um alto nível de generalização entre os diversos tipos de topologia de rede. Para este objeto de estudo, mostra-se relevante a concepção de uma forma eficiente de gerar base de dados mais complexas e diversificadas a partir de *datasets* que apresentem limitações estatísticas em relação à distribuição de probabilidade real de conjuntos de imagens de RPDA. Por exemplo, produzir um *dataset* realista de RPDA a partir de imagens de componentes coletadas em estúdio fotográfico, como é o caso da estação de coleta de imagens

(ECI) [53], apresentada na Subseção 4.1.

A partir das informações apresentadas nesta subseção, objetiva-se desenvolver uma metodologia inteligente que apresente as seguintes características fundamentais: (1) possuir alto nível de imunidade a fontes de ruído de planos de fundo (*background invariance*); (2) gerar base de dados mais diversificadas e semelhantes às estatísticas do domínio real do objeto de estudo a partir de um *dataset* limitado; (3) apresentar bons desempenhos de classificação de material/*design* de componente isoladores de distribuição, independente do *status* de conformidade e, finalmente, (4) ser capaz de identificar, de forma simultânea à classe do componente, se existe ou não a presença de alguma inconformidade visível – classificando o dispositivo como íntegro ou defeituoso. A metodologia proposta para esta etapa experimental é apresentada na Figura 4.23.

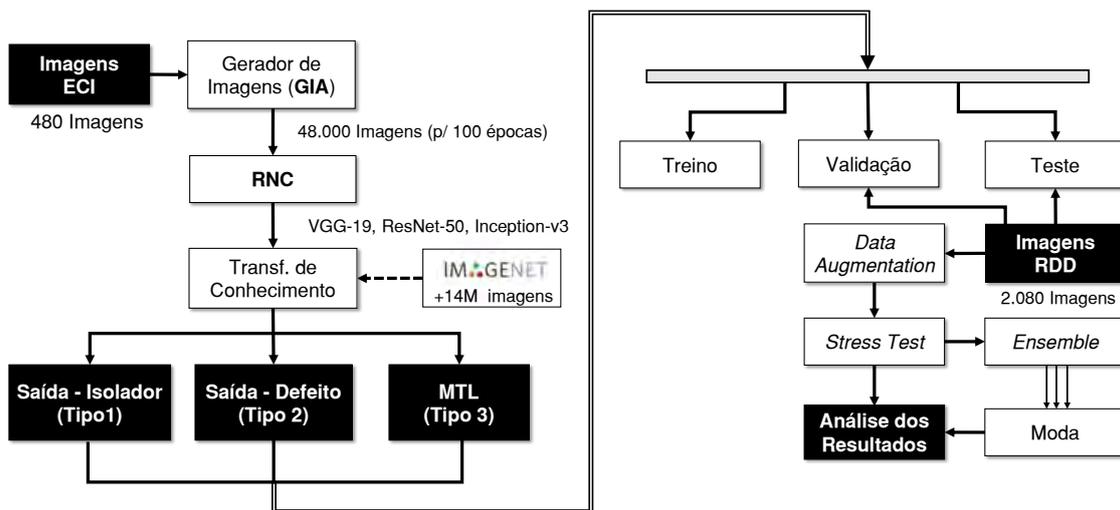


Figura 4.23: Diagrama da metodologia proposta para classificação de isoladores e defeitos.

Em termos gerais, o fluxograma apresentado consiste no desenvolvimento de três metodologias de aprendizagem profunda distintas. Diferentes arquiteturas de redes convolutivas são implementadas com o objetivo de discriminar a classe do isolador (denominada como tarefa do **Tipo 1**), a presença de defeitos (**Tipo 2**) e uma combinação de ambas as tarefas através da abordagem *Multi-Tasking Learning* (MTL) – **Tipo 3**. Durante a etapa de treinamento dos modelos, 480 imagens coletadas na ECI são então utilizadas como entrada para o GIA. A implementação do GIA permite elevar o número e o nível de diversidade das amostras de entrada,

executando-as de forma dinâmica no interior da rotina de treinamento. Considerando 100 épocas de treinamento, um total de 48.000 novas imagens serão geradas como *dataset* de entrada - de acordo com o que foi apresentado na Subseção 4.2.

Para as tarefas de validação e teste dos classificadores, foram utilizadas 2.080 imagens provenientes da RDE - 1.600 imagens de isoladores íntegros e 480 imagens de componentes defeituosos (ver Tabela 4.1). Esta base de dados também foi alterada digitalmente para simular cenários mais desafiadores, através de técnicas como *stress test* [206, 207] e *ensemble test* [208, 209]. Os detalhes dessas etapas de processamento da informação são fornecidos na Subseção 5.1. Os experimentos foram realizados com o emprego de três redes convolutivas bem conhecidas na literatura: **VGG-19**, **Inception-v3** e **ResNet-50**. Com o intuito de acelerar o processo de aprendizagem e otimizar os desempenhos de classificação, foram implementadas técnicas de transferência de conhecimento para realizar o treinamento dos modelos computacionais, de forma restrita a apenas algumas camadas das redes pré-treinadas, técnica denominada de *ajuste fino* [210, 120]. O *dataset* utilizado para etapa de pré-treinamento foi o ImageNet [91].

Para a etapa de ajuste fino das redes convolutivas, os parâmetros ajustáveis dos modelos computacionais foram congelados durante o treinamento, com exceção das camadas densas, e de duas camadas convolucionais para o caso específico do VGG-19. Além disso, a última camada foi substituída para classificar os tipos de material/*design* (softmax) e/ou classes de defeito (sigmoid). Para fins de ilustração, a arquitetura de adaptação do VGG-19 é mostrada na Figura 4.24. Os modelos treinados e respectivos códigos são disponibilizados juntamente com o conjunto de dados em http://www.dee.eng.ufba.br/dslab/index.php/opdl_dataset/.

Uma das investigações conduzidas neste estudo está relacionada à possível correlação entre as características visuais presentes nas diferentes classes de componentes isoladores e seus respectivos tipos de inconformidades. Deste modo, uma das arquiteturas avaliadas nesta etapa experimental consistiu no processamento simultâneo de dois tipos de informações distintas – configuração denominada de *Multi-Task learning*. Em síntese, mesmo quando o objetivo é uma tarefa de classificação específica, pode ser computacionalmente vantajoso aprender objetivos adicionais. A configuração (MTL) pode proporcionar ganhos de desempenho na tarefa principal [189, 188].

O *Multi-Task learning* se utiliza da relação intrínseca entre as tarefas de aprendizagem individuais para melhorar o desempenho da classificação. Portanto, nesta pesquisa, além das tarefas de aprendizagem individuais de material/*design*, e do *status* de conformidade, avaliou-se também o desempenho da aprendizagem multi-tarefas, que combina os dois tipos de saídas diferentes de forma simultânea [33].

Em cada caso das três formas de aprendizagem avaliadas, a função de custo (\mathcal{L}) a ser minimizada foi o erro de entropia-cruzada, com o emprego de mini-lotes. Para N observações e K classes, o modelo apresenta uma probabilidade $\hat{p}_{i,k}$ para cada observação $i \in \{1, \dots, N\}$ e classe $k \in \{1, \dots, K\}$, que é contrastada com o verdadeiro rótulo $y_{i,k}$. O resultado é 1 quando a observação i pertence à classe k e 0 caso contrário. Obtém-se então a seguinte equação:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{p}}) = -\frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K y_{i,k} \log(\hat{p}_{i,k}). \quad (4.11)$$

No caso da configuração MTL, a função custo final, \mathcal{L}' , combina ambas as funções custo das tarefas dos Tipos 1 e 2. Considerando ($\mathcal{L}^{(m)}$) para o material e ($\mathcal{L}^{(d)}$) para o defeito, o resultado de \mathcal{L}' é uma soma não ponderada,

$$\mathcal{L}'(\mathbf{y}, \hat{\mathbf{p}}) = \mathcal{L}^{(m)}(\mathbf{y}, \hat{\mathbf{p}}) + \mathcal{L}^{(d)}(\mathbf{y}, \hat{\mathbf{p}}). \quad (4.12)$$

A função Adam [95], apresentada na Subseção 3.2.1, com uma baixa taxa de aprendizagem, foi utilizada como otimizador da CNN. Taxas de aprendizagem superiores a 10^{-4} foram consideradas prejudiciais ao treinamento dos modelos computacionais. Em um segundo momento, com o propósito de fornecer evidências visuais de comportamentos internos apresentados pelos modelos computacionais trabalhados, foram implementadas rotinas para geração de *Mapas de Calor* (MCO) - do inglês, *heat-maps* [211].

Estes MCO possibilitam investigar características internas dos modelos de aprendizagem profunda através de suas respectivas sensibilidades para as imagens de entrada [212]. As matrizes relativas aos mapas de calor, definidas como $h_{x,y}$, são produzidas a partir do gradiente da soma das saídas das classes previstas, \hat{P}_k , em função de cada *pixel* de entrada e cores dos canais RGB, $X_{x,y,z}$, produzindo a equação:

$$g_{x,y,z} = \frac{\partial \sum_k \hat{P}_k}{\partial X_{x,y,z}}. \quad (4.13)$$

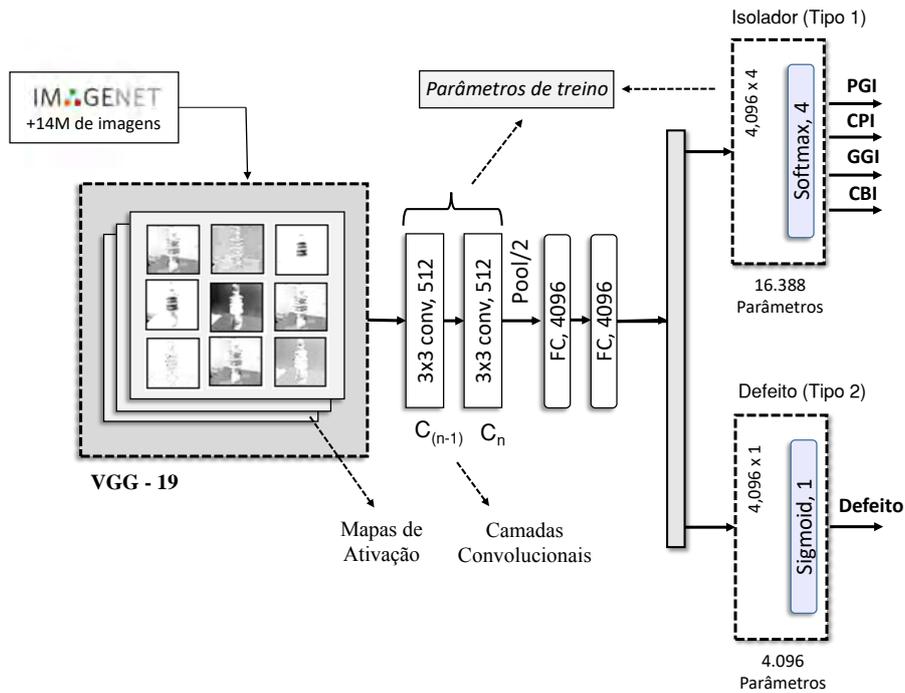


Figura 4.24: Arquitetura da RNC VGG-19 com a configuração *multi-task learning*.

A matriz gerada é monocromática e advém da escolha da cor absoluta máxima para cada *pixel*, com posterior etapa de normalização,

$$h_{x,y} = \frac{\max_c \|g_{r,y,c}\|}{\max_{x,y,c} \|g_{x,y,c}\|}. \quad (4.14)$$

Na etapa final, $h_{x,y}$ é mesclada à imagem de entrada na proporção de 40-60% do canal vermelho. Um exemplo de imagem resultante do mapa de calor é apresentado na Figura 4.25.



Figura 4.25: Exemplo de mapa de calor produzido para uma imagem de entrada.

4.4 Aplicações com Redes Generativas Adversárias

As Redes Generativas Adversárias (em inglês, *Generative Adversarial Networks* - GANs), conforme foi apresentado na Subseção 3.11, podem ser compreendidas como um arranjo de arquiteturas de aprendizagem profunda que vêm demonstrando atualmente desempenhos satisfatórios na geração de conjuntos de imagens artificiais diversificadas e com alto nível de realismo [21, 86]. De acordo com estas características, infere-se que estes arranjos de modelos inteligentes possuem alguns dos atributos necessários para serem adaptados a fim de compor aplicações semelhantes às técnicas de *data augmentation*. Deste modo, estas GANs seriam capazes de produzir novos conjuntos de imagens com estatísticas semelhantes ao domínio real do problema, mas que, ainda assim, contenham informações novas e relevantes para contribuir na aprendizagem de modelos supervisionados de classificação de imagens.

Outro ponto de investigação considerado como relevante nesta pesquisa consiste no aumento do nível de tolerância dos modelos computacionais a interferências de contexto nos *datasets* de imagens, a exemplo da incidência de planos de fundo com características não controladas - conforme mencionado na Subseção 2.4.1. Sabe-se que os algoritmos de aprendizagem profunda, a exemplo das redes neurais convolucionais, normalmente apresentam um nível moderado de vulnerabilidade a micro e macro irregularidades [35]. As micro irregularidades podem ser definidas como pequenas perturbações nas imagens que geralmente são imperceptíveis ao olho humano, mas possuem o potencial de gerar comportamentos imprevisíveis no desempenho de classificação de modelos inteligentes. Como exemplo, este fator tende a fragilizar as redes convolutivas a ataques adversários. Para estes casos, foram propostas algumas estratégias de solução [213].

Quanto as macro irregularidades, este tipo de interferência pode ser explicada como alterações visuais presentes nas imagens que são perceptíveis ao olho humano, mas que não se sobrepõe aos objetos de interesse [35]. Assumindo que algumas propriedades estatísticas dos *datasets* de imagens, tais como planos de fundo e objetos próximos, são mantidas durante os estágios de treinamento e de teste, esta exploração de contexto pode ser traduzida em um maior nível de generalização por parte dos modelos de classificação de imagens. Contudo, quando existe um descompasso entre essas informações, esta condição pode acarretar em um elevado nível de degradação

de desempenho dos modelos computacionais [33]. Com o intuito de propor soluções correlatas a este tipo de problema, Guo et al. [214] propuseram um mecanismo de atenção (em inglês, *attention mechanism*), que objetiva estabelecer o foco dos modelos a condições indesejadas, a exemplo de fronteiras irregulares próximas da região do objeto avaliado. Esta técnica, contudo, não apresenta imunidade a alterações de planos de fundo presentes fora da região de interesse.

Por conseguinte, a fim de propor soluções para as demandas de investigação elencadas nesta Seção, objetiva-se explorar arranjos generativos adversários para o desenvolvimento de dois tipos de aplicações, sendo elas:

1. **Geração de *datasets* de imagens artificiais** com um nível adequado de realismo e diversificação, mas que, ainda assim, apresentem distribuições de probabilidades que se aproximem do objeto real de estudo.
2. **Desenvolvimento de uma técnica de *background invariance*** - direcionada à redução de sensibilidade dos modelos computacionais a interferências de planos de fundo (macro irregularidades) [35].

Para o primeiro caso listado, a abordagem desejada consiste no desenvolvimento de uma metodologia direcionada à geração artificial de imagens de componentes isoladores de distribuição, íntegros ou defeituosos, a partir de *datasets* que apresentam diferentes padrões visuais, tais como *design*, formato, cor ou textura. Por conseguinte, objetiva-se propor uma técnica que seja capaz de gerar um novo conjunto de imagens com um nível adequado de diversidade, e que ainda agregue diferentes características do *dataset* original de treinamento para produzir novos padrões visuais de inconformidades. Outro ponto de interesse está na elaboração de estratégias para que esta metodologia seja implementada através de um *dataset* reduzido de treinamento, sem que ocorra a condição de *mode collapse*.

No caso da segunda aplicação mencionada, objetiva-se desenvolver um arranjo de modelos adversários direcionados à geração de planos de fundo artificiais com o propósito de elevar o nível de imunidade de modelos classificadores de imagens a fontes de interferências externas. Este tipo de aplicação poderia contribuir no treinamento de modelos inteligentes para condições menos usuais que podem existir no domínio real do problema, a exemplo de RPDA situadas em cenários urbanos

ou rurais com diferenças significativas de macro irregularidades de planos de fundo relativas ao conjunto de imagens de treinamento.

Desta forma, as Subseções 4.4.1 e 4.4.2 apresentarão, respectivamente, aplicações com redes generativas adversárias direcionadas à geração de *datasets* artificiais de imagens de isoladores de distribuição (íntegros e defeituosos), e uma metodologia de *background invariance*, complementar ao Gerador de Imagens Artificiais (GIA) - apresentado na Seção 4.2.

4.4.1 Geração de *dataset* Artificial com cGANs

Considerando as potenciais estratégias mencionadas para o desenvolvimento de modelos geradores de *datasets* de componentes de RPDA, apresentar-se-á uma metodologia direcionada à geração de imagens artificiais de isoladores de distribuição através do emprego de redes generativas adversárias condicionais (cGANs), cujo princípio de funcionamento foi detalhado na Seção 3.11. O objetivo principal desta etapa da pesquisa reside na investigação de uma técnica que seja capaz de produzir *datasets* de imagens de componentes íntegros e defeituosos, e que ao mesmo tempo possibilite misturar os padrões visuais característicos da base de dados original.

A Figura 4.26 ilustra os arranjos de arquiteturas propostas para as cGANs e o quadro de legendas dos *blocos* de processamento da informação. De acordo com os elementos apresentados na figura, estas cGANs são compostas por um *modelo gerador* (G) e um *modelo discriminador* (D), com configurações de entrada e saída idênticas ao que foi apresentado na Figura 3.38, que ilustra um arranjo específico de cGANs. É possível observar que, em ambas as imagens (contidas nas Figuras 3.38 e 4.26), estão presentes as formas de interligação de múltiplas entradas (*multi-input*) no gerador e de múltiplas saídas (*multi-task learning*) no discriminador. A cadência de atividades envolvidas para o processo de aprendizagem deste arranjo de redes convolutivas pode ser listada da seguinte forma:

1. Como etapa inicial, define-se o *dataset* real de treinamento, que consiste em exemplos de imagens de uma das classes de componente isolador, sem defeitos (DATA-INT) e defeituosas (DATA-DEF), provenientes do dataset adquirido na ECI.

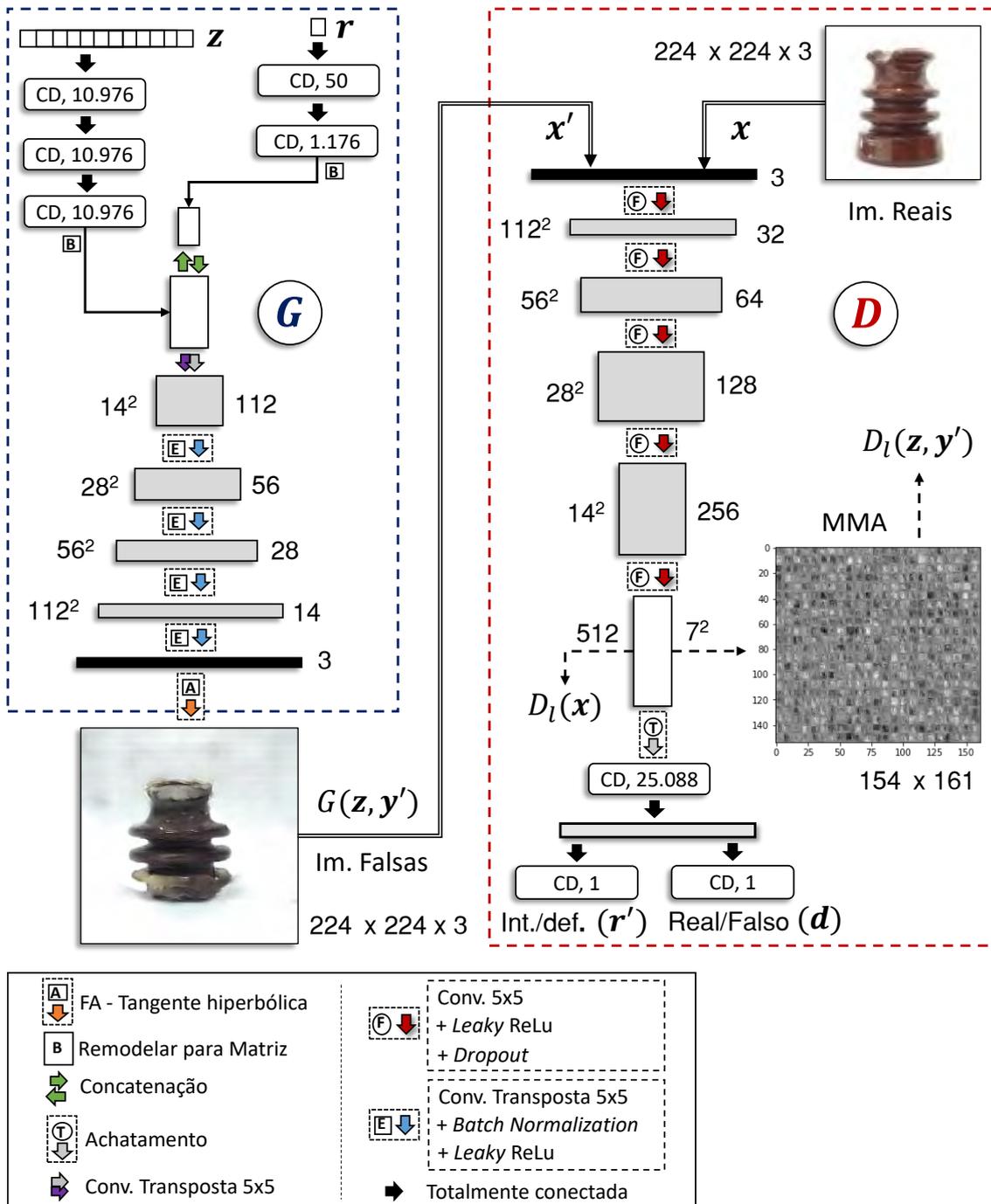


Figura 4.26: Arranjo das arquiteturas das redes convolutivas adversárias condicionais (cGANs) propostas e quadro de legendas referente aos blocos de processamento da informação.

2. Para o modelo gerador (G), são definidas as entradas do vetor aleatório, z , e da variável condicional, r , que é empregada para estabelecer o *status* de conformidade do dispositivo, em que $r = 0$ representa os componentes íntegros e $r = 1$ a condição de inconformidade.

3. Para o modelo discriminador (D), a saída d é empregada para classificar a imagem como real ou falsa e o termo de rótulo, r' , representa os *status* de conformidade dos componentes, estabelecidos tanto para os exemplos reais quanto para os falsos.
4. Configura-se a rotina de aprendizagem das cGANs. A estrutura de treinamento que foi adotada está descrita, de forma detalhada, no Algoritmo 01, presente na Subseção 3.11.1. Nesta rotina de treinamento, são definidos os seguintes hiper-parâmetros: **(a)** tamanho do mini-lote de imagens \mathbf{X} (32 imagens); **(b)** quantidade de vetores aleatórios \mathbf{Z}_A e \mathbf{Z}_B (32 para cada mini-lote); **(c)** número de épocas de treinamento ($N = 30.000$) e **(d)** função de otimização das cGANs (Adam).
5. Como etapa adicional ao algoritmo de aprendizagem, é definida uma equação para o fator de diversidade, $\mathcal{L}_z(\cdot)$. Para esta aplicação, foi realizada uma adaptação na técnica de regularização proposta por Yang et al.[86], obtendo-se o seguinte termo para $\mathcal{L}_z(\cdot)$:

$$\max_G \mathcal{L}_z(G) = E_{z_a, z_b} \left[\min \left(\frac{\|D_l[G(\mathbf{Z}_A, \mathbf{r})] - D_l[G(\mathbf{Z}_B, \mathbf{r})]\|}{\|\mathbf{Z}_A - \mathbf{Z}_B\|}, \tau \right) \right], \quad (4.15)$$

onde o parâmetro D_l corresponde ao conjunto de Camadas de Ativação (CA) correspondente ao último estágio convolutivo do modelo discriminador, mostrado na Figura 4.26. Deste modo, o referido termo de regularização realiza uma comparação entre as respostas produzidas pelos dois mini-lotes de vetores aleatórios (\mathbf{Z}_A e \mathbf{Z}_B). Para o cálculo da *taxa de controle de regularização*, $\lambda_f(\cdot)$, foi implementada uma sequência de atividades:

- (a) concatenação, $\uparrow\uparrow$, dos $N_l = 512$ mapas de ativação presentes em D_l , criando uma única imagem, com dimensões de 154×161 *pixels*, intitulada de *Mosaico de Mapas de Ativação* (MMA), obtida através da equação:

$$D_{\text{MMA}}(\mathbf{z}, \mathbf{r}) = \uparrow\uparrow_{j=1}^{N_l} D_{l_j}(\mathbf{z}, \mathbf{r}), \quad (4.16)$$

Um exemplo de mosaico de mapas de ativação ilustrado na Figura 4.26.

- (b) implementa-se os *descritores de Haralick* [113], $h_R(\cdot)$, para extração de características de textura para um conjunto de MMA provenientes do

mini-lote, através da equação:

$$h_R(D_{\text{MMA}}(\mathbf{z}, \mathbf{r})) = \sqrt{\sum_{i=1}^{N_h} |f_i|^2}, \quad (4.17)$$

onde N_h corresponde ao número de características de Haralick, e f_i representa cada uma delas.

(c) Realiza-se o cálculo de $\lambda_f(\cdot)$ através da fórmula:

$$\lambda_f(h_R(\mathbf{z}, \mathbf{r})) = \left(\frac{h_R(D_{\text{MMA}}(\mathbf{Z}_A, \mathbf{r})) - h_R(D_{\text{MMA}}(\mathbf{Z}_B, \mathbf{r}))}{h_R(D_{\text{MMA}}(\mathbf{Z}_A, \mathbf{r})) + h_R(D_{\text{MMA}}(\mathbf{Z}_B, \mathbf{r}))} \right)^2 \quad (4.18)$$

6. implementação da rotina de aprendizagem com a inclusão do termo de regularização, expresso da forma:

$$\min_G \max_D \mathcal{L}_{cGAN}(D, G) - \lambda_f(h_R(\mathbf{z}, \mathbf{r})) \mathcal{L}_z(G) \quad (4.19)$$

7. Durante o estágio de treinamento, implementa-se uma rotina de *data augmentation* para alteração digital das imagens de treinamento (\mathbf{X}). Os valores estabelecidos para esta rotina de aumento de dados foram os mesmos apresentados na Seção 4.2.

Para a aplicação desenvolvida nesta pesquisa, a implementação do fator de diversidade, $\mathcal{L}_z(\cdot)$, e da taxa de controle de regularização, $\lambda_f(\cdot)$, teve como principais objetivos elevar o nível de imunidade do modelo gerador à condição de *mode collapse* e também aumentar a diversidade e o número de características visuais das amostras produzidas pelo modelo gerador. Ao final do processo de aprendizagem das cGANs propostas, foram avaliados os padrões de imagens sintéticas produzidas (a exemplo de formato, cor e textura), inerentes aos *datasets* de componentes isoladores de distribuição.

4.4.2 Técnica de *Background Invariance*

Neste estágio da pesquisa, foi proposta uma metodologia direcionada a ampliar o nível de tolerância de modelos computacionais a macro interferências inerentes a planos de fundo não controlados - abordagem conhecida como *background invariance* [35]. Esta aplicação consiste no desenvolvimento de uma técnica de geração de

planos de fundo artificiais, produzidos adversariamente, que fornecem representações significativas de contexto durante o processo de treinamento de modelos inteligentes de classificação de imagens. Contudo, para impedir que os planos de fundo adversários não comprometam a interpretação da classe de interesse, a inserção dos mesmos ficou restrita apenas aos *pixels* que não fazem parte do objeto. Como resultado desta metodologia, obteve-se modelos menos propensos a apresentar níveis inadequados de sensibilidades às informações de plano de fundo contidas nas imagens de teste. A Figura 4.27 ilustra o diagrama de fluxo da informação adotado para o desenvolvimento desta aplicação.

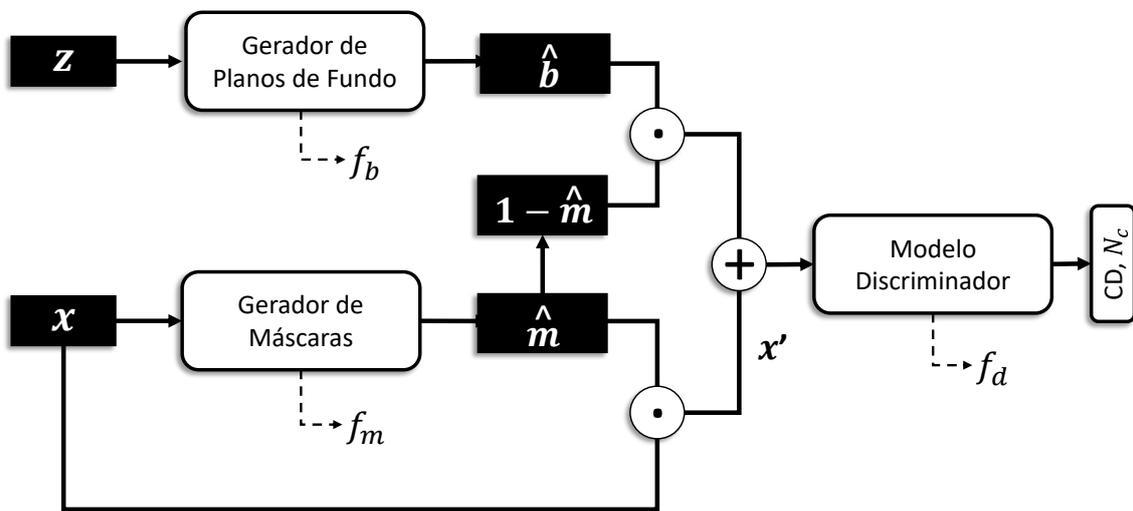


Figura 4.27: Diagrama de fluxo da informação adotado para o desenvolvimento da proposta de geração de planos de fundos adversários durante o estágio de treinamento.

Dentre os estágios de fluxo da informação apresentados na Figura 4.27, pode-se citar três blocos principais, nomeadamente: (a) rede convolutiva para classificação de imagens (modelo discriminador - f_d); (b) modelo gerador de máscaras (f_m) e (c) modelo gerador de planos de fundo adversários (f_b). Para o caso do modelo discriminador, foi desenvolvida uma rede convolutiva (criada a partir de blocos da rede VGG-19), composta por três camadas convolucionais - com 32, 64 e 128 filtros receptivos -, adicionadas a camadas de agrupamento do tipo *max-pooling*. As funções de ativação implementadas foram a *ReLU* e a função de otimização escolhida foi a Adam, com taxa de aprendizado na ordem de 10^{-3} . Este modelo é otimizado com foco na minimização da função custo $\mathcal{L}_B(f(x), y)$.

O modelo gerador de máscaras, por sua vez, é composto por uma rede convolutiva de segmentação semântica do tipo U-Net, com arquitetura e forma de aprendizagem idênticas ao que foi apresentado na Subseção 4.2.1. Considerando os dados de treinamento, utiliza-se um conjunto de imagens \mathbf{x} como tensor de entrada e o termo y como rótulo de classe. Estas imagens também estão sujeitas ao emprego de técnicas de *data augmentation* - seguindo os mesmos parâmetros apresentados na Seção 4.2. Durante a etapa de treinamento, o modelo de segmentação é utilizado para produzir máscaras (\hat{m}) usando uma função de ativação sigmoide para assegurar que $\hat{m} \in [0, 1]$. Deste modo, ele pode ser empregado para segmentar a imagem através de um produto escalar, da forma

$$x' = x \odot \hat{m} \quad (4.20)$$

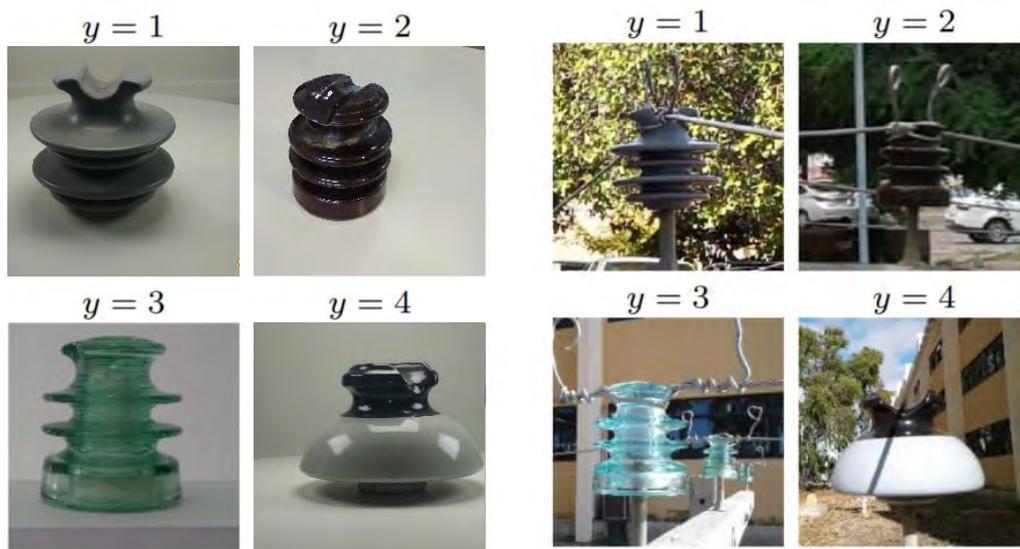
De forma simultânea à etapa de segmentação, amostras de planos de fundo artificiais são produzidas por meio de um modelo gerador, f_b . Ele se inicia com uma matriz de ruído com distribuição uniforme, \mathbf{z} , com dimensões 4×4 , e realiza estágios de convoluções transpostas até que se obtenha o tamanho da imagem de treino, adotando as mesmas quantidades de filtros receptivos citadas anteriormente. Para este caso, foi escolhida a função de ativação do tipo *LeakyReLU* com um gradiente de 0,2, com o propósito de evitar a presença de neurônios de valor nulo ocasionados pela natureza dinâmica do treinamento. O método de otimização escolhido foi o Adam, com uma taxa de aprendizagem da ordem 10^{-5} .

Em contraste com os demais modelos, o gerador de planos de fundo é treinado para maximizar a função custo, $\mathcal{L}_B(\cdot)$, tendo em vista que ele se foca na produção de imagens ou artefatos que possuem o potencial de afetar negativamente os desempenhos de classificação. No entanto, o escopo dos planos de fundo realistas deve ser limitado, pois estas imagens não podem gerar cópias dos objetos de interesse para a classificação. Isto é assegurado através da definição das seguintes restrições: cada um dos geradores de n_b^2 idênticos produz apenas um pequeno fragmento (*patch*), $n_b \times n_b$ de imagem de plano de fundo; estes fragmentos são então concatenados de forma independente, $\uparrow\uparrow$, através de:

$$\hat{b} = \uparrow\uparrow_{j=1}^{n_b^2} f_b(z_j). \quad (4.21)$$

Isto evita planos de fundos não realistas, mantendo o fluxo de sinais diferenciável.

Para a análise de desempenho da metodologia proposta, foram empregados diferentes tipos de *datasets* de imagens. Na primeira etapa experimental, fez-se uma avaliação das bases de dados apresentadas na Seção 4.1, utilizando as amostras coletadas em ambiente controlado (ECI) para treino e a base de dados realista (RDE) para testes. A Figura 4.28 mostra exemplos de diferentes conjuntos de imagens de isoladores de distribuição e revela o descompasso entre os planos fundo inerentes a estas duas abordagens de coleta de *datasets*.



(a) Imagens de treino.

(b) Imagens de teste.

Figura 4.28: *Datasets* de isoladores de distribuição empregados para as etapas de treino e de testes da metodologia proposta

Em um segundo momento, com o propósito de investigar o nível de generalização desta técnica, também foram avaliados diferentes tipos de *datasets* de imagens em escala de cinza, a exemplo das base de dados MNIST e Fashion-MNIST [215]. Em ambos os casos, estão contidas 60.000 imagens de treinamento e 10.000 amostras para teste. Estas imagens, originalmente nas dimensões 28×28 , são aumentadas para 32×32 através da implementação de uma função *padding*. Logo após esta etapa, são introduzidos artificialmente diferentes tipos de planos de fundo, para serem utilizados na rotina de teste, conforme apresentado na Figura 4.29.

Para o caso de MNIST e Fashion-MNIST, onde os objetos possuem a mesma cor,

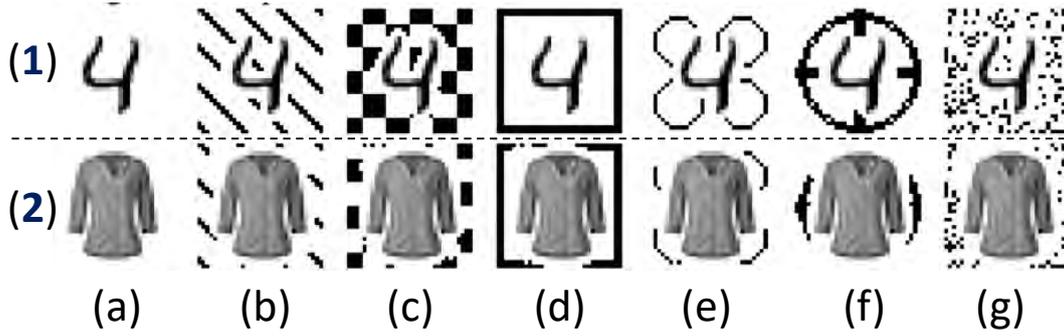


Figura 4.29: Planos de fundos introduzidos nos *datasets* (1) MNIST e (2) Fashion-MNIST: (a) original, (b) listras, (c) tabuleiro, (d) quadro, (e) círculos, (f) relógio e (g) ruído aleatório.

o gerador de planos de fundo teria a capacidade de “enganar” o modelo discriminador, produzindo contextos com a mesma cor do objeto e, conseqüentemente, ofuscando toda a imagem. Nestes casos, um termo adicional de regularização $\mathcal{L}_{BR}(\hat{\mathbf{b}})$ é adicionado para impedir os planos de fundo de preencher mais de metade dos pixels, cuja equação é dada por:

$$\mathcal{L}_{BR}(\hat{\mathbf{b}}) = \max\left(\frac{1}{N} \sum_{i=1}^N \hat{b}_i - \frac{1}{2}, 0\right). \quad (4.22)$$

A partir das informações citadas, o problema de otimização min-max pode ser resumido como:

$$\min_{f_d} \max_{f_b} \sum_{i=1}^N \mathcal{L}_B\left(f_d(x_i \odot \hat{m}_i + \hat{b}_i \odot (1 - \hat{m}_i)), y_i\right) + \mathcal{L}_{BR}(\hat{\mathbf{b}}). \quad (4.23)$$

É importante enfatizar que, embora o problema de otimização proposto tenha sido inspirado pelas redes generativas adversárias, não se trata de uma estrutura de GANs, pois não existe um modelo discriminador de imagens verdadeiras ou falsas. Neste caso, é implementada uma rede convolutiva de classificação de imagens, f_d . Outro fator importante é que os *datasets* de treino e de teste são compostos por imagens com distribuição de probabilidades distintas em relação às informações de planos de fundo.

4.5 Critérios de Avaliação de Desempenho

Com a finalidade de realizar avaliações quantitativas dos resultados experimentais obtidos nesta pesquisa, mais especificamente aqueles relacionados às tarefas de classificação de imagens e geração de *datasets* artificiais, foram implementados diversos critérios de avaliação de desempenho. As Subseções seguintes trataram destas técnicas através de duas temáticas distintas, sendo elas: **(a)** medidas de desempenho de tarefas de classificação e **(b)** avaliação estatística de *datasets* artificiais.

4.5.1 Desempenho de Classificação

Para as tarefas de detecção multi-classes, o principal parâmetro avaliado foi o de Acurácia (**Acc**), ou seja, o percentual de vezes que os modelos computacionais apresentaram a saída correta, de acordo com a resposta real $y_{i,k}$. A saída do algoritmo de aprendizagem fornece o valor previsto $\hat{p}_{i,k}$ para cada observação i da classe k . Estes valores são discretizados em uma matriz no formato *one-hot encoding* [216], definindo $\hat{p}_{i,k}$ como 1 para toda classe k prevista pelo modelo e 0 caso contrário. Para N amostras e K classes, a acurácia pode ser definida como:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{i,k} \hat{p}_{i,k}. \quad (4.24)$$

No que se refere às tarefas de detecção de defeitos, considerou-se a desproporção numérica entre as imagens de componentes íntegros e defeituosos (desbalanço entre classes) na avaliação do desempenho dos modelos computacionais propostos. O parâmetro **F1-Score** foi então utilizado como uma das abordagens direcionadas para os casos de detecção binária desbalanceada [217, 218]. São considerados para a classe k (presente em $y_{i,k}$) os valores $k = 0$ e $k = 1$ como as classes negativa e positiva, respectivamente, resultando nos seguintes parâmetros:

1. Verdadeiros Positivos (VP):

$$\text{VP} = \sum_{i=1}^N y_{i,1} \hat{p}_{i,1} \quad (4.25)$$

2. Falsos Positivos (FP):

$$\text{FP} = \sum_{i=1}^N y_{i,0} \hat{p}_{i,1} \quad (4.26)$$

3. Falsos Negativos (FN):

$$FN = \sum_{i=1}^N y_{i,1} \hat{p}_{i,0} \quad (4.27)$$

O parâmetro VP se refere ao número de amostras positivas (classes reais) que são classificadas corretamente e FP determina a quantidade de exemplos erroneamente classificados como positivos. O parâmetro FN, por sua vez, condiz com o número de casos positivos falsamente classificados como negativos. De posse destas informações, é possível se obter a equação que determina o valor de F1, dada por:

$$F1 = \frac{2VP}{2VP + FN + FP}. \quad (4.28)$$

Com a finalidade de obter informações mais apuradas sobre possíveis erros de classificação, foram geradas *Matrizes de Confusão* para parte dos algoritmos trabalhados. As colunas da MC representam as instâncias de uma classe prevista, enquanto as linhas demonstram os casos das classes reais [43]. Dessa forma, a MC gera uma Matriz $M_{j,k}$ que pode ser visualizada como:

$$M_{j,k} = \sum_{i=1}^N y_{i,j} \hat{p}_{i,k} \quad (4.29)$$

Um exemplo de matriz de confusão, com apresentação dos percentuais de classificação, pode ser visualizado na Figura 4.30 [16]. No caso perfeito (sem erros de classificação), a MC possui zero em todos os elementos, exceto na diagonal. Esta ferramenta permite identificar quais classes do modelo estão enfrentando maiores dificuldades de serem preditas.

As principais equações usadas para se trabalhar com uma MC foram apresentadas por Fawcett [44] e são listadas, para cada classe k , da forma:

- Taxa de Verdadeiro Positivo (*TVP*):

$$TVP_k \approx \frac{VP_k}{(Total\ de\ Positivos)_k} \quad (4.30)$$

- Taxa de Falso Positivo (*TFP*):

$$TFP_k \approx \frac{FP_k}{(Total\ de\ Negativos)_k} \quad (4.31)$$

- Precisão (Pr):

$$Pr_k = \frac{VP_k}{(Total\ de\ Positivos\ Preditos)_k} \quad (4.32)$$

- Sensibilidade (S):

$$S_k = TVP_k \quad (4.33)$$

- Especificidade (Es):

$$Es_k = 1 - TFP_k \quad (4.34)$$



Figura 4.30: Exemplo de matriz de confusão para três classes distintas.

Uma outra forma de analisar o desempenho de classificadores para diferentes configurações é dada por meio das *curvas ROC* [219]. A curva ROC é um gráfico bidimensional em que TVP é plotada no eixo vertical e TFP é plotada no eixo horizontal. A curva ROC descreve a relação relativa entre os benefícios (verdadeiros positivos) e os custos (falsos positivos). A Figura 4.31 ilustra os comportamentos de três curvas ROC sobrepostas em um problema de classificação [16]. Os desempenhos são melhores quando as áreas abaixo da curva (em inglês, *Area Under the Curve - AUC*) da ROC estão próximas do valor 1 ou a referida curva é achatada na parte de cima do gráfico, correspondendo a 100% de sensibilidade e 100% de especificidade.

Com o objetivo de testar a capacidade de generalização dos modelos computacionais propostos para diferentes conjuntos de dados (pertencentes ao mesmo domínio estudado) e/ou assinalar eventuais problemas de desempenho, a exemplo de enviesamento ou sobreajuste, fez-se uso da técnica de *Validação Cruzada* (VC). A validação

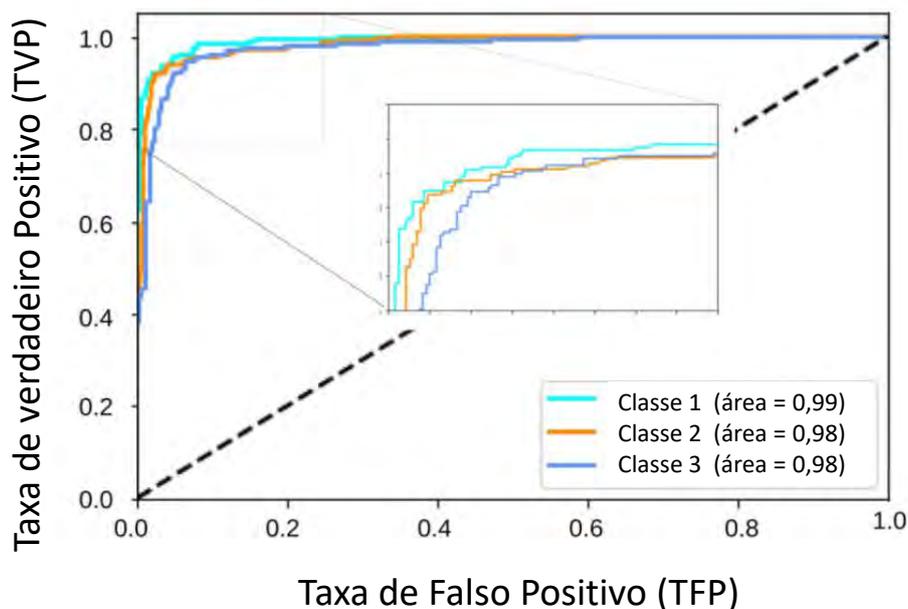


Figura 4.31: Exemplo de curvas ROC sobrepostas para três classes distintas.

cruzada (em inglês, *cross validation*) pode ser compreendida como um procedimento de *reamostragem* empregado para avaliar modelos de aprendizagem de máquina que se utilizam de base de dados com quantidades limitadas – os *datasets* coletados na ECI e na RDD, por exemplo, estão inseridos nesta categoria. Este procedimento possui um único hiper-parâmetro, denominado de k , que se refere ao número de grupos (em inglês, *folds*) em que uma determinada base de dados deve ser dividida [220].

4.5.2 Desempenho Estatístico de Geradores de Imagens

Outro ponto também avaliado nesta pesquisa foi o nível de qualidade e a diversidade de *datasets* produzidos por metodologias de geração de imagens, a exemplo das redes generativas adversárias. Com este propósito, foram utilizadas duas técnicas usualmente empregadas para avaliar o desempenho de modelos generativos, chamadas de *Inception Score (IS)* e *Fréchet Inception distance (FID)*. No caso desta pesquisa, serão avaliadas as características estatísticas de *datasets* produzidos pelos GIA e cGANs propostas. As referidas técnicas serão apresentadas a seguir:

***Inception Score (IS)*:** Esta técnica foi proposta por Salimans et al. [171], e proporciona uma forma quantitativa de avaliar o nível de qualidade de amostras de imagens produzidas por modelos generativos. O IS se utiliza de uma rede *In-*

ception [164], pré-treinada com o *dataset* ImageNet [91], para calcular os dados logísticos das imagens geradas. A pontuação é dada por:

$$\text{IS}(G) = \exp \left(E_{\mathbf{x} \sim p_g} [D_{\text{KL}}(p(y|\mathbf{x}) \parallel p(y))] \right) \quad (4.35)$$

onde \mathbf{x} é a amostra das imagens geradas pelo modelo gerador e p_g representa sua distribuição de probabilidades. O termo E é o valor da expectativa de um conjunto de imagens geradas e D_{KL} , por sua vez, é a *divergência-KL* entre a distribuição condicional de classes $p(y|\mathbf{x})$ – do rótulo y da rede Inception –, e a distribuição marginal de classes, dada por $E_{\mathbf{x} \sim p_g} [p(y|\mathbf{x})]$ [221]. O desenvolvimento do IS foi motivado por dois fatores principais [222]: **(1)** a distribuição condicional de amostras rotuladas contendo objetos significativos deve apresentar baixa entropia; **(2)** caso o nível de diversidade das amostras seja elevado, a distribuição marginal deve apresentar alta entropia.

A pontuação obtida através dessa técnica limita-se a medir apenas o nível de diversidade do *dataset* de imagens geradas. Barratt e Sharma [223] apresentaram outros tipos de limitações, a exemplo da grande variação de pontuações e a alta sensibilidade da técnica a pequenas alterações nos pesos da rede Inception. O IS é considerado atualmente como uma técnica satisfatória caso seja utilizada de forma complementar ao procedimento que será apresentado a seguir.

Fréchet Inception Distance (FID): Esta técnica foi proposta por Heusel et al. [224] e consiste na comparação entre duas funções de probabilidades distintas produzidas por uma das camadas intermediárias da rede convolutiva do tipo *Inception-v3* – mais especificamente os 2.048 parâmetros de ativação da camada pool3 [164]. No caso desta técnica, em contraste com o IS, são avaliadas tanto as imagens reais (\mathbf{x}) quanto as amostras artificiais (\mathbf{g}), criadas pelo modelo gerador, resultando nas distribuições de probabilidade p_x e p_g . O parâmetro FID, por sua vez, é obtido através do cálculo da *distância de Fréchet* [225], $d^2(.,.)$, entre duas gaussianas com média e covariância dadas por μ e C , respectivamente. A equação característica da FID é dada por:

$$\text{FID}(\mathbf{x}, \mathbf{g}) = d^2((\mu_x, C_x), (\mu_g, C_g)) = \|\mu_x - \mu_g\|_2^2 + \text{Tr}(C_x + C_g - 2(C_x C_g)^{\frac{1}{2}}), \quad (4.36)$$

onde $\text{Tr}(\cdot)$ representa o traço de uma matriz [225]. Os termos μ_x e μ_g representam,

respectivamente, as médias das distribuições de probabilidade das imagens reais (\mathbf{x}) e das amostras produzidas pelo modelo gerador (\mathbf{g}). Já os parâmetros C_r e C_g correspondem ao valores de covariância de p_r e p_g , respectivamente.

Valores mais baixos de FID são desejados, tendo em vista que ele corresponde a uma proximidade estatística entre os dois *datasets* avaliados, que são medidas pela distância entre suas distribuições de probabilidade de ativação da RNC *inception*. De acordo com os referidos autores desta técnica, a FID demonstra um elevado nível de sensibilidade para a condição de *mode collapse* e ainda é mais resistente a ruídos do que a técnica IS. Por conseguinte, ela se mostra como a técnica mais adequada, comparativamente à *inception score*, para fornecer medidas quantitativas do nível de diversidade de *datasets* de imagens.

4.6 Técnica de Validação Cruzada

A técnica de validação cruzada é considerada de fácil entendimento e resulta geralmente em parâmetros de desempenho menos tendenciosos em comparação com outros métodos existentes na literatura [220]. Ela assegura que cada observação do conjunto de dados original possa estar presente nos conjuntos de treino e de testes. Este método segue os seguintes passos: (1) dividir o dataset de treino aleatoriamente em k grupos, com valores normalmente compreendidos entre 5 e 10, a depender do tamanho da base dados; (2) treinar o modelo usando os $(k - 1)$ grupos e validar o mesmo usando o k -ésimo grupo restante, anotando as acurácias e erros associados; (3) repetir o processo anterior até que cada lote k sirva como *dataset* de teste e, ao final, tirar a média das pontuações registradas - estes valores serão utilizados para definir o desempenho do modelo. A Figura 4.32 ilustra a implementação da técnica de validação cruzada tradicional para $k = 10$ (dez lotes de treinamento).

Para viabilizar o emprego da técnica de validação cruzada em demandas específicas desta pesquisa, a exemplo das rotinas de treinamento integradas ao gerador de imagens artificiais (GIA), foi realizada um conjunto de adaptações na forma de atuação da VC com relação aos estágios de treino, validação e testes dos modelos computacionais propostos. A Figura 4.33 ilustra a implementação desta abordagem, intitulada, para fins de simplificação de nomenclatura, de *Validação Cruzada Adap-*

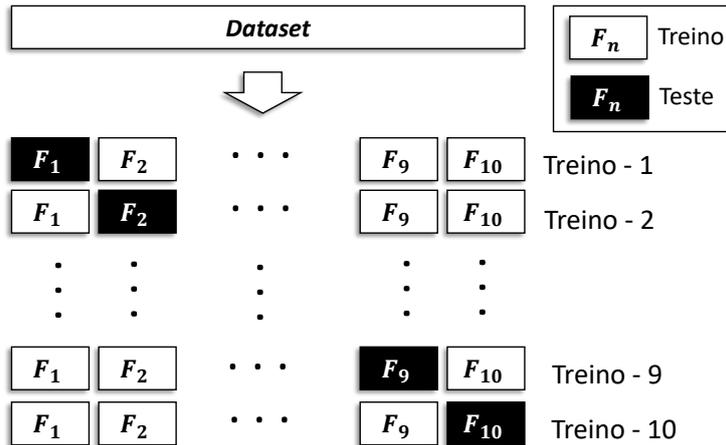


Figura 4.32: Técnica de Validação Cruzada (VC) convencional.

tada (VCA). Basicamente, esta técnica consiste no emprego de amostras produzidas pelo GIA para o estágio de treinamento do modelo e do direcionamento de lotes do *dataset* realista (RDE) para as rotinas de validação e testes. Neste caso particular de validação cruzada, não é possível utilizar os mesmos *datasets* para as etapas de treinamento, validação e teste do modelo computacional, como ocorre com a técnica tradicional. Para a VCA, as imagens produzidas pelo GIA são usadas unicamente para treinamento, enquanto o *dataset* real (coletado na RDE) é dividido em lotes para realização dos estágios de validação e teste.

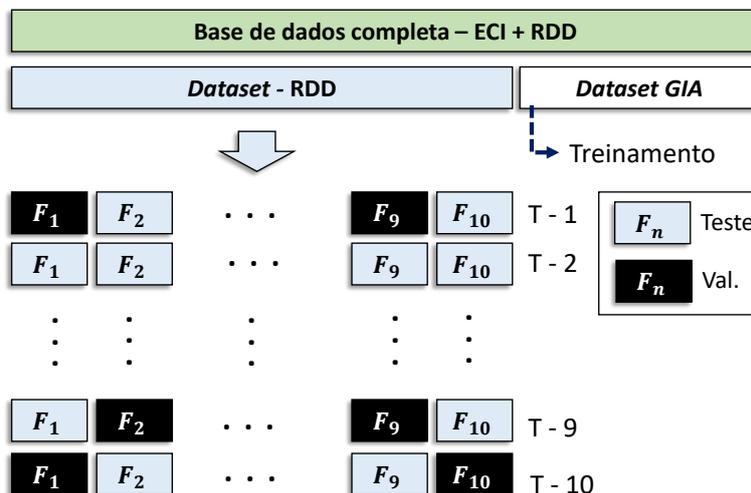


Figura 4.33: Técnica de Validação Cruzada Adaptada (VCA).

De forma mais específica, a base de dados de teste da VCA é dividida em 10 lotes e dois deles são sorteados aleatoriamente para compor o *dataset* de validação, que, conforme será apresentado no Subseção 5.4, também possui atribuições importantes

para aprendizagem do modelo, pois definirá qual época de treinamento será escolhida para gravação dos parâmetros ajustáveis do modelo computacional. Ao longo das 10 rotinas de treino realizadas, são utilizados sempre dois grupos aleatórios que podem se repetir somente um única vez. Deste modo, todos os grupos participam duas vezes do processo de validação e $(k - 2)$ vezes para as etapas de teste, haja vista que os 80% dos grupos são utilizados para avaliar o desempenho de classificação do modelo. Ao final deste processo, será realizada a média dos 10 valores de Acc registradas nas simulações.

Capítulo 5

Resultados e Discussões

Este capítulo apresenta os principais resultados experimentais obtidos através das metodologias propostas nesta pesquisa de Doutorado. A base de dados utilizada em parte dos projetos de sistemas de reconhecimento padrões, descrita no Seção 4.1, é constituída por componentes isoladores (íntegros e defeituosos) específicos de **Redes Primárias de Distribuição Aérea de energia elétrica (RPDA)**. Para os casos avaliados, são apresentados problemas distintos de classificação binária, multi-classes e *multi-task learning*, além da avaliação visual e estatística de *datasets* de imagens gerados artificialmente. As referidas metodologias são, então, comparadas entre si e com outras técnicas apresentas na literatura especializada.

5.1 Gerador de Imagens Artificiais (GIA)

Como uma das etapas de análise dos resultados experimentais desta pesquisa, foram avaliados os estágios de processamento da informação inerentes às rotinas que compõem o *Gerador de Imagens Artificiais (GIA)*, apresentado na Seção 4.2. Além das simulações computacionais que envolvem o GIA, outras estratégias foram adotadas para as etapas de treinamento e teste das rotinas de aprendizagem pertencentes aos modelos de classificação de imagens. O algoritmo de aprendizagem profunda escolhido para avaliação de desempenho das etapas de processamento da informação do GIA foi a rede convolutiva VGG-19, pré-treinada com o *dataset* de imagens ImageNet [91].

Com o objetivo de avaliar o impacto de cada uma das atividades descritas na

Seção 4.2, foram desenvolvidas rotinas computacionais que atuam de forma independente, sendo referidas como *estágios de treinamento*. Dessa forma, além do treinamento direto das imagens brutas da ECI, outras abordagens foram progressivamente sendo introduzidas. A Figura 5.1 apresenta, de forma mais detalhada, um diagrama de funcionamento com os principais estágios de fluxo da informação. Estas etapas serão apresentadas a seguir:

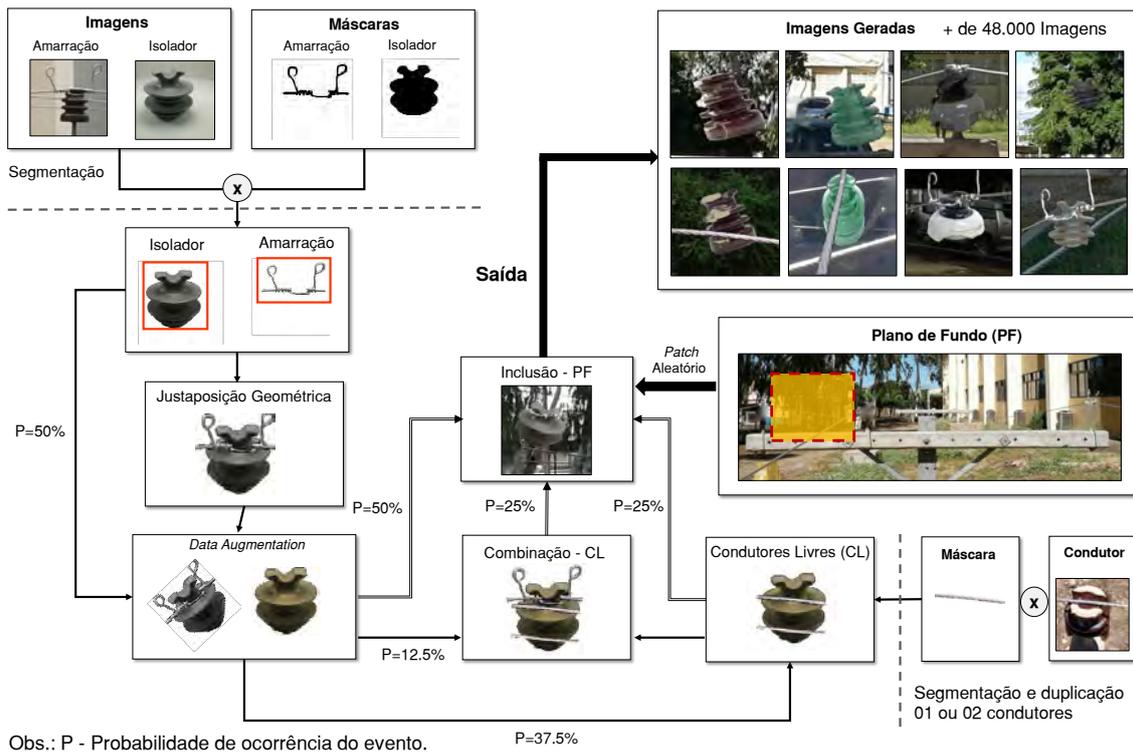


Figura 5.1: Diagrama processamento da informação do Gerador de Imagens Artificiais (GIA).

- A: A rotina de *data augmentation* é ativada conforme detalhamento presente no item 2, Seção 4.2.
- B: Ajuste fino das duas últimas camadas convolucionais da RNC. No caso da VGG-19, as camadas escolhidas foram $C_{(n-1)}$ e $C_{(n)}$ (Ilustradas na Figura 4.24).
- C: Segmentação de imagens do *dataset* de treinamento a partir das máscaras exemplificadas na Figura 4.12, descritas no item 3 da Seção 4.2. As imagens resultantes consistiram na sobreposição do isolador com um fundo preto. A Figura 5.2 ilustra alguns exemplos de imagens com os estágios A e C ativados.



Figura 5.2: Imagens geradas com ativação dos estágios C e A.

D: Adição de planos de fundo com recortes aleatórios de fragmentos (*patches*) de imagens retiradas de fotografias de ambientes externos (ver Figura 5.3), de acordo com a metodologia descrita no item 4 da Seção 4.2.



Figura 5.3: Imagens geradas com ativação dos estágios A, C e D.

E: Adição de amarrações metálicas e condutores livres nas imagens de treino de acordo com o item 3 da Seção 4.2. A Figura 5.4 ilustra imagens resultantes da ativação dos estágios A, C, D e E.



Figura 5.4: Imagens geradas com ativação dos estágios A, C, D e E.

F: Adoção da modalidade de treino *Multi-Task Learning*, conforme a configuração do Tipo 3 para classificação simultânea do modelo de isolador e respectivo *status* de conformidade, exemplificada na Seção 4.3.3, Figura 4.24.

Testes Convencionais: Este teste foi implementado utilizando o dataset coletado em ambiente externo, proveniente da rede de distribuição didática.

Stress Test: Para simular cenários mais realistas e o emprego de robôs móveis para aquisição de imagens, os *datasets* de teste também sofreram *data augmentation* para torná-los mais desafiadores. Este procedimento envolveu as seguintes transformações combinadas: rotação, variação de brilho, cisalhamento, deslocamento, ampliação e redução do zoom. Essas mudanças foram semelhantes às realizadas na fase

de treinamento, contudo mais intensamente aplicadas. A Figura 5.5 ilustra exemplos das imagens do dataset de testes após a realização do estágio de *stress test*.



Figura 5.5: Exemplos de imagens do *dataset* de teste após o estágio de *Stress test*

Ensemble Test Esse procedimento é constituído essencialmente de duas etapas: processo de classificação e fusão em nível de pontuação [208]. Cada imagem de teste passou por seis tarefas isoladas de *data augmentation* (as mesmas citadas para a fase de *stress test*). Ao final, foi obtida uma previsão para cada imagem transformada e a decisão do conjunto baseou-se no sistema de votação por maioria simples.

5.1.1 Avaliação Estatística dos *datasets* de Imagens

Para a medição quantitativa do nível de diversidade e correlação estatística dos conjuntos de imagens trabalhados neste estágio da pesquisa, foram avaliados os parâmetros *Inception Score* (IS) e o *Fréchet Inception Distance* (FID). Para o caso do *Inception Score*, objetiva-se identificar os níveis de diversidade apresentados pelos *datasets*. Isto implica que, quanto maior for o valor deste parâmetro, mais acentuadas serão as diferenças entre as imagens que compõem o mesmo *dataset*. No caso do FID, avalia-se o nível de correlação entre dois conjuntos de imagens. Mais precisamente, obtém-se um valor unidimensional que quantifica a proximidade entre duas distribuições de probabilidade distintas. Valores baixos de FID indicam que os dois grupos de imagens possuem forte similaridade estatística. A Tabela 5.1 apresenta os valores de IS para cada um dos referidos *datasets*, bem como o FID desses conjuntos de imagens em relação ao *dataset* de referência coletado na RDE.

A partir das principais características citadas para os parâmetros de desempenho IS e FID e das informações presentes na Tabela 5.1, é possível inferir que o GIA proporcionou uma elevação considerável no nível de diversidade do conjunto de imagens provenientes da ECI, obtendo o IS médio de 6,0 - 91% superior ao calculado para o *dataset* da ECI, na sua configuração original. Considerando o FID,

Tabela 5.1: Valores de IS e FID para os *datasets* apresentados.

Tipo de Imagem	Dataset Original		Data Augmentation		GIA	
	IS	FID	IS	FID	IS	FID
ECI	3,1 ± 0,1	228,7	4,8 ± 0,3	233,6	6,0 ± 0,3	96,9
RDE	5,0 ± 0,2	0	5,4 ± 0,2	83,7	—	—

o GIA também apresentou alterações significativas de desempenho - o valor de 96,9 é 57% inferior ao apresentado pelo dataset ECI, e próximo do valor obtido para o conjunto de imagens da RDE após ocorrer a rotina de *data augmentation* (83,7). A partir desses resultados, pode-se inferir que as imagens artificiais produzidas pelo GIA possuem maior similaridade estatística com o dataset de referência (imagens do ambiente realista - RDE) que o conjunto de imagens coletados na ECI.

5.1.2 Segmentação Semântica de Imagens

Nesta etapa da pesquisa, foram avaliados os desempenhos de segmentação para as duas técnicas apresentadas no Capítulo 4, sendo elas: a limiarização múltipla de Otsu e a rede convolutiva do tipo U-Net, esta última adaptada com a rotina de geração de imagens artificiais descrita na Seção 4.2. Deste modo, para a análise quantitativa da qualidade das máscaras geradas por ambas as técnicas, foi empregado o *coeficiente Dice* (DSC) como parâmetro de avaliação [226]. O DSC é obtido através do cálculo do número de *pixels* sobrepostos de duas regiões, **A** e **B**, dividido pelo número total de *pixels* em ambas as regiões pertencentes as imagens testadas, resultando na equação:

$$DSC(\mathbf{A}, \mathbf{B}) = \frac{2 \times |\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A}| + |\mathbf{B}|}. \quad (5.1)$$

Os valores calculados por este procedimento podem ser representados de forma percentual, sendo que o patamar de 100% é a condição desejada, tendo em vista que, neste caso particular, a máscara avaliada é idêntica à segmentação dada como referência. Os conjuntos de imagens utilizados como *máscaras de referência* (variável **A** da equação 5.1), foram obtidos através do algoritmo de segmentação interativa *graph cut*, também abordado na Seção 4.2. No total, foram avaliadas 480 imagens

para cada uma das técnicas de segmentação propostas - 120 para cada modelo de componente, divididos em 60 amostras de isoladores íntegros e 60 de defeituosos. A Tabela 5.2 apresenta um sumário dos valores médios de DSC obtidos para as máscaras preditas, considerando os quatro tipos de isoladores de distribuição trabalhados nesta pesquisa. Essa tabela também revela a superioridade da U-Net para as tarefas de segmentação de imagens dos isoladores de distribuição avaliados, situados em ambientes que apresentam iluminação e planos de fundo controlados.

Tabela 5.2: Coeficientes *Dice* de segmentação das técnicas propostas - valores em %.

Técnica	IPC	ICP	IVV	ICB
Otsu	82,1 ± 9,0	89,8 ± 4,2	85,0 ± 7,5	71,1 ± 1,0
U-Net	90,8 ± 1,6	98,6 ± 7,9	98,5 ± 8,1	97,9 ± 1,5

As Figuras 5.6 e 5.7, por sua vez, ilustram exemplos de imagens segmentadas obtidas através das referidas técnicas. É possível observar que o algoritmo U-Net apresentou regiões mais condizentes com os formatos reais dos componentes para o *dataset* trabalhado do que a técnica de Otsu. Estas características, por exemplo, possibilitam o cálculo de parâmetros geométricos mais condizentes com as características visuais dos componentes e justificam a implementação de um algoritmo que, comparativamente as duas técnicas trabalhadas, possui um maior custo computacional para simulação.

Para o exemplo do Isolador Cerâmico Bicolor (ICB), presente nas Figuras 5.6d e 5.7d, somente a parte superior do componente foi segmentada corretamente pela técnica de limiarização de Otsu. É possível observar que o ICB possui diferenças significativas de coloração ao longo de sua estrutura, condição que dificulta a identificação de um valor ideal de limiar (*threshold*) para separação das regiões de interesse presentes nas imagens deste componente. O valor de 71,1% (contido na Tabela 5.2), por exemplo, indica este comportamento problemático que envolve a segmentação do componente ICB pela técnica de limiarização de Otsu, tendo em vista que este resultado representa o pior desempenho de segmentação entre todos os outros componentes avaliados. Outro ponto relevante para esta pesquisa que deve ser enfatizado consiste no bom desempenho de segmentação da U-Net para compo-

mentes isoladores defeituosos. Este algoritmo de aprendizagem profunda apresentou um nível adequado de sensibilidade às informações de forma, textura e cor presentes nas inconformidades dos componentes. As Figuras 5.9 e 5.8 ilustram exemplos de segmentação de componentes defeituosos através das técnicas apresentadas.

5.2 Classificação por Atributos Geométricos

Para a etapa experimental envolvida na metodologia apresentada na Subseção 4.3.1, foi realizado um estudo comparativo entre cinco algoritmos inteligentes empregados em sistemas de classificação, nomeadamente: **(1)** rede neural artificial *Multi-Layer Perceptron* (MLP), **(2)** *Hidden Markov Model* (HMM), **(3)** *Support Vector Machine* (SVM), **(4)** *Random Forest* e **(5)** *k*-Nearest Neighbor (*k*NN). Para os referidos algoritmos de aprendizado supervisionados, foram avaliados os desempenhos de classificação de imagens de isoladores de distribuição, íntegros e defeituosos, presentes em ambientes com iluminação e planos de fundo controlados.

Considerando a base de dados empregada para as simulações computacionais, foi utilizado um *dataset* de imagens composto por três tipos de isoladores de distribuição constituídos de materiais distintos (polímero, vidro e cerâmica). Estes componentes foram especificados na Seção 4.1, e correspondem ao Isolador Polimérico Cinza (IPC), Isolador de Vidro Verde (IVV) e Isolador Cerâmico Bicolor (ICB). No total, foram avaliadas 1.500 imagens de componentes, 500 para cada classe. Essa base de dados é dividida em 400 imagens de componentes íntegros (DATA-INT) e 100 de imagens componentes defeituosos (DATA-DEF). Os tipos e características das inconformidades presentes em DATA-DEF, inerentes a cada classe de isolador, também foram previamente apresentados na Seção 4.1.

Para a etapa de extração de atributos geométricos, as máscaras geradas pelas técnicas de segmentação sofrem as operações apresentadas na Figura 4.19. Num estágio posterior, são calculados os cinco parâmetros pertencentes ao vetor \mathbf{G}_a . Estas informações, a depender da técnica de classificação empregada, podem passar por um estágio preliminar de normalização e quantização, resultando em vetores com as características presentes na Tabela 5.3. As cinco primeiras colunas são usadas para compor o vetor de entrada $\mathbf{G}_{Qa} = [L_1, L_2, L_3, L_4, L_5]$ e as três últimas são

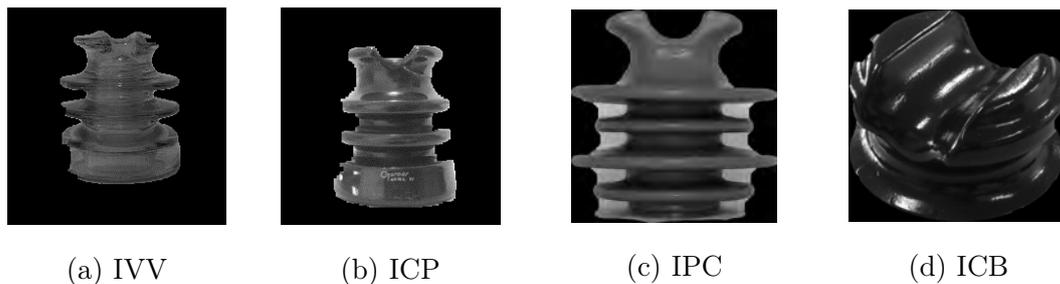


Figura 5.6: Exemplos de imagens de componentes íntegros segmentadas pela técnica do limiarização múltipla de Otsu.

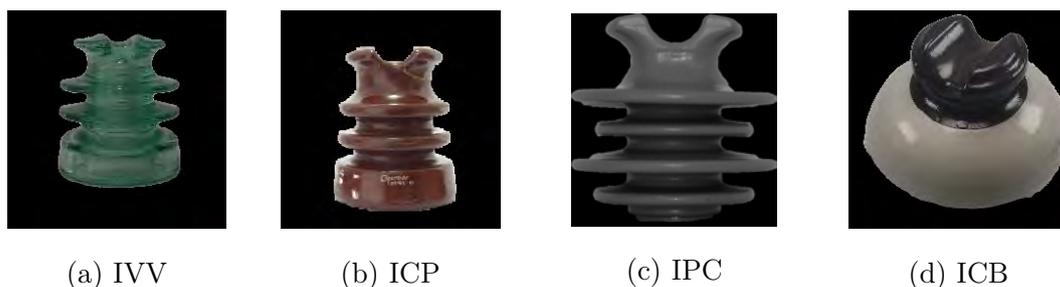


Figura 5.7: Exemplos de imagens de componentes íntegros segmentadas pelo algoritmo adaptado da Rede Neural Convolutiva (RNC) U-Net.

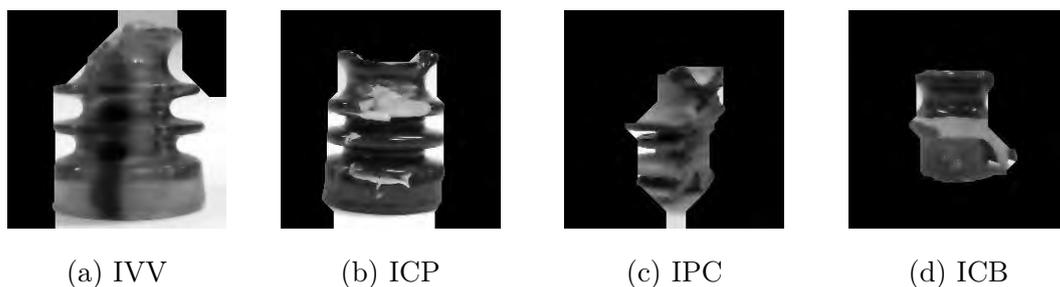


Figura 5.8: Exemplos de imagens de componentes defeituosos segmentadas pela técnica de limiarização múltipla de Otsu.

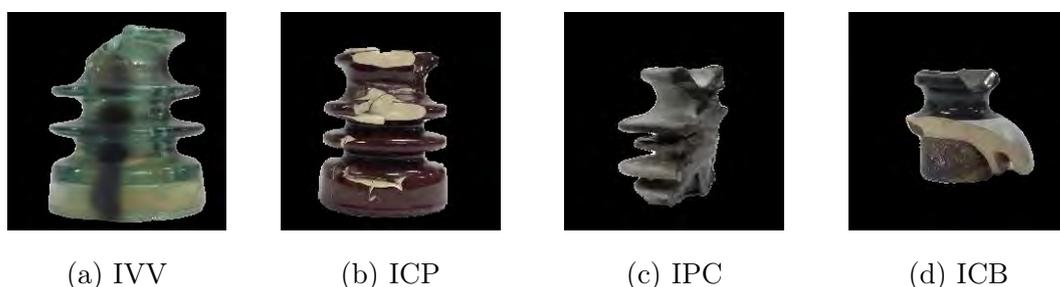


Figura 5.9: Exemplos de imagens de componentes defeituosos segmentadas pelo algoritmo adaptado da U-Net.

empregadas para criação do vetor de rótulo. Os percentuais do *dataset* de imagens, destinados aos estágios de treinamento e teste dos classificadores, foram arbitrados

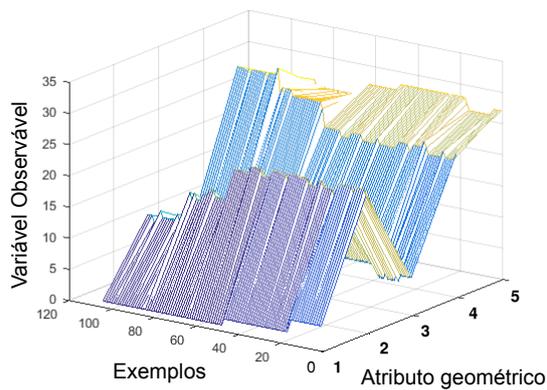
de acordo com a técnica de validação cruzada apresentada na Seção 4.6, com o número de grupos definidos em $k = 10$.

No caso do algoritmo de classificação HMM, os cinco parâmetros geométricos de cada vetor, \mathbf{G}_{Q_a} , são usados para compor uma sequência de observações $O = (O_1, O_2, \dots, O_T)$, inseridas na entrada do modelo de forma sequencial. Na Figura 5.10a, é possível visualizar um gráfico formado pela composição de 120 vetores \mathbf{G}_{Q_a} , extraídos do mesmo exemplar de um isolador de vidro verde, para fotografias coletadas de diferentes ângulos. É possível observar uma “assinatura” no comportamento sequencial dos parâmetros de \mathbf{G}_{Q_a} - número do atributo dimensional *vs* amplitude da variável observável -, que podem ser explorados pela HMM.

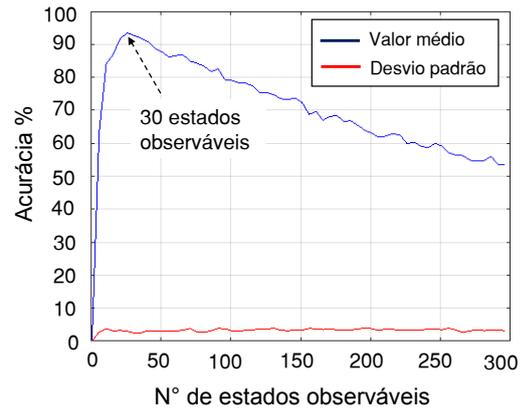
Tabela 5.3: Estrutura de dados empregada no treinamento dos modelos.

AMOSTRA	L1	L2	L3	L4	L5	IVV	ICB	IPC
1	4	22	6	23	27	1	0	0
2	5	26	4	17	23	0	1	0
3	7	17	6	26	21	0	0	1
.
.
1.500	3	24	7	18	22	1	0	0

Para a definição do número de hiper-parâmetros de cada HMM, optou-se pela escolha de 5 estados ocultos, valor que corresponde ao número de características presentes nos vetores introduzidos no sistema de classificação. Para a definição do número de variáveis observáveis (N_m), foi desenvolvida uma rotina que apresenta a acurácia de classificação para cada valor do nível de quantização (número de estados observáveis). Devido à relativa variabilidade dos resultados a cada execução, testou-se cada configuração 60 vezes e plotou-se a média dos resultados de acurácia obtidos para cada valor, conforme apresentado na Figura 5.10b. É possível observar que o desvio padrão da acurácia do sistema em cada configuração fica em torno de 3 pontos percentuais. Ainda de acordo com o gráfico apresentado, os melhores desempenhos de classificação são obtidos quando os valores de níveis de N_m ficam na região próxima a 30 estados observáveis.



(a) Atributos geométricos do vetor \mathbf{G}_{Qa} .



(b) Acurácia da HMM.

Figura 5.10: (a) Exemplos de vetores pertencentes ao *dataset* de entrada; (b) valor da acurácia média dos testes em relação ao número de estados observáveis do arranjo de HMMs.

A acurácia média de classificação para todos os modelos considerados é apresentada na Figura 5.11. O classificador HMM obteve o melhor desempenho, com 89,7% de exatidão na classificação do tipo de isolador, superando o segundo melhor classificador (*Random Forest*) em quase 5 pontos percentuais. Para os valores de precisão, mostrados na Tabela 5.4, pode-se observar que o HMM apresentou os melhores resultados de classificação para os componentes IVV e ICB, atingindo 98,7% de eficiência de classificação. Através dessas informações, é possível afirmar que foi viável transformar um vetor de parâmetros estáticos em um conjunto de valores de observação sequencial que permitiu o treinamento e teste de um conjunto de HMMs.

Tabela 5.4: Acurácias dos modelos para classificação do tipo de isolador (valores em%).

CLAS.	K-NN	RF	SVM	HMM	RNA MPL
IPC	75,4 ± 0,6	80,3 ± 0,5	97,5 ± 0,8	83,6 ± 1,5	66,1 ± 3,2
IVV	90,0 ± 0,7	93,2 ± 0,3	89,1 ± 1,2	98,6 ± 0,6	91,4 ± 1,6
ICB	82,0 ± 0,4	84,1 ± 0,7	50,4 ± 1,7	91,7 ± 1,2	74,2 ± 2,5

A Figura 5.12 apresenta a Matriz de Confusão (MC) do modelo HMM, para uma

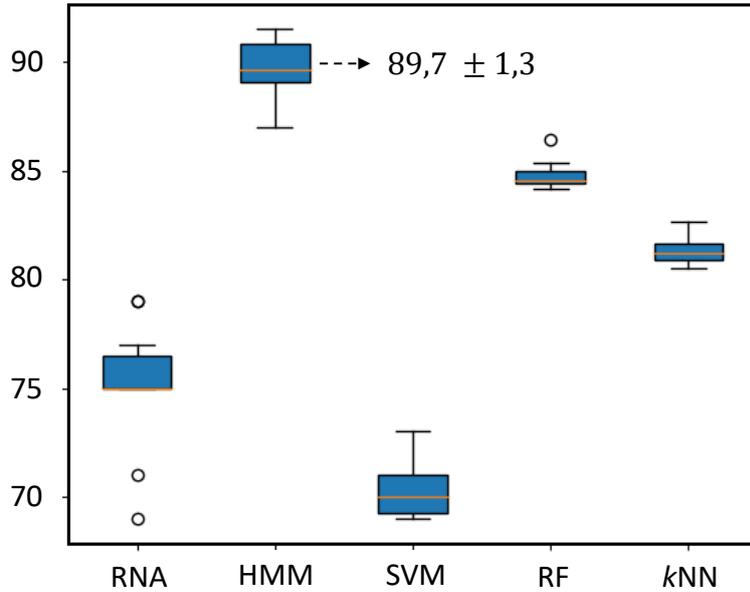


Figura 5.11: Acurácia geral de acordo com o tipo de modelo – dados do *dataset* coletados na ECI.

rotina de treinamento convencional. Pode-se observar que a precisão e a sensibilidade de todas as classes foram superiores a 80%. No entanto, esse modelo apresentou incidência de erro significativa para o componente IPC. Em estudos futuros, pretende-se comparar o desempenho entre HMMs uni-dimensionais e bidimensionais para este mesmo problema.

		Predito			
		IPC	IVV	ICB	
Real	IPC	456	6	38	91,2%
	IVV	55	441	4	88,2 %
	ICB	35	0	465	93,0%
		83,5%	98,7%	91,7%	90,8%

Figura 5.12: MC do modelo HMM para classificação do tipo de isolador.

Considerando agora o desempenho de classificação deste sistema de reconhecimento de padrões para o *dataset* de imagens coletado na Rede de Distribuição de Ensaio (RDE), foi avaliado o nível de generalização da referida técnica para base de dados mais desafiadoras. Pode-se afirmar que as imagens de componentes de RPDA, provenientes do ambiente realista (conforme descrito na Subseção 2.4.1),

normalmente apresentam alto nível de variabilidade de planos de fundo, o que pode dificultar, por exemplo, as tarefas de segmentação de imagens. A Figura 5.36 ilustra exemplos típicos de imagens de isoladores segmentadas pela técnica apresentada na Subseção 4.2.1 – referente ao algoritmo U-Net com o emprego do GIA.

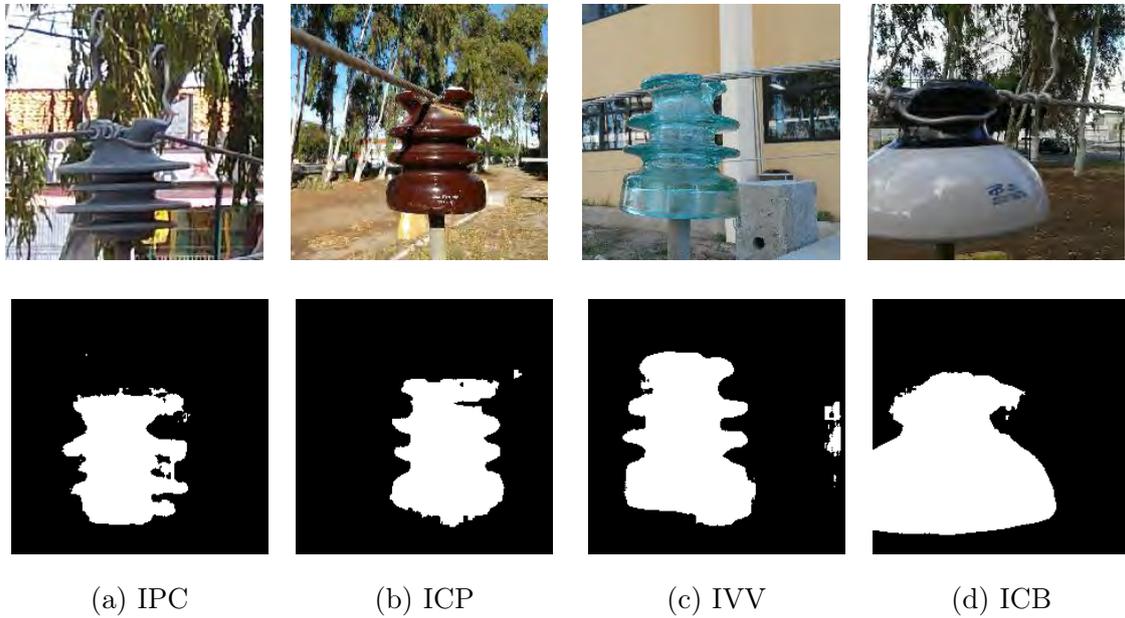


Figura 5.13: Imagens de isoladores (posicionados acima) e respectivas máscaras produzidas pelo algoritmo de segmentação U-Net treinado com o Gerador de Imagens Artificiais (GIA).

É possível observar na figura mencionada que existem deformações nas regiões presentes nas máscaras produzidas pela U-Net que podem provocar desvios indesejados nos vetores de parâmetros geométricos, \mathbf{G}_a . Deste modo, pode-se inferir que esta condição possui o potencial de impactar negativamente no desempenho de classificação dos modelos computacionais. A Figura 5.14 apresenta o gráfico *boxplot* com as acurácias médias de classificação do algoritmo HMM para o *dataset* coletado na RDE, considerando tanto os componentes íntegros quanto os defeituosos.

Para este caso, o HMM foi treinado apenas com a base de dados de isoladores íntegros, e posteriormente foi empregado para a classificação do material/*design* de componentes para ambos os *status* de conformidade – íntegros ou defeituosos. Através das informações presentes no gráfico, é possível observar que o emprego do *dataset* realista reduziu em 15 pontos percentuais a acurácia de classificação da HMM para os componentes íntegros e também resultou em baixo desempenho de

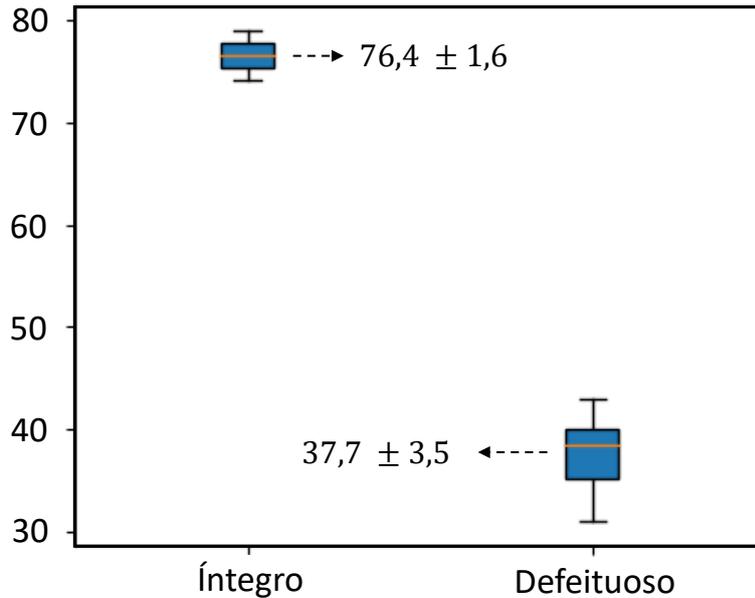


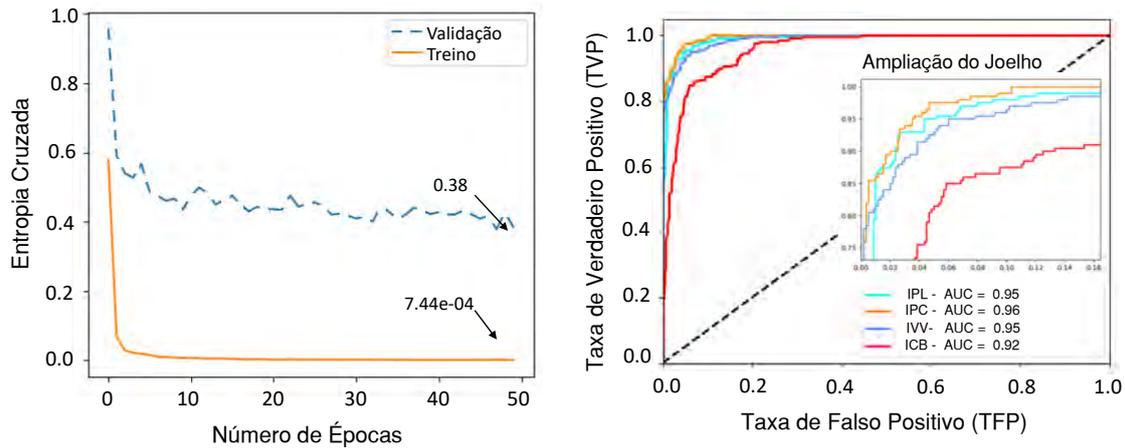
Figura 5.14: Acurácia geral da HMM para os *datasets* íntegros e defeituosos coletados na RDE.

identificação para os componentes defeituosos. Esta última condição pode também estar relacionada às possíveis alterações geométricas nas segmentações proporcionadas por determinados tipos de defeitos nos componentes, a exemplo de quebra, deformação e fratura.

5.3 Classificação por Técnicas Híbridas

Para este estágio da pesquisa, foram avaliados os desempenhos de classificação de imagens do algoritmo de aprendizagem profunda ResNet-50, juntamente com suas configurações híbridas, constituídas pela adição dos seguintes classificadores: *Support Vector Machine* (SVM), *k-Nearest Neighbors* (*k*NN), *Random Forest* (RF) e *Bayes Net* (BN). A definição da metodologia proposta e dos objetivos envolvidos nestes experimentos foram apresentados na Subseção 4.3.2. Para o caso desta RNC, os parâmetros de treinamento foram previamente ajustados para o *dataset* ImageNet, através da técnica de transferência de conhecimento, conforme descrito na Subseção 3.7.3. A camada de saída da RNC foi substituída por uma função de ativação do tipo *softmax* composta pelas quatro classes de isoladores avaliadas.

Para a rotina de treinamento da última camada achatada da rede convolutiva, o



(a) EEC para DATA-INT.

(b) Curva ROC para DATA-DEF.

Figura 5.15: Parâmetros de desempenho da ResNet-50 com camada de saída softmax.

Erro Entropia Cruzada (EEC), presente na Figura 5.15a, é uma função que representa a diferença entre os resultados obtidos no treinamento/validação e os resultados esperados. Pode-se observar na figura que o valor mínimo da curva aparece na interação 50, com EEC de treino e validação de $7,44e^{-4}$ e 0,38, respectivamente. As curvas ROC, apresentadas na Figura 5.15b, mostram o desempenho do classificador ResNet-50 para os diferentes modelos de isoladores avaliados. É possível visualizar que eles alcançaram valores próximos ao topo do gráfico e AUC acima de 0,9.

Os valores médios de acurácias de classificação obtidos nestes experimentos são apresentados na Figura 5.16. Para este procedimento, foi utilizada a técnica de validação cruzada convencional para a classificação do *dataset* de isoladores íntegros (DATA-INT) e a Validação Cruzada Adaptada (VCA) para identificação dos modelos de componentes defeituosos (DATA-DEF). Para o caso da VCA, descrita na Seção 4.6, DATA-INT foi definido para o estágio de treinamento e DATA-DEF para as etapas de validação e teste. Por conseguinte, é possível visualizar, através da Figura 5.16b, que a configuração RN + SVM apresentou o melhor desempenho de classificação, com 94,21% de Acc média para o *dataset* de componentes defeituosos. No que diz respeito à discriminação de componentes íntegros, todas as configurações trabalhadas demonstraram acurácias próximas de 100%. Para os valores de precisão (Pr), mostrados na Figura 5.5, pode-se afirmar que as configurações híbridas proporcionaram ganhos substanciais neste quesito. A precisão de ICB, por exemplo,

apresentou ganho de aproximadamente 27 pontos percentuais.

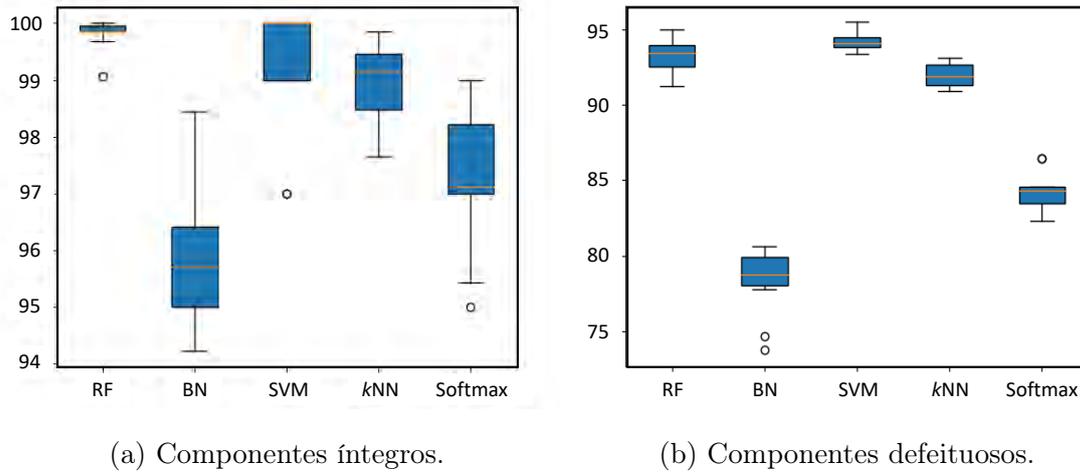


Figura 5.16: Acurácia de classificação das técnicas propostas para o *dataset* coletado na ECI - valores em %.

Tabela 5.5: Precisão das classes para DATA-DEF.

CLAS.	RN	RN + BN	RN + <i>k</i> NN	RN + SVM	RN + RF
IPC	96,4 ± 0,6	92,1 ± 1,2	83,1 ± 0,8	94,3 ± 0,6	93,8 ± 0,9
ICP	83,1 ± 1,3	83,4 ± 3,2	95,2 ± 0,8	91,1 ± 0,7	92,4 ± 1,3
IVV	95,4 ± 0,7	97,3 ± 0,8	97,3 ± 1,1	98,2 ± 0,2	98,8 ± 0,3
ICB	69,4 ± 3,3	72,4 ± 2,0	96,6 ± 0,6	96,3 ± 0,3	89,2 ± 0,8

A Figura 5.17 apresenta as matrizes de confusão para as configurações RN e RN+SVM. Analisando ambas as MCs, visualizam-se ganhos em termos de precisão e sensibilidade com a implementação da configuração híbrida. Em contraste com a condição apresentada na Figura 5.17a, todos os valores de *Pr* e *S* presentes na Figura 5.17b, foram superiores a 90%. No entanto, foi observado que 7,5% dos componentes pertencentes à classe ICB foram incorretamente classificados para a classe ICP. Acredita-se que defeitos muito agressivos em ICB (ver Figura 5.18) podem gerar erros de classificação entre estes dois componentes.

Ao final desta etapa, foi avaliado o desempenho de classificação dos referidos arranjos híbridos para o *dataset* de imagens coletado na RDE. O objetivo deste experimento foi verificar o nível de influência das interferências de plano de fundo nos con-

		Predito							Predito				
		IPC	ICP	IVV	ICB				IPC	ICP	IVV	ICB	
Real	IPC	143	6	7	44	71,5%	Real	IPC	185	5	0	10	92,5%
	ICP	2	195	0	3	97,5 %		ICP	5	195	0	0	97,5 %
	IVV	0	12	161	27	80,5%		IVV	0	0	200	0	100%
	ICB	3	20	0	177	88,5%		ICB	5	15	0	180	90,0%
		96,7%	83,7%	95,8%	70,5%	84,5%			94,9%	90,7%	100%	94,7%	95,0%

(a) ResNet-50 + softmax. (b) ResNet-50 + SVM.

Figura 5.17: MCs da ResNet-50 para DATA-DEF.



Figura 5.18: Exemplos de erros de classificação para VGG-19 + SVM.

juntos de modelos inteligentes empregados para a identificação do material/*design* de componentes isoladores. A Figura 5.19 apresenta dois gráficos *boxplots* com as acurácias médias de classificação das técnicas propostas para o *dataset* coletado na RDE, considerando tanto os componentes íntegros quanto os defeituosos.

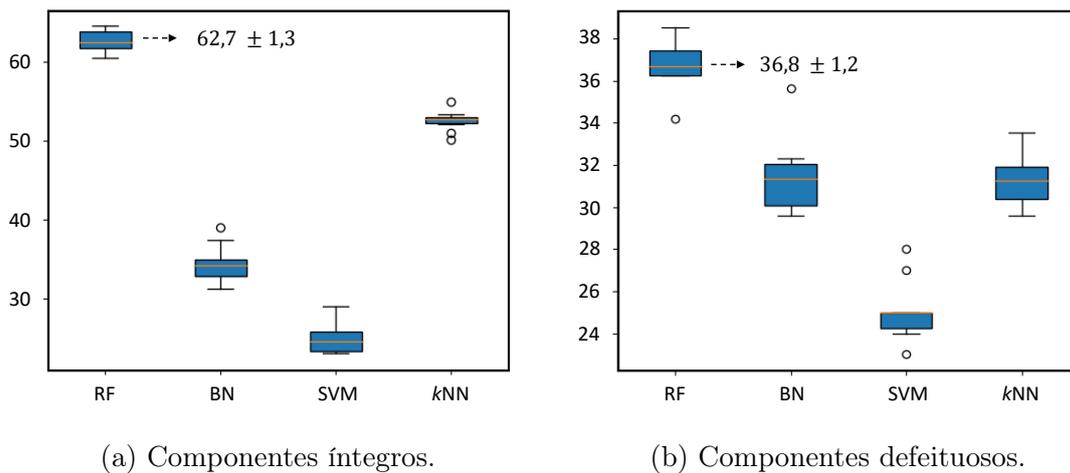


Figura 5.19: Acurácia de classificação das técnicas propostas para o *dataset* coletado na RDE - valores em %.

Através destes gráficos, pode-se inferir que a alta variabilidade de planos de fundo

do ambiente realista proporcionou uma perda acentuada de desempenho dos sistemas de classificação. No pior cenário avaliado, relativo aos isoladores defeituosos, ocorreu uma redução de acurácia de 61,2 pontos percentuais. Este comportamento indesejado, relacionado ao *dataset* de componentes defeituosos, também pode estar associado à semelhanças de regiões dos planos de fundo com as características visuais das inconformidades nos isoladores, a exemplo de alterações acentuadas de coloração e/ou textura. Por conseguinte, a partir das principais dificuldades relatadas, observa-se, como uma abordagem promissora para viabilizar a composição de sistemas inteligentes de inspeção, o desenvolvimento de técnicas direcionadas à redução do nível de sensibilidade dos modelos computacionais a fontes de interferência de planos de fundo.

5.4 Diagnóstico de Defeitos por Aprendizagem Profunda

Para as simulações computacionais que envolveram a classificação de material/*design* de componentes isoladores e a identificação de seus respectivos *status* de conformidade, foi realizada a abordagem experimental apresentada na Subseção 4.3.3. Seguindo o que foi descrito na etapa metodológica, foram considerados os casos de classificação de imagens para uma única tarefa (Tipos 1 e 2) e também para a configuração *Multi-Task Learning* (Tipo 3). Um sumário dos melhores resultados de classificação obtidos para cada arquitetura avaliada pode ser visualizado na Tabela 5.6, correspondendo aos valores médios de acurácia calculados. Os resultados apresentados na referida tabela foram obtidos através da técnica de validação cruzada adaptada, conforme procedimento detalhado na Seção 4.6.

Dentre as redes convolutivas trabalhadas neste estágio da pesquisa, a VGG-19 foi a que obteve os melhores desempenhos de classificação para todas as abordagens testadas. Para a tarefa de identificação do modelo de isolador, a melhor configuração foi a do Tipo 1 (somente componente no treinamento) com acurácia média de 90,3%. Para classificação da presença de defeitos visíveis, a abordagem do Tipo 3 apresentou os melhores resultados, com os valores de Acc e F1 de 85,0% e 0,76, respectivamente. As Figuras 5.20a e 5.20b apresentam os *Boxplots* de acurácia de

Tabela 5.6: Valores médios obtidos com a técnica de Validação Cruzada Adaptada (VCA) – acurácia em %.

Arquiteturas	Tipo 1	Tipo 2	MTL (Tipo 3)	
	Somente Material	Somente Defeito	Material	Defeito
VGG-19	90,3 ± 1,3	68,1 ± 4,4	87,6 ± 1,6	85,0 ± 1,4
Inception-v3	80,1 ± 2,9	61,4 ± 6,0	67,6 ± 1,4	64,9 ± 4,3
ResNet-50	26,2 ± 1,1	50,0 ± 2,1	25,6 ± 1,3	41,1 ± 4,6

classificação do modelo de isolador e o *F1-Score* de discriminação do *status* de conformidade dos componentes. Através dos referidos gráficos é possível inferir que a técnica *multi-task learning* melhorou consideravelmente o desempenho de classificação de defeitos, em comparação com cada técnica Tipo 1 e Tipo 2 isoladamente, como também reduziu a variabilidade dos resultados inerentes ao processo de aprendizagem. Esta última informação pode ser observada na menor distância entre o primeiro e o terceiro quartil apresentada pelo *boxplot* para abordagem de classificação de defeitos pela técnica de múltiplas saídas (MTL-Def) em relação à condição isolada de identificação.

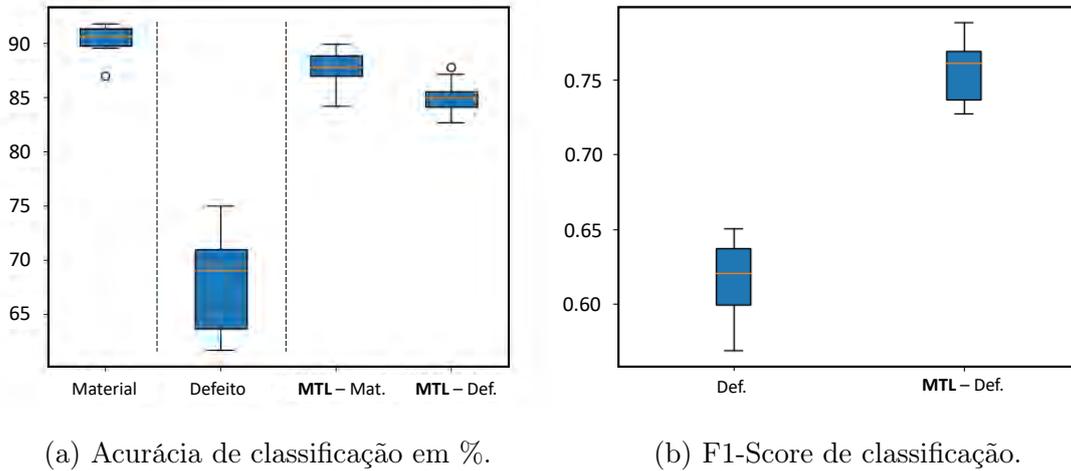


Figura 5.20: Gráficos Boxplots para o desempenho de classificação da CNN VGG-19.

Os desempenhos de classificação da VGG-19 foram posteriormente discretizadas de acordo com as diferentes especificações das sequências de treinamento (estágios A, B, C, D, E e F) - detalhadas na Seção 5.1. A Figura 5.21a apresenta a Acc da referida RNC do Tipo 1 para 100 épocas de treinamento, considerando as etapas de

simulação de I até VIII. Este gráfico indica que todas as implementações propostas promoveram melhorias na acurácia do modelo. O caso onde não são empregados o gerador de imagens e o ajuste-fino de camadas convolucionais (B) corresponde ao primeiro ponto do gráfico. A Figura 5.21b, por sua vez, ilustra os resultados da VGG-19 do Tipo 3 para identificação de defeitos em função da implementação de planos de fundo e detalhes realísticos (D e E). Em contraste com o gráfico anterior, a adição de amarrações metálicas e condutores livres reduziu a precisão do modelo para classificação de inconformidades. Os elementos do estágio E aparentemente podem produzir sobreposições em áreas defeituosas das imagens dos componentes, conforme pode ser visualizado na Figura 5.29d, que demonstra dois exemplos de imagens após os estágios D e E.

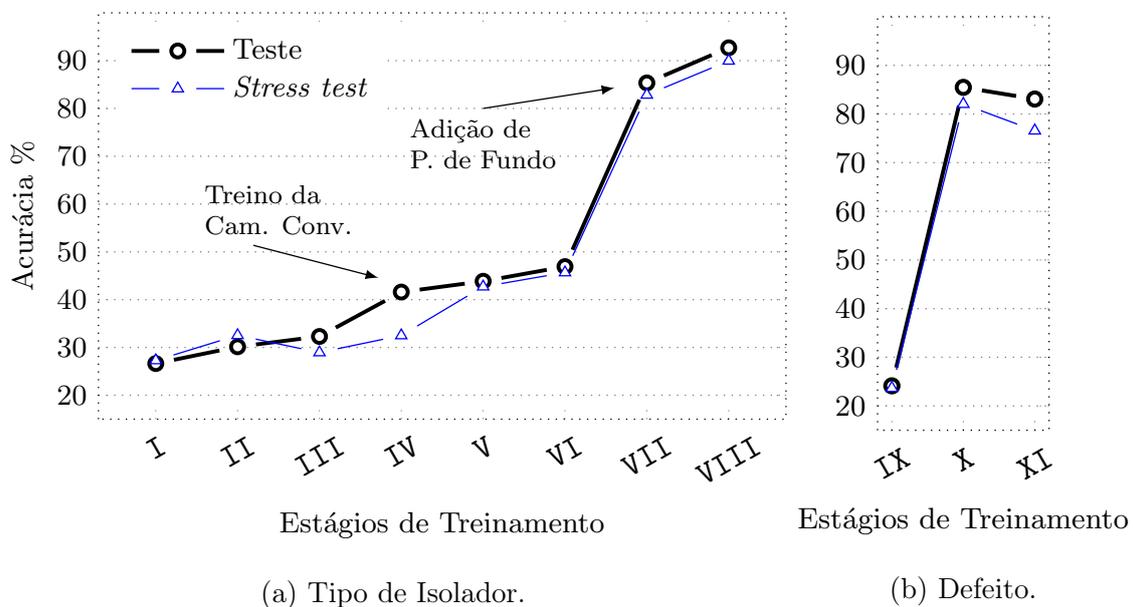
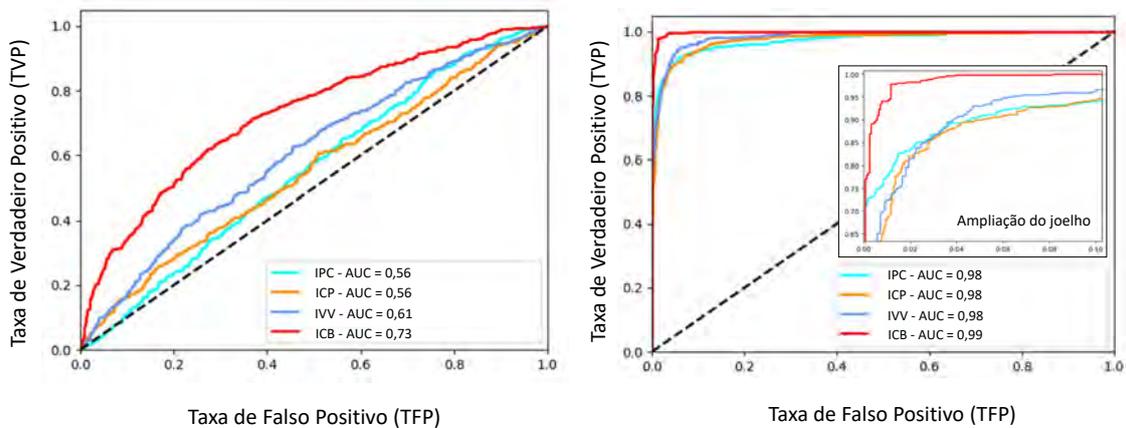


Figura 5.21: Acurácia dos classificadores de acordo com os estágios de treinamento: **I)** treinamento convencional; **II)** Inclusão de rotina de *data augmentation* (A); **III)** segmentação das imagens do dataset de treinamento (C); **IV)** ajuste das camadas $C_{(n-1)}$ e $C_{(n)}$ (B); **V)** inclusão dos estágios A e B; **VI)** inclusão dos estágios A, B e C; **VII)** adição de planos de fundo (D) e adição dos estágios A, B e C; **VIII)** inclusão de componentes menores (E) e adição dos estágios A, B, C e E; **IX)** adição dos estágios A, B e C e adoção da modalidade de treino *multi-task learning* (F); **X)** adição dos estágios A, B, C, D e F; **XI)** adição dos estágios A, B, C, D, E e F.

Em ambos os gráficos apresentados na Figura 5.21, é possível verificar que o

estágio de adição de planos de fundo (D) demonstrou considerável influência nos indicadores de desempenho do modelo. Este estágio do GIA proporcionou um incremento de precisão de, respectivamente, 39 e 61 pontos percentuais para classificação de isoladores e de defeitos. A implementação do estágio de *data augmentation* (A), por sua vez, tornou o modelo menos suscetível a alterações visuais nos objetos avaliados, trazendo os valores de *stress test* mais próximos dos resultados obtidos para os testes convencionais. Considerando o desempenho dos *ensemble tests*, foram identificadas melhorias na estabilidade dos modelos, aproximando-se dos melhores valores de teste obtidos, alcançando as acurácias de 91,5% para o Tipo 1 e 84,5% para o Tipo 2.

A Figura 5.22 ilustra as curvas ROC para a VGG-19 Tipo 1 sem o GIA (Figura 5.22a) e o mesmo modelo com a adição dos estágios de treinamento A, B, C, D e E (Figura 5.22b). É possível observar que as técnicas propostas melhoraram os desempenhos das curvas ROC (valores mais elevados de sensibilidade e especificidade) e aumentaram consideravelmente as AUC para valores acima de 0,98.



(a) Sem adição de GIA.

(b) Com GIA para: A, B, C, D e E.

Figura 5.22: Curvas ROC para a VGG-19 do Tipo 1 de acordo com as especificações de treinamento.

Para obtenção de informações visuais do comportamento da sensibilidade dos modelos propostos para determinados estágios de simulação, foram implementadas rotinas para geração de *mapas de calor*, conforme apresentado na Subseção 4.3.3. Como abordagem complementar aos mapas de calor, avaliou-se a influência do tipo de Camada de Agrupamento (CA) no comportamento da matriz $h_{x,y}$ (equação 4.14).

A VGG-19, configurada nesta pesquisa com as CAs do tipo *max-pooling*, foi posteriormente alterada para conter camadas de agrupamento do tipo *average-pooling*. Pelos resultados obtidos, ilustrados nas Figuras de 5.23 a 5.26, torna-se visualmente perceptível a influência da adição de planos de fundo (D) no desempenho dos algoritmos trabalhados. O modelo que nunca viu um plano de fundo no treinamento, apresentado nas Figuras 5.23 e 5.25, se “distrai” mais facilmente por ele, enquanto o modelo treinado pela introdução dos planos de fundo realísticos, presentes nas Figuras 5.24 e 5.26, fica com o foco mais preciso nos objetos de interesse para classificação.

Para as configurações da VGG-19 em função das camadas de agrupamento supracitadas, os mapas de calor gerados pelos modelos do tipo *average-pooling* possuem áreas realçadas mais contínuas e bem definidas (Figura 5.26), porém a configuração do tipo *max-pooling* delimita de forma mais eficiente as regiões desejadas das imagens (Figura 5.24). Os mapas de calor produzidos nesta etapa da pesquisa podem ser empregados futuramente para o desenvolvimento de rotinas para detecção de *key-points* [227], como também para auxiliar na criação de algoritmos que realizem as tarefas de classificação e segmentação de forma combinada.

A Figura 5.27 fornece as matrizes de confusão para a arquitetura VGG-19 do Tipo 1 sem o gerador de imagens (Figura 5.27a) e para a configuração em que o mesmo apresentou o melhor desempenho de Acc (Figura 5.27b). Através dessas matrizes, é possível visualizar que o Isolador Polímero Cinza (IPC) possui elevada predominância nas classificações da RNC do Tipo 1. Esta circunstância pode estar relacionada às características de cor e textura das estruturas de sustentação das RPDA serem mais próximas das características do IPC. Realizando o treinamento sem o emprego do GIA, os modelos não possuem um nível adequado de imunidade às interferências provenientes dos planos de fundo, características de ambientes externos de RPDA. Deste modo, sem a existência de uma forma de treinamento adequada para os modelos, pode-se inferir que os componentes isoladores priorizados para a classificação são aqueles que possuem características visuais mais próximas das referidas fontes de interferência.

Considerando agora a Figura 5.27b, que apresenta os melhores desempenhos de classificação da VGG-19 relativos aos *status* de conformidade dos isoladores de dis-

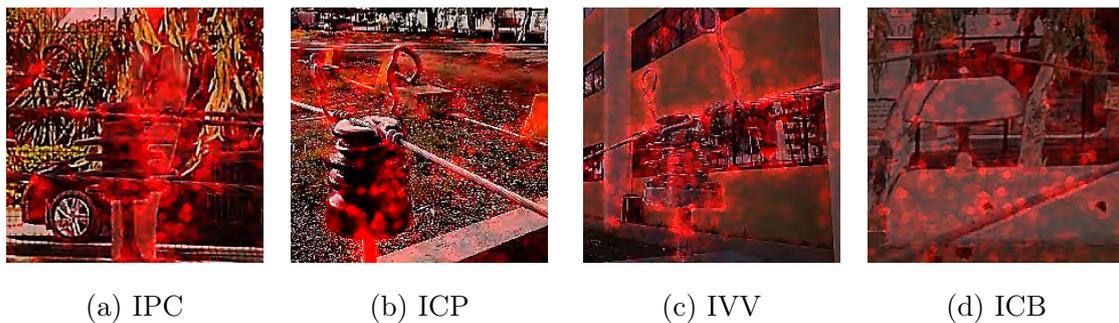


Figura 5.23: MCOs + *max-pooling* obtidos sem a adição de planos de fundo.

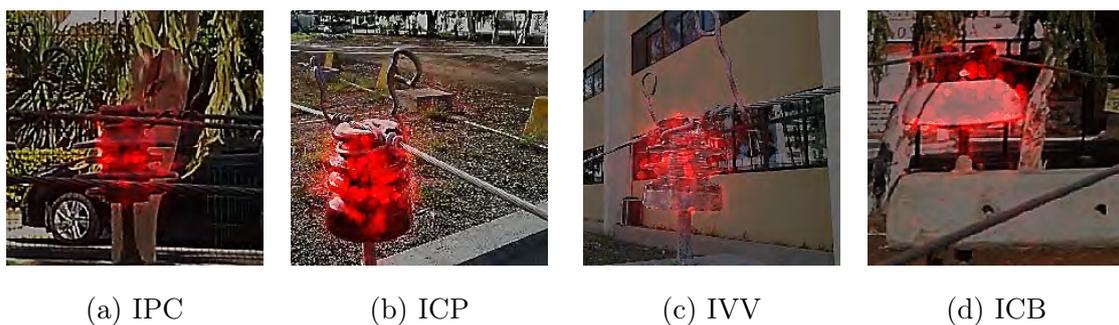


Figura 5.24: MCOs + *max-pooling* com a adição de planos de fundo.

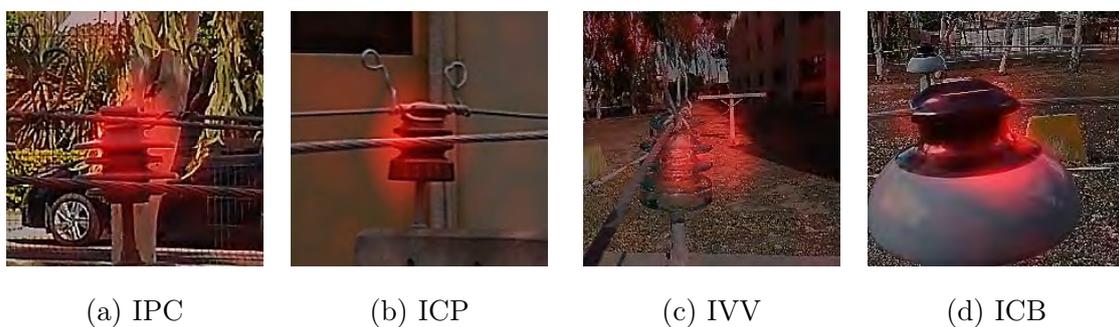


Figura 5.25: MCOs + *average-pooling* obtidos sem a adição de planos de fundo.

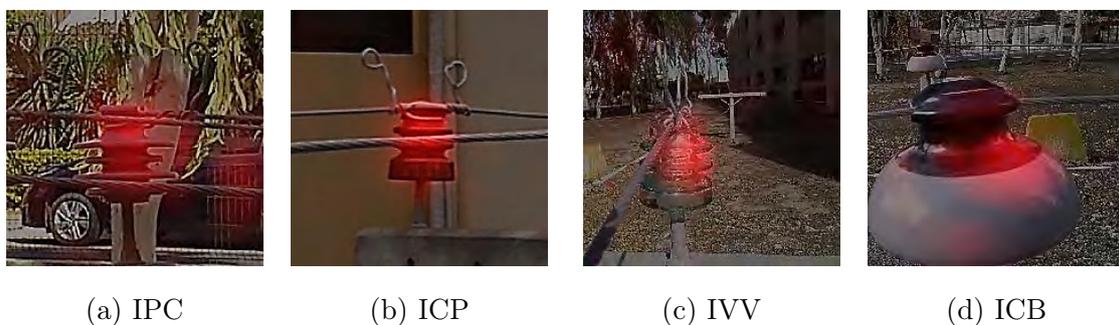


Figura 5.26: MCOs + *average-pooling* obtidos com a adição de planos de fundo.

		Predito					Predito						
		IPC	ICP	IVV	ICB		IPC	ICP	IVV	ICB			
Real	IPC	508	1	11	0	97,7%	Real	IPC	469	32	6	13	90,2%
	ICP	482	5	27	6	1,0%		ICP	29	465	7	19	89,4%
	IVV	494	1	24	1	4,6%		IVV	19	19	482	0	92,7%
	ICB	494	1	8	17	3,3%		ICB	7	1	0	512	98,5%
		25,7%	62,5%	34,3%	70,8%	26,6%			89,5%	89,9%	97,4%	94,1%	92,7%

(a) Sem adição de GIA.

(b) Com GIA para: A, B, C, D e E.

Figura 5.27: MCs da VGG-19 (Tipo 1) para classificação do tipo de isolador.

tribuição, podemos verificar que, em ambos os casos, a precisão obtida ficou acima de 89%. Através dos exemplos de erros de classificação de imagens apresentados na Figura 5.29, pode-se inferir que parte deles podem estar associados a certas características visuais que podem ocorrer em imagens de RPDA coletadas por VANT, tais como: diferenças agressivas no formato e textura (Grupo FT); variação intensa de luminosidade (Grupo IL) e sobreposição de imagens (Grupo SI).

		Predito		
		Íntegro	Defeituoso	
Real	Íntegro	1.391	209	86,9%
	Defeituoso	93	387	80,6%
		93,7%	64,9%	85,5%

Figura 5.28: MC da VGG-19 (Tipo 3) para classificação de inconformidades.

As Figuras 5.29a, 5.29b e 5.29c ilustram exemplos de imagens em que os modelos propostos não conseguiram classificar o componente corretamente. Estas informações demonstram uma possível margem para melhoria de procedimentos de coleta de imagens/vídeos, além do desenvolvimento de possíveis técnicas complementares para o processamento da informação. Algumas abordagens se mostram promissoras, podendo ser mencionadas: **(1)** desenvolvimento de procedimentos operacionais com VANTs para coleta de dados, avaliando, por exemplo, condições de iluminação e distância/ângulos de sobrevoo; **(2)** otimização do GIA (através da inclusão de uma maior quantidade de características do ambiente real no dataset de teste), **(3)** implementação de estágios de segmentação e de técnicas *Multi-View* focadas na melhoria

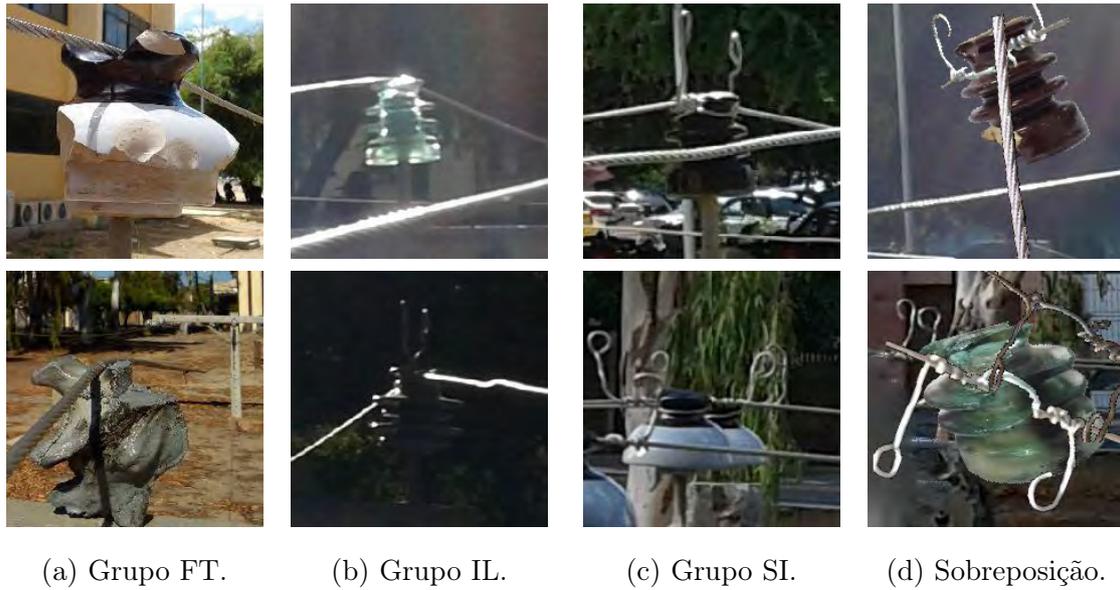


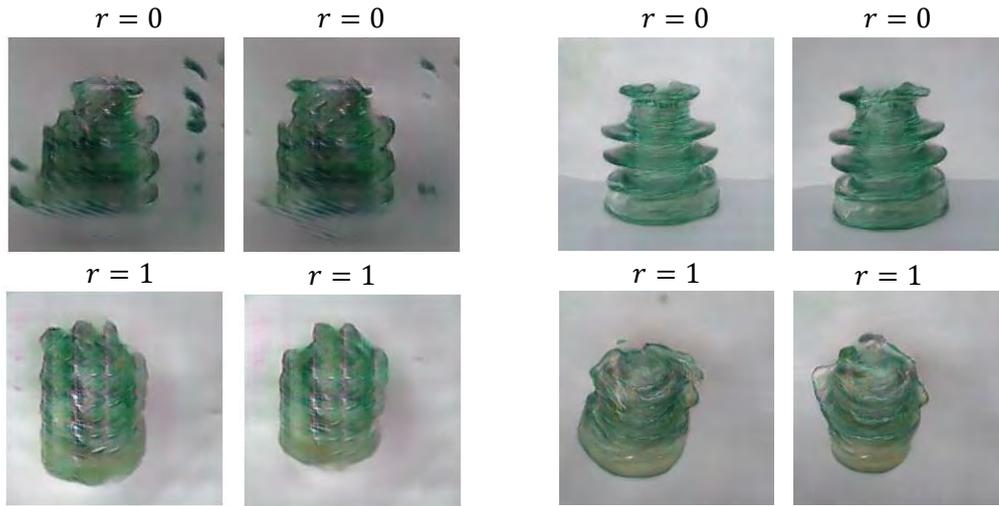
Figura 5.29: Exemplos de erros de classificação de isoladores.

de desempenho dos modelos de classificação de imagens.

5.5 Geração de *dataset* Artificial com cGANs

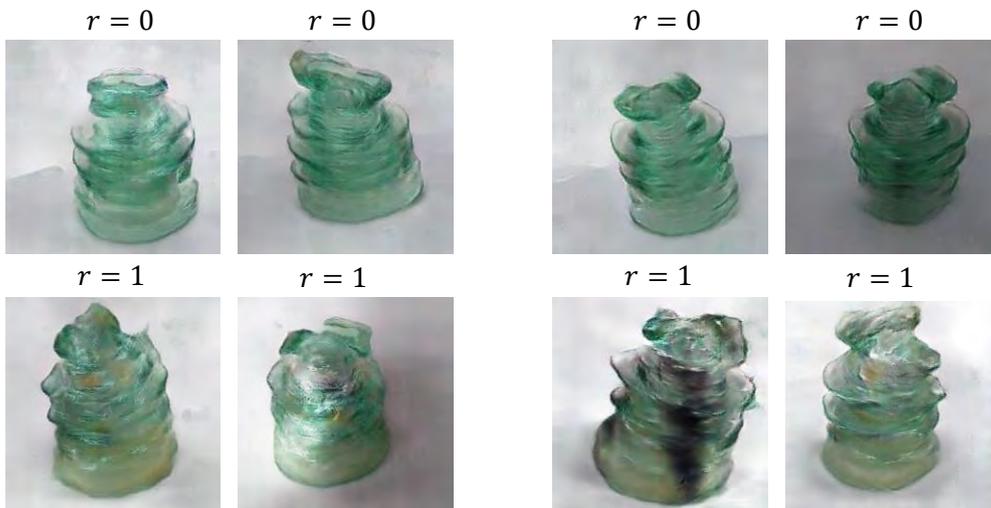
A partir da metodologia apresentada na Subseção 4.4.1, foram conduzidos uma série de experimentos direcionados ao desenvolvimento de técnicas de geração de imagens artificiais de componentes isoladores de distribuição de RPDA. Em suma, objetivou-se avaliar se as cGANs propostas, juntamente com as estratégias de treinamento concebidas para esta pesquisa, foram capazes de produzir imagens de componentes com um nível adequado de diversidade, em que também são contempladas alterações visuais características de defeitos conhecidos, tais como forma, cor e textura. Este estágio experimental pode ser dividido em quatro etapas principais, enumeradas como: **1)** treinamento convencional das cGANs; **2)** aplicação da técnica de regularização do tipo *data augmentation*, descrita na Seção 4.2; **3)** implementação de um fator de diversidade $\mathcal{L}_z(\cdot)$ na função objetiva das cGANs – de acordo com a equação 4.15; e **4)** adição de termo variável de taxa de aprendizagem, $\lambda_R(\cdot)$, através da extração de parâmetros de textura (*descritores de Haralick*), seguido o procedimento de cálculo adotado na equação 4.19.

Os *datasets* escolhidos para o estudo de caso foram o DATA-INT e DATA-DEF, pertencentes a classe de componente Isolador de Vidro Verde (IVV). Para cada



(a) Treinamento convencional.

(b) *Data Augmentation* (DA).



(c) Fator de diversidade ($\mathcal{L}_z(\cdot)$).

(d) Taxa de aprendizagem ($\lambda_R(\cdot)$).

Figura 5.30: Exemplos de imagens geradas pelas cGANs para diferentes modalidades de treinamento.

um dos referidos experimentos, foram realizadas um total de 10.000 épocas de treinamento. Considerando os principais hiper-parâmetros inerentes ao processo de aprendizagem das cGANs, foram realizadas as seguintes especificações: definição da modalidade de entropia binária cruzada para as funções objetivas, $\mathcal{L}_A(\cdot)$ e $\mathcal{L}_B(\cdot)$, pertencentes às saídas *multi-task learning* do discriminador (apresentadas na Seção 3.11); ajuste das taxas de aprendizagem em 10^{-4} para as funções de otimização Adam pertencentes aos modelos gerador (G) e discriminador (D); definição do tamanho do mini-lote de treinamento (m) em 32 imagens tanto para os *datasets* reais

quanto para os exemplos falsos. Ao final das simulações, foram geradas 1.000 imagens artificiais para cada modelo gerador trabalhado. A Figura 5.30 exemplifica imagens de isoladores produzidas pelas cGANs para as diferentes abordagens implementadas nesta etapa da pesquisa. A variável condicional r determina o *status* de conformidade do componente gerado artificialmente, sendo $r = 0$ para isoladores íntegros e $r = 1$ para os defeituosos.

Para a Figura 5.30a, a modalidade de treinamento convencional das cGANs resultou na condição de *mode collapse* do modelo gerador. Deste modo, as imagens produzidas por G apresentaram características visuais pouco realistas e com um baixo nível de diversidade, em que apenas pequenos subconjuntos de distribuição são representados pela saída do gerador. Para o segundo caso, presente na Figura 5.30b, é perceptível que os exemplos produzidos pela rede generativa apresentaram características visuais mais realistas do que no caso anterior, contudo o nível de diversidade do *dataset* continuou indicando baixo desempenho, pois o modelo gerador privilegiou apenas determinados ângulos e posições dos objetos presentes nas imagens. Outro ponto que vale ser enfatizado nesta abordagem está relacionado às características visuais das imagens de defeitos produzidas, tendo em vista que nenhum exemplo conseguiu reproduzir o tipo de inconformidade que envolve alteração de cor do componente (mancha de coloração escura no corpo do isolador).

Considerando agora a terceira modalidade de treinamento, apresentada na Figura 5.30c, é possível observar estas cGANs conseguiram produzir imagens realistas e com maior nível de diversificação se comparado aos dois casos anteriores. No entanto, essa abordagem também não conseguiu reproduzir os subconjuntos de imagens de treino que envolvem inconformidades com alterações acentuadas de coloração do componente. Este arranjo de redes convolutivas produziu majoritariamente defeitos com alterações de formato – classe de inconformidade condizente com a quebra e fratura de componentes.

Para o último caso trabalhando, exemplificado na Figura 5.30d, infere-se que o parâmetro variável de taxa de aprendizagem, $\lambda_R(\cdot)$, conseguiu proporcionar um melhor ajuste no termo de diversidade ($\mathcal{L}_z(\cdot)$) incorporado a função do custo do modelo gerador. Esta observação está relacionada a dois fatores principais: **1)** o modelo gerador conseguiu gerar imagens realistas e com o maior nível de diversi-

dade entre todos os casos avaliados e **2**) foi capaz de produzir defeitos com alterações significativas de coloração do componente, condição não identificada nas abordagens anteriores. De forma quantitativa, das 1.000 imagens de componentes defeituosos produzidas nesta etapa, 26,1% apresentaram este tipo de específico de inconformidade.

A Figura 5.33 ilustra, com maior nível de detalhamento, exemplos de imagens de isoladores produzidas pela quarta e última abordagem. Para facilitar a visualização dos componentes e viabilizar a avaliação estatística dos resultados, os *datasets* de imagens passaram por uma etapa posterior de segmentação através do algoritmo U-Net, apresentado na Subseção 4.2.1. Após a referida etapa de segmentação, onde foram subtraídas eventuais fontes de interferências presentes nos planos de fundo, gerou-se os parâmetros IS e FID dos conjuntos de imagens de componente íntegros e defeituosos relativos a abordagem exemplificada nas Figuras 5.30d e 5.33, em função dos *datasets* reais de treino do componente IVV. A Tabela 5.7 faz um sumário destes resultados.

Tabela 5.7: Valores de IS e FID para os *datasets* de imagens produzidos pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$, considerando os *datasets* íntegros (INT) e defeituosos (DEF) da base de dados real de IVV.

Técnica	DATA-INT	DATA-DEF	cGAN-INT	cGAN-DEF
IS	$1,8 \pm 0,1$	$1,8 \pm 0,2$	$1,5 \pm 0,1$	$1,5 \pm 0,1$
FID	—	—	4,9	3,6

A tabela mencionada indica que os níveis de diversidade dos *datasets* reais e falsos são muito próximos entre si, haja vista que as bases de dados de imagens DATA-INT e DATA-DEF apresentam valores de IS muito próximos aos correspondentes conjuntos de imagens produzidos pelas cGANs. Outro ponto de mérito reside nos baixos valores de FID apresentados pelos *datasets* cGAN-INT e cGAN-DEF quando comparados, respectivamente, às base de dados DATA-INT e DATA-DEF. Avaliando agora a capacidade de generalização do arranjo de cGANs proposto em relação a outras classes de componentes isoladores, a Figura 5.32 exemplifica imagens de ICP íntegros e defeituosos produzidas através da modalidade de treinamento com termo

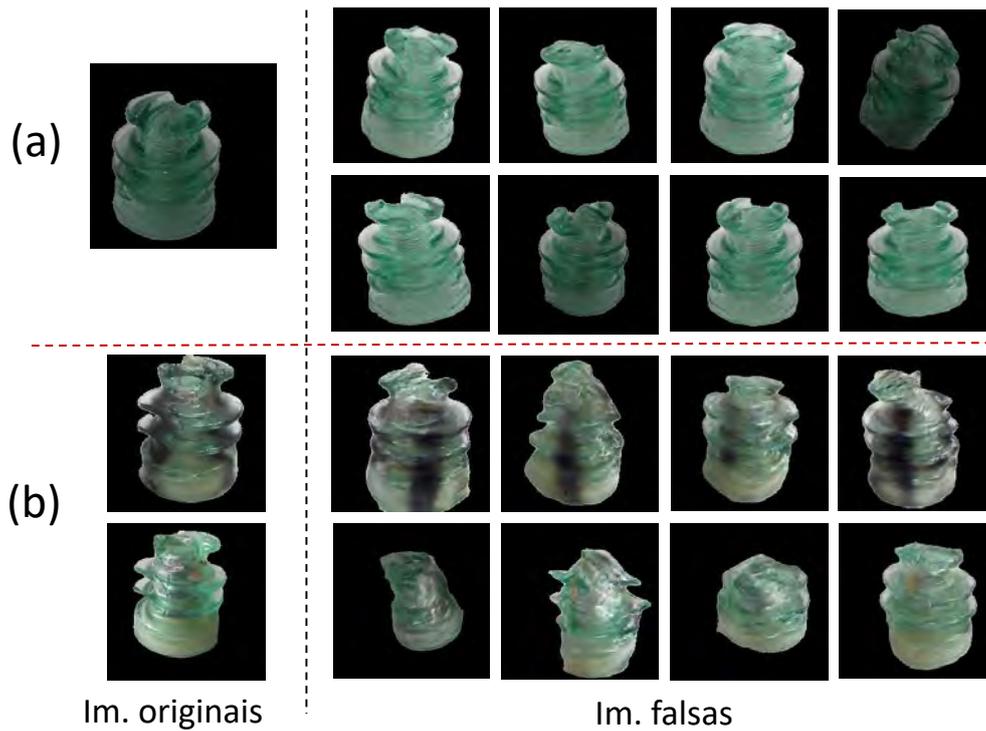


Figura 5.31: Exemplos de imagens segmentadas do *dataset* real de IVV (posicionadas à esquerda) e amostras falsas geradas pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$; a) imagens de isoladores íntegros e b) exemplos de componentes defeituosos.

de regularização, $\lambda_R(\cdot)$.



Figura 5.32: Exemplos de imagens de ICP íntegros e defeituosos produzidos pelas cGANs com taxa variável de aprendizagem, $\lambda_R(\cdot)$; componentes íntegros posicionados acima e defeituosos na parte inferior.

Através dessa figura, é possível observar que a técnica proposta para geração de imagens também foi capaz de incorporar características visuais de componentes cerâmicos. Os padrões de defeitos destes componentes também estão presentes nas imagens, a exemplo de alterações típicas de formato, cor e textura. Por conseguinte, este estudo abre margem para o desenvolvimento de modelos computacionais geradores de *datasets* artificiais e podem vir a contribuir na implementação de técnicas diferenciadas de *data augmentation* com o propósito de aumentar a capacidade de generalização de modelos classificadores de imagens de componentes de RPDA.

5.6 Técnica de *Background Invariance*

Considerando a etapa metodológica descrita na Subseção 4.4.2, foram empregadas duas abordagens experimentais para a análise de desempenho da técnica proposta de *background invariance*, denominada, para fins de simplificação da nomenclatura, de **Back-Inv**. As referidas abordagens consistem em um estudo comparativo de desempenho de classificação do modelo discriminador, f_d , para dois tipos de base de dados distintas, nomeadamente: **1)** *datasets* de imagens, em escala de cinza, intitulados de MNIST e Fashion MNIST [215]; **2)** *dataset* de imagens de isoladores de distribuição coletadas na ECI e RDE – procedimento descrito na Seção 4.1.

Para a primeira abordagem experimental, a técnica proposta para geração de planos de fundo adversários e treinamento de f_d foi comparada com a forma convencional de aprendizagem (treinamento com o *dataset* original) e com a técnica de *attention mechanism* (Att), concebida por Guo et al. [214]. As Tabelas 5.8 e 5.9 apresentam as acurácias (Acc) de classificação das referidas técnicas em função de alterações de plano de fundo nos *datasets* de teste, exemplificadas na Figura 4.29. A última coluna representa a média (Avg) dos valores obtidos para todas as modalidades de plano de fundo inseridas no *dataset* de teste.

Os resultados apresentados nas referidas tabelas mostram melhorias significativas de desempenho de classificação proporcionadas pela técnica de *background invariance*. Em termos de valores médios, os ganhos de acurácia ficam em torno de 50 a 100 pontos percentuais. Estes valores são muito superiores aos resultados fornecidos pela técnica de *attention mechanism*, que proporcionou apenas uma leve melhoria

de desempenho para o classificador de base f_d . Deste modo, somente o treinamento com planos de fundo adversários foi capaz de gerar um nível de imunidade consistente a interferências de planos de fundo. A Figura 5.33 ilustra exemplos da natureza dinâmica do processo de geração de planos de fundo adversários durante o treinamento do modelo computacional. Alguns dos erros de classificação cometidos pelo modelo treinado de forma tradicional são exibidos na Figura 5.34.

Tabela 5.8: Tabela das Acc de classificação do *dataset* MNIST: (A) listras; (B) tabuleiro; (C) borda; (D) círculos; (E) relógio e (F) aleatório - valores em %.

Método	A	B	C	D	E	F	Avg
Tradicional	39,0	24,3	61,4	32,9	19,7	11,2	31,4
Att	28,1	26,8	57,5	40,1	29,3	25,1	34,5
Back-Inv	92,3	76,8	93,1	93,7	70,8	86,2	85,5

Tabela 5.9: Tabela das Acc de classificação do *dataset* Fashion MNIST: (A) listras; (B) tabuleiro; (C) borda; (D) círculos; (E) relógio e (F) aleatório - valores em %.

Método	A	B	C	D	E	F	Avg
Tradicional	21,3	24,6	36,9	28,5	29,6	16,8	26,7
Att	18,2	20,1	51,8	26,0	31,8	36,2	30,7
Back-Inv	62,9	61,5	66,5	60,9	60,8	45,9	59,8

Considerando agora a segunda abordagem experimental, que envolve a avaliação da base dados de isoladores de distribuição, foi realizado um estudo comparativo dos desempenhos de identificação para oito sistemas de classificação, enumerados como: **I**) treinamento convencional da VGG-19 com o *dataset* da ECI (presente na Seção 5.4); **II**) utilização da técnica de *attention mechanism* [214]; **III**) fusão entre técnicas de extração de atributos com emprego de redes neurais [53]; **IV**) geração de planos de fundo com ruídos de distribuição uniforme; **V**) emprego de arranjo híbrido de modelos inteligentes (apresentado na Seção 4.3.2); **VI**) técnica de classificação por atributos geométricos (descrita na Seção 4.3.1; **VII**) técnica proposta de *background invariance* e **VIII**) implementação de gerador artificial de imagens (apresentado na

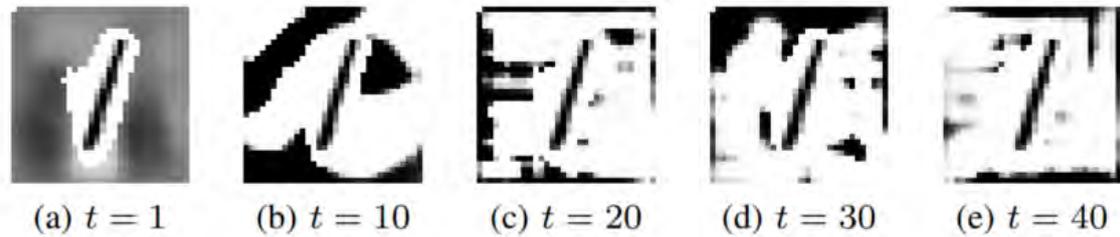


Figura 5.33: Exemplos de geração de planos de fundo dinâmicos no decorrer das épocas (t).

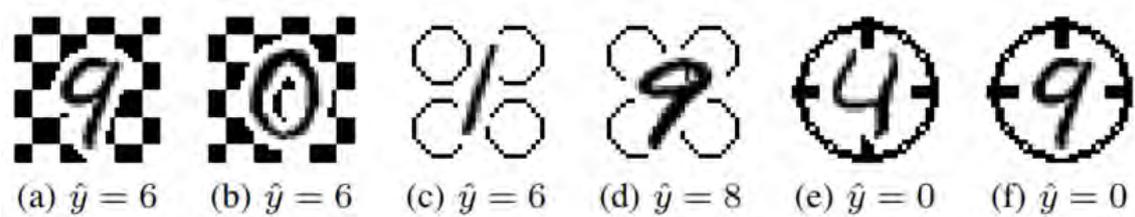


Figura 5.34: Exemplos de erros de classificação realizados pelo classificador treinado de forma convencional, que não ocorrem no treino de forma de adversária.

Seção 4.2).

A Tabela ?? apresenta os melhores desempenhos de classificação para cada uma das técnicas mencionadas. É possível observar que o modelo base (VGG-19), treinado com o GIA, apresentou os melhores desempenhos de classificação de imagens. No entanto, vale salientar que esta técnica, em uma de suas rotinas internas, é alimentada com planos de fundo realistas durante as iterações do estágio de treinamento, enquanto a abordagem *Back-Inv*, que apresentou um desempenho levemente inferior, só se utiliza de imagens coletadas em ambientes controlados. Deste modo, ela ainda pode se mostrar como uma técnica vantajosa quando não se tem conhecimento da distribuição de probabilidades dos planos de fundo do *dataset* de teste.

Para avaliação visual do impacto da técnica de *background invariance*, foram gerados os mapas de calor dos modelos base para a forma de treinamento convencional e para a abordagem proposta de *background invariance*. Estas imagens mostram com clareza que a técnica proposta torna o classificador significativamente menos sensível aos planos de fundo. Por conseguinte, considerando as informações apresentadas nos experimentos, pode-se inferir que esta técnica de geração de planos de fundo adversários pode ser adaptada para diferentes propósitos e possuem o potencial de otimizar o processo de aprendizagem de classificadores de imagens que

Tabela 5.10: Acurácias de identificação de material/*design* do componente isolador de acordo com o sistema de classificação adotado – imagens pertencentes ao *dataset* coletado na RDE (valores em %).

Técnica	Média	Máximo
I) Nenhum (VGG-19) [33]	$26,5 \pm 2,3$	27,8
II) Att (VGG-19) [214]	-	45,8
III) Fusão (RNA-MLP) [53]	$49,0 \pm 6,0$	55,0
IV) Ruído (VGG-19) [35]	-	59,6
V) Arranjos Híbridos (ResNet-50 + RF) [34]	$62,7 \pm 1,3$	64,0
VI) P. Geométrico (U-Net + HMM) [31, 32]	$76,4 \pm 1,6$	78,0
VII) Back-Inv (VGG-19) [35]	-	88,7
VIII) GIA (VGG-19) [33]	$90,3 \pm 1,3$	92,7

utilizam *datasets* de treinamento sujeitos a macro interferências de planos de fundo.

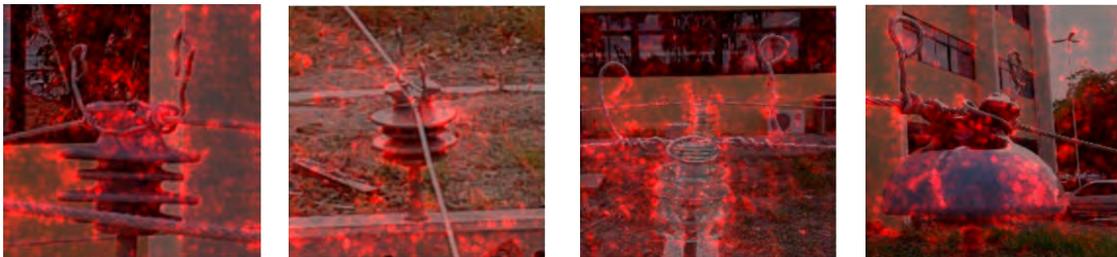


Figura 5.35: Mapas de Calor (MCO) para a forma de treinamento convencional.



Figura 5.36: Mapas de Calor (MCO) para a forma de treinamento através da técnica de *background invariance* (Back-Inv).

Capítulo 6

Considerações Finais

Esta pesquisa de doutorado apresentou um conjunto de técnicas inteligentes direcionadas a tarefas específicas de inspeção visual automática de isoladores de distribuição encontrados em RPDA através da implementação de algoritmos de aprendizado de máquina e de processamento digital de imagens. Como abordagem adicional para atender a demandas desta pesquisa, foram implementadas técnicas de *data augmentation*, transferência de conhecimento, geração artificial de imagens e redução de sensibilidade dos modelos a macro interferências em planos de fundo, abordagem conhecida como *background invariance*. Os principais objetivos envolvidos nos projetos de sistemas de classificação realizados nesta pesquisa residiram na identificação do tipo de material/*design* de componentes isoladores de distribuição e na especificação dos seus respectivos *status* de conformidade.

Tendo em vista a escassez de pesquisas ou *benchmarks* relacionados às tarefas de inspeção visual inteligentes/automatizadas de isoladores de distribuição, foram desenvolvidas estratégias específicas direcionadas à aquisição e tratamento de *datasets* de imagens. Parte da base de dados foi adquirida dentro de uma Estação de Coleta de Imagens (ECI), com iluminação e ângulos controlados, enquanto outra parcela das imagens foi coletada em ambiente externo, através de uma Rede de Distribuição de Ensaio (RDE) – com aparência realista, não energizada e construída para fins acadêmicos. No total, foi coletado um conjunto de dados contendo um número próximo a 5.000 imagens e foram produzidas 480 máscaras de segmentação.

Outro ponto de interesse neste trabalho foi o desenvolvimento de um Gerador de Imagens Artificiais (GIA) destinado a produzir conjuntos de dados de treinamento.

Conforme um dos objetivos propostos, o GIA foi capaz de editar digitalmente o *dataset* coletado na ECI e gerar um novo conjunto de imagens com características desejadas de ambientes realistas. Tal implementação proporcionou ganhos de 66 pontos percentuais na classificação de componentes e 61 pontos percentuais na identificação de defeitos, em relação a modalidade de treinamento convencional – sem o emprego dos estágios de treinamento pertencentes ao GIA e considerando como conjunto de teste o *dataset* proveniente da RDE.

Considerando o desenvolvimento de sistemas de classificação por atributos geométricos, os resultados obtidos nesta pesquisa indicaram que o modelo HMM proposto foi capaz de discriminar o material/*design* do componente, com um nível de acurácia média geral de 90%. Para estes casos, foram avaliados isoladores de distribuição com diferentes formatos, tanto íntegros quanto defeituosos. Observou-se que, diante de testes comparativos realizados com outras técnicas, a aplicação do arranjo de HMMs proposto é de fato uma alternativa viável aos difundidos sistemas empregados em classificação de imagens, como redes perceptron, SVM e *Random Forest*, obtendo, para o *dataset* trabalhado, resultados superiores às técnicas comparadas.

Para as configurações híbridas de algoritmos inteligentes, objetivou-se desenvolver uma técnica para classificação de componentes isoladores sem que houvesse a necessidade de uma etapa prévia de segmentação. Para o *dataset* coletado na ECI, a arquitetura da ResNet-50 + SVM apresentou o melhor desempenho de classificação dentre as configurações testadas, alcançando uma acurácia média de 95% na classificação do modelo de componente defeituoso e aproximadamente 100% para os não defeituosos. Outro ponto que foi avaliado nesta pesquisa foi o emprego da ResNet-50 como um extrator de atributos. Os 2.048 valores extraídos da penúltima camada achatada desta rede convolutiva foram capazes de fornecer informações úteis para todos os classificadores testados. O Isolador Cerâmico Bicolor (ICB), por exemplo, mostrou um aumento de precisão de 27 pontos percentuais devido ao uso das técnicas híbridas. No entanto, esta abordagem apresentou um alto nível de sensibilidade a macro interferências de plano de fundo, o que comprometeu o seu desempenho de classificação em ambientes realistas de RPDA.

Dentro do escopo desta pesquisa, objetivou-se também classificar o *status* de conformidade do isolador de distribuição – identificando o componente como íntegro

ou defeituoso. Para os casos que envolvem a identificação de múltiplas informações, a arquitetura VGG-19 foi o algoritmo que apresentou os melhores desempenhos de classificação dentre todas as técnicas investigadas. Para as duas modalidades que envolvem uma única tarefa de classificação, denominadas como Tipo 1 (material/*design* do componente) ou Tipo 2 (presença de defeito defeito visível), a VGG-19 apresentou uma acurácia média de 91,5% para a classificação de componentes e 68,1% para identificação do *status* da presença de inconformidades. Para a configuração do tipo *Multi-Task Learning* (MTL), denominada como Tipo 3, esta mesma RNC obteve acurácia de identificação de defeitos de 85% e F1 de 0.75. Isto indica que existe uma relação intrínseca entre as referidas tarefas que possui o potencial de promover melhorias no processo de classificação. Pode-se inferir também que as técnicas de aprendizagem profunda propostas, quando combinadas com os geradores de imagens artificiais, apresentaram um nível adequado de insensibilidade às fontes interferências provenientes de planos de fundo.

Com o intuito de propor soluções correlatas às demandas de investigação do problema trabalhado, foram explorados arranjos generativos adversários para o desenvolvimento de dois tipos de aplicações, sendo elas: geração de *datasets* de imagens artificiais de componentes isoladores defeituosos e desenvolvimento de técnica direcionada ao *background invariance*. Para o primeiro caso, foi realizado um teste de conceito com o sistema de geração de imagens proposto, em que foram adicionados um termo regularização com parâmetro variável de taxa de aprendizado. Nos casos avaliados, as cGANs implementadas conseguiram produzir *datesets* de imagens com um nível de realismo e diversificação adequados. No melhor caso avaliado, foi possível produzir defeitos com diferenças significativas de formato, cor e textura. Considerando a abordagem relacionada ao *background invariance*, os resultados indicam que a técnica proposta tornou o sistema de classificação de imagens trabalhado menos sensível a fontes de interferência de planos de fundo. Deste modo, a referida técnica pode se tornar uma opção desejável de treinamento de algoritmos de classificação de imagens quando não se tem acesso à informações estatísticas suficientes dos planos de fundo dos problemas investigados.

6.1 Perspectivas para Trabalhos Futuros

Os resultados obtidos nesta pesquisa de doutorado motivam a realização de trabalhos futuros envolvendo demandas correlatas a esta investigação. Deste modo, planeja-se investigar novas metodologias para especificação dos tipos de inconformidades e estabelecer estratégias para detecção de novidades não esperadas nos *datasets* de treinamento. Como uma abordagem mais direcionada ao desenvolvimento de uma inovação tecnológica, almeja-se conduzir um estudo de viabilidade técnica e econômica sobre a composição de um sistema inteligente de inspeção visual de isoladores de distribuição em RPDA reais. O planejamento abaixo indica os principais tópicos que podem ser explorados:

1. Desenvolvimento de técnicas de segmentação ordinal de componentes e defeitos a partir da combinação de aplicações previamente trabalhadas nesta pesquisa;
2. Desenvolvimento de técnicas *Multi-View* direcionadas à especificação do tipo de defeito no componente isolador e avaliação do respectivo nível de criticidade;
3. Investigação de estratégias de detecção de novidades direcionadas à identificação de componentes que não estão presentes na base de dados de treinamento;
4. Investigação de técnicas otimizadas de identificação de componentes da RPDA para correção de posicionamento de câmeras acopladas em VANTs;
5. Condução de Estudo de Viabilidade Técnica e Econômica (EVTE) do desenvolvimento de um Sistema de Inteligente de Inspeção Visual de RPDA que aplique as metodologias propostas em RPDA reais.

Referências Bibliográficas

- [1] NATIONAL AGENCY OF ELECTRIC ENERGY, *Underground energy distribution networks: current situation and assessment of the need to improve the associated regulation*, Tech. rep., ANEEL, Brazil, 2014.
- [2] SHORT, T. A., *Electric Power Distribution Handbook*. v. 2. CRC Press, 2014.
- [3] DARESTANI, Y. M., SHAFIEEZADEH, A., DESROCHES, R., “An equivalent boundary model for effects of adjacent spans on wind reliability of wood utility poles in overhead distribution lines”, *Engineering Structures*, v. 128, pp. 441—452, 2016.
- [4] TAJNSEK, V., PIHLER, J., ROSER, M., “Advanced Logistical Systems for the Maintenance of Overhead Distribution Lines Through DCC With the Use of Laser Monitoring”, *IEEE Transactions On Power Delivery*, v. 26, n. 3, pp. 1337–1343, 2011.
- [5] MIAO, X., LIU, X., CHEN, J., et al., “Insulator Detection in Aerial Images for Transmission Line Inspection Using Single Shot Multibox Detector”, *IEEE Access*, 2019.
- [6] MILLER, A., MOHAMMADPOUR, “Power line robotic device for overhead line inspection and maintenance”, *Industrial Robot: An International Journal*, v. 44, pp. 75–84, 2017.
- [7] KATRASNIK, PERNUS, L., “A Survey of Mobile Robots for Distribution Power Line Inspection”, *IEEE Transactions on Power Delivery*, v. 25, pp. 485–493, 2010.

- [8] TOMPKINS, “Look Up and Live: Developing an Effective Overhead Power-Line Safety Program”, *IEEE Transactions on Industry Applications*, v. 50, pp. 4323–4328, 2014.
- [9] KOUSTELLIS, J., ANAGNOSTATOS, S., HALEVIDIS, C., et al., “Contact of heavy vehicles with overhead power lines”, *Safety Science*, v. 26, pp. 951–955, 2011.
- [10] WANG, S., NIU, L., LI, N., “Research on Image Recognition of Insulators Based on YOLO Algorithm”. In: *2018 International Conference on Power System Technology (POWERCON)*, pp. 3871–3874, 2018.
- [11] XIE, X., “A review of recent advances in surface defect detection using texture analysis techniques”, *Electronic Letters on Computer Vision and Image Analysis*, v. 7, n. 3, pp. 1–22, 2008.
- [12] “Deep learning for smart manufacturing: Methods and applications”, *Journal of Manufacturing Systems*, v. 48, pp. 144 – 156, 2018, Special Issue on Smart Manufacturing.
- [13] LUCKOW, A., KENNEDY, K., ZIOLKOWSKI, M., et al., “Artificial Intelligence and Deep Learning Applications for Automotive Manufacturing”. In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3144–3152, 2018.
- [14] YU, K.-H., BEAM, A. L., KOHANE, I. S., “Artificial intelligence in healthcare”, *Nature Biomedical Engineering*, v. 2, n. 10, pp. 719, 2018.
- [15] KAMILARIS, A., PRENAFETA-BOLDU, F. X., “Deep learning in agriculture: A survey”, *Computers and Electronics in Agriculture*, v. 147, pp. 70–90, 2018.
- [16] RAMOS, R., GOMES, S., PRATES, M., et al., “Non-invasive setup for grape maturation classification using deep learning”, *Journal of the Science of Food and Agriculture*, pp. 1–10, 2020.

- [17] PRASAD, R., “Review on Machine Vision based Insulator Inspection Systems for Power Distribution System”, *Journal of Engineering Science and Technology Review*, v. 9, pp. 135–141, 2016.
- [18] NGUYENA, V. N., JENSSENA, R., ROVERSO, D., “Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning”, *Electrical Power and Energy Systems*, pp. 107–120, 2018.
- [19] ZHOU, S., CHEN, Y., ZHANG, D., et al., “Classification of Surface Defects on Steel Sheet Using Convolutional Neural Networks”, *Materials and technology*, v. 51, n. 1, pp. 123–131, 2017.
- [20] CHEN, S., YANG, J., QIAN, J., “Recurrent Neural network for Facial Landmark Detection”, *Neurocomputing*, v. 219, pp. 26–38, 2017.
- [21] GOODFELLOW, I. J., J., P.-A., MIRZA, M., et al., “Generative Adversarial Nets”, *arXiv preprint arXiv:1406.2661*, pp. 1–9, 2014.
- [22] MISHKIN, D., SERGIEVKIY, N., MATAS, J., “Systematic Evaluation of Convolution Neural Network Advances on the Imagenet”, *Neurocomputing. Computer Vision and Image Understanding*, v. 161, pp. 11–19, 2017.
- [23] LAHOUD, J., GHANEM, B., POLLEFEYS, M., et al., “3D Instance Segmentation via Multi-task Metric Learning”, , pp. 1–112019.
- [24] GENG, H., ZHANG, H., XUE, Y. ZHOU, M., et al., “Semantic Image Segmentation with Fused CNN Features”, *Optoelectronics Letters*, v. 13, n. 5, pp. 381–385, 2017.
- [25] XU, Y., LI, Y., WANG Y.; LIU, M., et al., “Instance Segmentation Using Deep Multichannel Neural Networks”, *IEEE Transactions*, v. 64, n. 12, pp. 2901–2911, 2017.
- [26] REN, S., K., H., J., G. R. S., “Towards Real-Time Object Detection with Region Proposal Networks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 6, pp. 1137–1149, 2017.

- [27] KANG, K., H., L., ET AL, “T-CNN: Tubelets with Convolutional Neural Networks for Object Detection from Videos”, *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–11, 2017.
- [28] SCHWARTING, W., MORA, J. A., , et al., “Planning and Decision-Making for Autonomous Vehicles”, *Annual Review of Control, Robotics, and Autonomous Systems*, pp. 1:8.1—8.24, 2018.
- [29] SHIN, D., KIM, I., “Deep Image Understanding Using Multilayered Contexts”, *Mathematical Problems in Engineering*, pp. 1–11, 2018.
- [30] TANGA, P., WANGA, H., KWONGD, S., “G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition”, *Neurocomputing*, pp. 188–197, 2017.
- [31] PRATES, R. M., SIMAS FILHO, E. F. D., CERQUEIRA, J. J. F., et al., “Desenvolvimento de metodologia inteligente para classificação de tipos de isoladores em redes de distribuição”, *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBRT*, v. XXXV, pp. 398–402, 2017.
- [32] PRATES, M. P., SIMAS FILHO, E. F., MAROTTA, A. P., et al., “Development of Statistical Model for Insulator Type Classification in Overhead Power Lines”. In: *IX Conferencia Nacional em Comunicações, Redes e Segurança da Informação - ENCOM*, pp. 91–92, 2019.
- [33] PRATES, R. M., CRUZ, R., MAROTTA, A. P., et al., “Insulator Visual Non-Conformity Detection in Overhead Power Distribution Lines using Deep Learning”, *Computers & Electrical Engineering, Elsevier*, v. 78, pp. 343–355, 2019.
- [34] PRATES, R. M., RAMOS, R. P., SIMAS, E. F., et al., “Power Distribution Insulators Classification Using Image Hybrid Deep Learning”. pp. 1—5, 2019.
- [35] CRUZ, R., PRATES, R. M., SIMAS FILHO, E. F., et al., “Background Invariance by Adversarial Learning”. In: *25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021.

- [36] BRAZILIAN ASSOCIATION OF TECHNICAL STANDARDS, *NBR 5032 - Insulators for overhead lines with a nominal voltage above 1000 V Ceramic or glass insulator units for a.c. systems*, Tech. rep., ABNT NBR, 2004.
- [37] NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION, *C29.5 - American National Standard for Wet-Process Porcelain Insulators - Low- and Medium-Voltage Types*, Tech. rep., ANSI/NEMA, 2015.
- [38] DE NORMAS TÉCNICAS, A. B., *NBR 15688 - Redes de Distribuição Aérea de Energia Elétrica com Condutores Nus*, Tech. rep., ABNT NBR, 2013.
- [39] YUNQIAN, M., GUO, G., *Support Vector Machines Applications*. v. 1. Springer, 2014.
- [40] LIAW, A., WIENER, M., “Classification and Regression by randomForest”, , pp. 18—222002.
- [41] MITCHELL, T. M., *Machine Learning*. v. 1. McGraw-Hill, 1997.
- [42] GALES, M., YOUNG, S., “A branch and bound algorithm for computing k-nearest neighbors”, v. 1, pp. 195–304, 2008.
- [43] DENG, X., LIUB, Q., DENG, Y., et al., “An improved method to construct basic probability assignment based on the confusion matrix for classification problem”, *Information Sciences*, 2016.
- [44] FAWCETT, T., “An introduction to ROC analysis”, *Pattern Recognition Letters*, v. 27, pp. 861–874, 2006.
- [45] KIESSLING, E., NEFZGER, P., NOLASCO, J., et al., *Overhead Power Lines: Planning, Design, Construction*. v. 1. Springer, 2003.
- [46] KERSTING, W. H., *Distribution System Modeling and Analysis*. v. 4. CRC Press, 2018.
- [47] VR01.02-01.001, “Elaboração de Projeto de Rede de Distribuição Aérea Urbana”, *Grupo NeoEnergia*, pp. 1–45, 2008.

- [48] VR01.02-01.007, “Montagem de Rede Urbana de Distribuição em Média Tensão com Condutores Nus”, *Grupo NeoEnergia*, pp. 1–45, 2009.
- [49] BRAZILIAN ASSOCIATION OF TECHNICAL STANDARDS, *NBR 15992 - Overhead distribution lines of electrical energy using cables covered with polymeric material in spacers for voltages up to 36.2 kV*, Tech. rep., ABNT NBR, 2011.
- [50] SHOEMAKER, T. M., MACK, J. E., *The Linesman’s and Cableman’s Handbook*. v. 13. MC Graw Hill Education, 2017.
- [51] SR01.24-00.002, “Inspeção em Redes de Distribuição Aéreas de 15 kV e 34,5 kV”, *Grupo NeoEnergia*, pp. 1–11, 2010.
- [52] PORTELLA, K. F., AT ALL, “Efeitos Da Poluição Atmosférica (Litorânea E Industrial) Em Isoladores Da Rede Elétrica Da Região Metropolitana De Salvador”, *Química Nova*, v. 31, pp. 340–348, 2008.
- [53] MAROTTA, A. P., SIMAS FILHO, E. F., PRATES, M. P., et al., “Fusão entre Técnicas de Extração de Atributos na Classificação de Isoladores de Redes de Distribuição Usando Redes Neurais”. In: *Simpósio Brasileiro de Automação Inteligente - SBAI*, pp. 2350–2353, 2019.
- [54] REDDY, M. J. B., CHANDRA, K. B., MOHANTA, D. K., “Condition Monitoring of 11 kV Distribution System Insulators Incorporating Complex Imagery Using Combined DOST-SVM Approach”, *IEEE Transactions on Dielectrics and Electrical Insulation*, pp. 664–674, 2013.
- [55] ZHAO, Z., XU, G., QI, Y., et al., “Multi-patch deep features for power line insulator status classification from aerial images”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3187–3194, 2016.
- [56] MURTHY, V. S., GUPTA, S., MOHANTA, D., “Digital image processing approach using combined wavelet hidden Markov model for well-being analysis of insulators”, *IET Image Processing*, v. 5, n. 2, pp. 171–183, 2011.

- [57] ZHANG, Y., LIM, C., DAI, Z., et al., “Photonics and Optoelectronics Using Nano-structured Hybrid Perovskite media and their Optical Cavities”, *Physics Reports*, v. 45, pp. 1—118, 2019.
- [58] XUE, T., LIANG, W., LI, Y., et al., “Ultrasensitive detection of miRNA with an antimonene-based surface plasmon resonance sensor”, *NATURE COMMUNICATIONS*, v. 10, n. 28, pp. 1–9, 2019.
- [59] JALIL, B. E. A., “Visible and Infrared Imaging Based Inspection of Power Installation”, *Pattern Recognition and Image Analysis*, v. 29, pp. 35—41, 2019.
- [60] SOLOMON, C., BRECKON, T., *Fundamentals of Digital Image Processing: a Practical Approach with Examples in Matlab*. v. 3. Wiley Blackwell, 2011.
- [61] GONZALES, R. C., WOODS, R. E., *Digital Image Processing*. v. 3. Pearson Prentice Hall, 2010.
- [62] KAY, S., *Fundamentals of Statistical Signal Processing: Detection Theory*. v. 2. Prentice Hall, 1998.
- [63] PIMENTEL, M. A., CLIFTON, D. A., CLIFTON, L., et al., “A review of novelty detection”, v. 99, pp. 215–249, 2014.
- [64] DING, X., LI, Y., BELATRECH, A. E., et al., “An experimental evaluation of novelty detection methods”, v. 135, pp. 313–327, 2014.
- [65] MITCHELL, T. M., *Machine Learning*. v. 1. McGraw-Hill, 1997.
- [66] GUO, Y., LIU, Y., OERLEMANS, A., et al., “Deep learning for visual understanding: A review”, *Neurocomputing*, v. 187, pp. 27–48, 2016.
- [67] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. v. 2. Springer, 2008.
- [68] KUNCHEVA, L. I., *Combining Pattern Classifiers: Methods and Algorithms*. v. 2. John Wiley and Sons, 2004.

- [69] SHANMUGAN, K. S., BREIPOHL, A. M., *Random Signals: Detection, Estimation and Data Analysis*. v. 1. John Wiley and Sons, 1988.
- [70] TREES, H. L. V., BELL, K. L., TIAN, Z., *Detection Estimation and Modulation Theory, Part I: Detection, Estimation, and Filtering Theory*. v. 2. Wiley-Interscience, 2001.
- [71] MURPHY, K. P., *Machine Learning: A Probabilistic Perspective*. v. 1. MIT Press, 2012.
- [72] WOLPERT, D. H., MACREADY, W. G., “No Free Lunch Theorems for Optimization”, *IEEE Transactions on Evolutionary Computation*, pp. 67–82, 1997.
- [73] WOLPERT, D. H., “The Supervised Learning No-Free-Lunch Theorems”, *Soft Computing and Industry*, pp. 1–18, 2002.
- [74] WOLPERT, D. H., MACREADY, W. G., “Coevolutionary Free Lunches”, *IEEE Transactions on Evolutionary Computation*, pp. 67–82, 2005.
- [75] SHALEV-SHWARTZ, S., BEN-DAVID, S., *UNDERSTANDING MACHINE LEARNING: From Theory to Algorithms*. v. 1. Cambridge University Press, 2014.
- [76] BISHOP, C. M., *Pattern Recognition and Machine Learning*. v. 1. Springer, 2006.
- [77] HAYKIN, S., *Neural Networks and Learning Machines*. v. 3. Pearson, 2008.
- [78] WITTEN, I. H., FRANK, E., HALL, M. A., et al., *Data Mining : Practical Machine Learning Tools and Techniques*. v. 4. Morgan Kaufman, 2016.
- [79] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., *Deep learning*. v. 1. MIT Press, 2017.
- [80] LIPTON, Z. C., BERKOWITZ, J., “A Critical Review of Recurrent Neural Networks for Sequence Learning”, *arXiv:1506.00019v4*, 2015.

- [81] RUMELHART, D. E., HINTON, G., WILLIAMS, G., “Learning Representations by Back-Propagation errors”, *Nature*, 1986.
- [82] ZHANG, Y., ZHANG, E., CHEN, W., “Deep neural network for half tone image classification based on sparse auto-encoder”, v. 50, pp. 245–255, 2016.
- [83] ZHU, Q., HE, Z., ZHANG, TAO, Z., et al., “Improving Classification Performance of Softmax Loss Function Based on Scalable Batch-Normalization”, v. 10, pp. 1–8, 2020.
- [84] LECUN, Y., BOTTOU, L., ORR, G. B., et al., “Efficient BackProp”, , pp. 1–441998.
- [85] MOORE, R. C., DENERO, J., “L1 AND L2 REGULARIZATION FOR MULTICLASS HINGE LOSS MODELS”. pp. 1–5, 2011.
- [86] YANG, D., S., H., Y., J., et al., “Diversity-sensitive Conditional Generative Adversarial Networks”. pp. 1–23, 2019.
- [87] SRIVASTAVA, A., VALKOV, L., RUSSELL, C., et al., “VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning”. pp. 1–11, 2017.
- [88] AMARI, S., “Natural Gradient Works Efficiently in Learning”, *Neural Computation*, pp. 251–276, 1998.
- [89] BOTTOU, L., BOUSQUET, O., “The Tradeoffs of Large Scale Learning”, In: PLATT, J. C., KOLLER, D., SINGER, Y., et al. (eds), *Advances in Neural Information Processing Systems 20*, pp. 161–168, Curran Associates, Inc., 2008.
- [90] GUO, Y., LIU, Y., OERLEMANS, A., et al., “Deep learning for visual understanding: A review”, *Neurocomputing*, v. 187, pp. 27–48, 2016.
- [91] DENG, J., DONG, W., SOCHER, R. L., et al., “ImageNet: A large-scale hierarchical image database”. 2009.
- [92] RUDER, S., “An overview of gradient descent optimization algorithms”, , pp. 1–142017.

- [93] LI, M., ZHANG, T., CHEN, Y., et al., “Efficient Mini-batch Training for Stochastic Optimization”. In: *International Conference on Knowledge Discovery and Data Mining - KDD*, pp. 661–670, 2014.
- [94] QIAN, N., (ed), “On the momentum term in gradient descent learning algorithms”, In: QIAN, N. (ed), *Neural Networks*, pp. 145–151, PERGAMON, 1999.
- [95] KINGMA, D. P., BA, J. L., “Adam : A method for stochastic optimization”, *ICLR*, pp. 1–15, 2014.
- [96] FUKUNAGA, K.; NARENDRA, P. M., “A branch and bound algorithm for computing k-nearest neighbors”, v. 100, pp. 750—753, 1975.
- [97] RODRIGUES, L. F., CRUZ, F. C., OLIVEIRA, M. A., et al., “Carburization level identification in industrial HP pipes using ultrasonic evaluation and machine learning”, *Ultrasonics*, v. 94, pp. 145 – 151, 2019.
- [98] VAPNIK, V. N., “An overview of statistical learning theory”, *Neural Networks, IEEE Transactions on*, v. 10, n. 5, pp. 988–999, 1999.
- [99] FUKUCHI, R. K., ESKOFIER, B. M., DUARTE, M., et al., “Support vector machines for detecting age-related changes in running kinematics”, *Journal of Biomechanics*, v. 44, n. 3, pp. 540–542, 2011.
- [100] LOPES, R., AYACHE, A., MAKNI, N., et al., “Prostate cancer characterization on MR images using fractal features”, *Medical Physics*, v. 38, n. 1, pp. 83–95, 2011.
- [101] YIN, Z., HOU, J., “Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes”, *Neurocomputing*, 2015.
- [102] CHERKASSKY, V., MA, Y., “Practical selection of SVM parameters and noise estimation for SVM regression”, *Neural Networks*, pp. 113–126, 2003.
- [103] NITIN, K., MUKESH, M. R., M., L. G., “Evaluating Kernel Effect on Performance of SVM Classification using Satellite Images”, *International Journal of Scientific Engineering Research*, 2016.

- [104] XU, Z. E., WEINGERGER, K., CHAPELLE, O., “Distance Metric Learning for Kernel Machines”, *arXiv preprint arXiv:1208.3422v2*, pp. 1–17, 2013.
- [105] HUSSAIN, M., KANWAL, S. W., ELZAART, A., et al., “A Comparison of SVM Kernel Functions for Breast Cancer Detection”. In: *Eighth International Conference Computer Graphics, Imaging and Visualization*, IEEE, 2011.
- [106] AL-MEJIBLI, I. S., ADB, D. H., ALWAN, J. K., et al., “Performance Evaluation of Kernels in Support Vector Machine”. In: *1st Annual International Conference on Information and Sciences (AICIS)*, IEEE, 2018.
- [107] REPOSITORY, U. M. L., “Iris Data Set”, 2020.
- [108] BREIMAN, L., “Random Forests”, *Machine Learning*, v. 45, n. 1, pp. 5–32, 2001.
- [109] BOBULSKI, J., “Comparison of the effectiveness of 1D and 2D HMM in the pattern recognition”, *Image Processing & Communication*, v. 19, n. 1, pp. 5–12, 2014.
- [110] GHAHRAMANI, Z., “An Introduction to Hidden Markov Models and Bayesian Networks”, *International Journal of Pattern Recognition and Artificial Intelligence*, v. 15, n. 1, pp. 9–42, 2001.
- [111] RABINER, L. R., “A tutorial on hidden Markov models and selected applications in speech recognition”, *Readings in Speech Recognition*, v. 15, n. 1, pp. 267—296, 1990.
- [112] ASADI, N., MIRZAEI, A., HAGHSHENAS, E., “Creating Discriminative Models for Time Series Classification and Clustering by HMM Ensembles”, *IEEE Transactions on Cybernetics*, v. 46, n. 12, pp. 2899—2910, 2016.
- [113] GONZALES, R. C., WOODS, R. E., *Digital Image Processing*. 4th ed. Pearson Education Limited, 2018.
- [114] MARAGOS, P., “Tutorial on Advances in Morphological Image processing and Analysis”. In: *Visual Communications and Image Processing*, v. 707, pp. 64–74, 1986.

- [115] PATTERSON, J., GIBSON, A., *Deep Learning: A Practitioner's Approach*. v. 1. O'Reilly, 2017.
- [116] CARRIO, A., SAMPEDRO, C., RAMOS, A., et al., "A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles", *Journal of Sensors*, pp. 1–13, 2017.
- [117] PIERSON, H., GASHLER, M., "Deep learning in robotics: a review of recent research", *Advanced Robotics*, pp. 1–16, 2017.
- [118] ZHANG, X., , WU, J., "Deep Belief Networks Based Voice Activity Detection", *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, v. 21, n. 4, pp. 697—710, 2013.
- [119] WARING, J., LINDVALL, C., UMETON, R., "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare", *Artificial Intelligence In Medicine*, pp. 1–12, 2020.
- [120] WAN, S., LIANG, Y. A., Y., Z., "Deep convolutional neural networks for diabetic retinopathy detection by image classification", *Computers and Electrical Engineering*, v. 72, pp. 274–282, 2018.
- [121] SOCHER, R., M., G., MANNING, C. D., N. A. Y., "Zero-Shot Learning Through Cross-Modal Transfer", *NIPS Proceedings*, pp. 1—10, 2013.
- [122] GARTNER, "Top Trends on the Gartner Hype Cycle for Artificial Intelligence - 2019", https://www.gartner.com/smarterwithgartner/top-trends-on-the-gartner-hype-cycle-for-artificial-intelligence-2019?utm_campaign=RM_NA_2019_SWG_NL_NL38_IT&utm_medium=email&utm_source=Eloqua&cm_mmc=Eloqua--Email--LM_RM_NA_2019_SWG_NL_NL38_IT--0000, acessado: 10.10.2020.
- [123] GARTNER, "Top Trends on the Gartner Hype Cycle for Artificial Intelligence - 2020", <https://www.gartner.com/smarterwithgartner/2-megatrends-dominate-the-gartner-hype-cycle-for-artificial-intelligence-2020>, acessado: 10.10.2020.

- [124] BINI, S. A., “Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing: what do these terms mean and how will they impact health care”, *The Journal of Arthroplasty*, pp. 1–9, 2018.
- [125] LEARY, D. E., “Gartner’s hype cycle and information system research issues”, *International Journal of Accounting Information Systems*, pp. 240–252, 2008.
- [126] DEDEHAYIR, O., STEINERT, M., “The hype cycle model: A review and future directions”, *Technological Forecasting Social Change*, pp. 1–14, 2016.
- [127] GARCIA, J., NERI, E., “Cognitive Computing: A Brief Survey and Open Research Challenges”. pp. 328–333, 2015.
- [128] ZHANG, Q., YANG, L. T., CHEN, Z. C., et al., “A survey on deep learning for big data”, *Information Fusion*, pp. 1–12, 2018.
- [129] LIU, J., LI, T., XIE, P., et al., “Urban big data fusion based on deep learning: An overview”, *Information Fusion*, pp. 123–133, 2020.
- [130] LI, Y., HUANG, C., DING, L., et al., “Deep learning in bioinformatics: Introduction, application, and perspective in the big data era”, *Methods*, pp. 4–21, 2019.
- [131] CHATTERJEE, A., GUPTA, U., CHINNAKOTLA, M. K., et al., “Understanding emotions in text using deep learning and big data”, *Computers in Human Behavior*, pp. 4–21, 2019.
- [132] CHEN, X., LIN, X., “Big Data Deep Learning: Challenges and Perspectives”, *IEEE Access*, pp. 1–12, 2014.
- [133] WANG, Y., WEI, G., BROOKS, D., “Benchmarking TPU, GPU, and CPU Platforms for Deep Learning”, *arXiv preprint arXiv:1907.10701v4*, pp. 1–13, 2019.
- [134] DAI, W., BERLEANT, D., “Benchmarking Contemporary Deep Learning Hardware and Frameworks:A Survey of Qualitative Metrics”, *arXiv preprint arXiv:1907.03626v4*, pp. 1–8, 2019.

- [135] WANG, Y., WANG, K., DHI, S., et al., “Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training”, *arXiv preprint arXiv:1909.06842v8*, pp. 1–16, 2020.
- [136] SHRESTHA, A., MAHMOOD, A., “Review of Deep Learning Algorithms and Architectures”, *IEEE Access*, v. 7, pp. 53040–53065, 2019.
- [137] GOOGLE, “TensorFlow”, <https://www.tensorflow.org/>, acessado: 01.10.2020.
- [138] FACEBOOK, “PyTorch”, <https://pytorch.org/>, acessado: 01.10.2020.
- [139] NVIDIA, “cuDNN”, <https://developer.nvidia.com/cudnn>, acessado: 17.10.2020.
- [140] AMD, “ROCm”, <https://rocmdocs.amd.com/en/latest/>, acessado: 17.10.2020.
- [141] SOLTANI, S., SAGDUYU, Y. E., HASAN, R., et al., “Real-Time and Embedded Deep Learning on FPGA for RF Signal Classification”, *arXiv preprint arXiv:1910.05765*, pp. 1–6, 2019.
- [142] ZHANG, S., WU, Y., MEN, C., et al., “Research on OpenCL optimization for FPGA deep learning application”, *PLOS ONE*, 2018.
- [143] MOUSOULIOTIS, P. G., PETROU, L. P., “Software-Defined FPGA Accelerator Design for Mobile Deep Learning Applications”, *arXiv preprint arXiv:1902.03192v2*, pp. 1–10, 2019.
- [144] LIN, T.-Y., MAIRE, M., BELONGIE, S., et al., “Microsoft COCO: Common Objects in Context”, *Lecture Notes in Computer Science*, pp. 740—755, 2014.
- [145] LECUN, BENGIO, “Convolutional Networks for Images, Speech and Time-Series”, v. 44, n. 3, pp. 1–14, 1995.
- [146] KHAN, S., RAHMANI, H., SHAH, S. A. A., et al., *A Guide to Convolutional Neural Networks for Computer Vision..* v. 1. MC, 2018.

- [147] A., V., DOULAMIS, N., DOULAMIS, A., et al., “Deep Learning for Computer Vision: A Brief Review”, , pp. 1–132018.
- [148] AGHDAM, H. H., HERAVI, E. J., *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification..* v. 1. Springer, 2017.
- [149] LECUN, Y., BOTTOU, L., BENGIO, Y., et al., “Gradient-Based Learning Applied to Document Recognition”, *PROCEEDINGS OF THE IEEE*, v. 86, pp. 2278–2324, 1998.
- [150] EICKENBERG, M., GRAMFORT, A., VAROQUAUX, G., et al., “Seeing it all: Convolutional network layers map the function of the human visual system”, *Elsevier*, v. 152, pp. 184–194, 2017.
- [151] FELLEMAN, D. J., ESSEN, D. C., “Distributed Hierarchical Processing in the Primate Cerebral Cortex”, *Cerebral Cortex*, pp. 1–47, 1991.
- [152] HUBEL, D. H., WIESEL, T. N., “Receptive Fields of Single Neurones in the Cat’s Striate Cortex”, *The Journal of Physiology*, v. 148, pp. 574–591, 1959.
- [153] HUBEL, D. H., WIESEL, T. N., “Receptive Fields, Binocular Interaction and Functional Architecture in The Cat’s Visual Cortex”, *The Journal of Physiology*, v. 148, pp. 106–154, 1961.
- [154] HUBEL, D. H., WIESEL, T. N., “Receptive Fields and Functional Architecture of Monkey Striate Cortex”, *The Journal of Physiology*, v. 195, pp. 215–243, 1968.
- [155] DIJK, G. V., *Distribution Theory: Convolution, Fourier Transform, and Laplace Transform.* v. 1. Degruyter, 2013.
- [156] HIRSCHMAN, I. I., WIDDER, D., *The Convolution Transform.* v. 1. Dover Publications, 2005.
- [157] ROSEBROCK, A., *Deep Learning for Computer Vision with Python..* v. 1. PYIMAGESEARCH, 2017.

- [158] ZEILER, M., *Hierarchical Convolutional Deep Learning in Computer Vision (Ph.D. thesis)*. v. 1. New York University, 2014.
- [159] BOUREAU, Y., PONCE, J., LECUN, Y. L., “A Theoretical Analysis of Feature Pooling in Visual Recognition”. pp. 1—8, 2010.
- [160] SCHERER, D., MÜLLER, A., BEHNKE, S., “Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition”, , pp. 92—1012010.
- [161] HE, K., ZHANG, X., R. S., SUN, J., “Spatial pyramid pooling in deep convolutional networks for visual recognition”, *Proceedings of the ICLR*, pp. 1—14, 2014.
- [162] ZEILER, M. D., FERGUS, R., “Stochastic Pooling for Regularization of Deep Convolutional Neural Networks”, *Proceedings of the ICLR*, pp. 1—9, 2013.
- [163] OUYANG, W., LUO, P., ZENG, X., “DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection”, *Proceedings of the ICLR*, pp. 1—14, 2014.
- [164] SZEGEDY, C., AT ALL, “Going deeper with convolutions”, *arXiv preprint arXiv:1409.4842v1*, pp. 1–12, 2014.
- [165] RONNEBERGER, O., FISCHER, P., BROX, T., “U-Net: Convolutional Networks for Biomedical Image Segmentation”, 2015.
- [166] DUMOULIN, V., VISIN, F., “A guide to convolution arithmetic for deep learning”, , pp. 1–312018.
- [167] CHOUDHARY, A., RISHI, R., AHLAWAT, S., et al., “Optimal Feed Forward MLPArchitecture for Off-Line Cursive Numeral Recognition”, *NIPS Proceedings*, v. 2, n. 1, pp. 1—7, 2010.
- [168] DUBEY, A. K., JAIN, V., “Comparative Study of Convolution Neural Network’s Relu and Leaky-Relu Activation Functions”, v. 553, pp. 873–880, 2019.

- [169] GAVRILOV, A. D., JORDACHE, A., VASDANI, M., et al., “Preventing Model Overfitting and Underfitting in Convolutional Neural Networks”. pp. 19–28, 2018.
- [170] YANG, Z., YOU, C., STEINHARDT, J., et al., “Rethinking Bias-Variance Trade-off for Generalization of Neural Networks”, *arXiv preprint arXiv:2002.11328v2*, pp. 1–21, 2020.
- [171] SALIMANS, T., GOODFELLOW, I. ZAREMBA, W., CHENG, V., et al., “Improved Techniques for Training GANs”, *arXiv preprint arXiv:1606.03498v1*, pp. 1–10, 2016.
- [172] MIRZA, M., OSINDERO, S., “Conditional Generative Adversarial Nets”, *arXiv preprint arXiv:1411.1784v1*, pp. 1–7, 2014.
- [173] WATT, J., BORHANI, R., KATSAGGELOS, A. K., *Machine Learning Refined: Foundations, Algorithms, and Applications*. v. 2. Cambridge University Press, 2020.
- [174] GUI, J., SUN, Z., WEN, Y., et al., “A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications”, *arXiv preprint arXiv:2001.06937v1*, pp. 1–28, 2015.
- [175] HUANG, L., PAN, W., ZHANG, Y., et al., “Data Augmentation for Deep Learning-Based Radio Modulation Classification”, .
- [176] ROY, P., TENNAKOON, R., CAO, K., et al., “A Novel Hybrid Approach For Severity Assessment Of Diabetic Retinopathy In Colour Fundus Images”, *IEEE*, v. 26, n. 3, pp. 1078–1082, 2017.
- [177] CAO, G., WANG, S., WEI, B., et al., “A Hybrid CNN–RF Method for Electron Microscopy Images Segmentation”, *Biomimetics Biomaterials and Tissue Engineering*, v. 18, pp. 1–6, 2013.
- [178] HINTON, G. E., SRIVASTAVA, N., KRIZHEVSKY, A., et al., “Improving neural networks by preventing co-adaptation of feature detectors”, , pp. 1–18 2012.

- [179] BALDI, P., SADOWSKI, P., “Understanding Dropout”, , pp. 1–92013.
- [180] ISOLA, P., ZHU, J., ZHOU, T., et al., “Image-to-Image Translation with Conditional Adversarial Networks”, *arXiv preprint arXiv:1611.07004v3*, pp. 1–17, 2018.
- [181] SIMONYAN, K., ZISSERMAN, A., “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.
- [182] KAIMING, H., XIANGYU, Z., SHAOQING, R., et al., “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [183] WU, Z., SHEN, C., HENGEL, A., “Wider or Deeper: Revisiting the ResNet Model for Visual Recognition”, *Pattern Recognition*, 2019.
- [184] GARCIA-GARCIA, A., ORTS-ESCOLANO, S., OPREA, S., et al., “A Review on Deep Learning Techniques Applied to Semantic Segmentation”, , pp. 1–232017.
- [185] HE, K., GKIOXARI, G., DOLLAR, P., et al., “Mask R-CNN”, *arXiv preprint arXiv:1703.06870v3*, pp. 1–12, 2018.
- [186] LI, Z., PENG, C., YU, G., et al., “Light-Head R-CNN: In Defense of Two-Stage Object Detector”, , pp. 1–92017.
- [187] CARUANA, R., “Multitask Learning”, v. 8, pp. 41–75, 1997.
- [188] THUNG, K., WEE, C., “A brief review on multi-task learning”, *Multimedia Tools and Applications*, v. 77, pp. 29705–29725, 2018.
- [189] KOKKINOS, IASONAS, “UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6129–6138, 2017.
- [190] TAO, W., LEU, M. C., YIN, Z., “American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion”, v. 76, pp. 202—213, 2018.

- [191] SU, H., MAJI, S., KALOGERAKIS, E., et al., “Multi-view Convolutional Neural Networks for 3D Shape Recognition”, , pp. 1–92015.
- [192] KARRAS, T., LAINE, S., AILA, T., “A Style-Based Generator Architecture for Generative Adversarial Networks”, *arXiv preprint arXiv:1812.04948v3*, pp. 1–12, 2019.
- [193] LI, C., WAND, M., “Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks”, *arXiv preprint arXiv:1604.04382v1*, pp. 1–17, 2016.
- [194] YU, P., SONG, K., LU, J., “Generating Adversarial Examples With Conditional Generative Adversarial Net”, *arXiv preprint arXiv:1903.07282*, pp. 1–6, 2019.
- [195] REED, S., AKATA, Z., MOHAN, S., et al., “Learning What and Where to Draw”. pp. 217–225, 2016.
- [196] HUANG, X., LI, Y., HOPCROFT, J., et al., “Stacked Generative Adversarial Networks”, *arXiv preprint arXiv:1612.04357v4*, pp. 1–10, 2017.
- [197] FREEDMAN, D., ZHANG, T., “Interactive graph cut based segmentation with shape priors”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 755–762, 2005.
- [198] PRITIKA, P., PRAJAPATI, A., MISHRA, S., “Review of Different Inpainting Algorithms”, *International Journal of Computer Applications*, v. 59, n. 18, pp. 30–34, 2012.
- [199] GONZALEZ, M., S., M., MIR, A., et al., “An Iterative Algorithm for Image Inpainting Using Aggregation Functions”. 2017.
- [200] PRITIKA, P., PRAJAPATI, A., MISHRA, S., “An Image Inpainting Technique Based on the Fast Marching Method”, *Journal of Graphics Tools*, v. 9, n. 1, pp. 23–34, 2004.
- [201] OTSU, N., “A Threshold Selection Method from Gray-Level Histograms”, *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, pp. 62–66, 1979.

- [202] LIM, J., *Two-dimensional signal and image processing*. Prentice Hall, 1990.
- [203] JOZWIK, K. M., KRIEGESKIRTE, N., SORRS, K. R., et al., “Deep Convolutional Neural Networks Outperform Feature-Based But Not Categorical Models in Explaining Object Similarity Judgments”, *Frontiers in Psychology*, v. 8, pp. 1–18, 2017.
- [204] NIU, X., SUEN, C. Y., “A novel hybrid CNN–SVM classifier for recognizing handwritten digits”, *Pattern Recognition*, v. 45, pp. 1318–1325, 2012.
- [205] MCCLELLAND, J. L., “Integrating Probabilistic Models Of Perception And Interactive Neural Networks: A Historical And Tutorial Review”, *Frontiers in Psychology*, v. 4, 2013.
- [206] NAIK, A., RAVICHANDER, A., SADEH, N., et al., “Stress Test Evaluation for Natural Language Inference”, *arXiv preprint arXiv:1806.00692v3*, 2018.
- [207] HOLMES, G., KIRKBY, R., PFAHRINGER, B., “Stress-Testing Hoeffding Trees”, *Lecture Notes in Computer Science*, v. 3721, pp. 495–502, 2005.
- [208] AHMAD, K., AT ALL, “Ensemble of Deep Models for Event Recognition”, *ACM Trans. Multimedia Comput. Commun. Appl.*, v. 14, pp. 1–20, 2018.
- [209] GUO, J., GOULD, S., “Deep CNN Ensemble with Data Augmentation for Object Detection”, *arXiv preprint arXiv:1506.07224v1*, 2015.
- [210] SHIN, H., ROTH, H. R., ET AL, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”, *IEEE Transactions on Medical Imaging*, v. 35, pp. 1285–1297, 2016.
- [211] SEIFERT, C., AAMIR, A., BALAGOPALAN, A., et al., “Visualizations of Deep Neural Networks in Computer Vision: A Survey”, *Transparent Data Mining for Big and Small Data. Studies in Big Data*, v. 32, pp. 123–144, 2017.

- [212] ZHOU, B., KHOSLA, A., LAPEDRIZA, A., et al., “Learning Deep Features for Discriminative Localization”. In: *Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.
- [213] GOODFELLOW, I. J., SHELNS, J., SZEGEDY, C., “Explaining and Harnessing Adversarial Examples”, *arXiv preprint arXiv:1412.6572v3*, pp. 1–11, 2015.
- [214] GUAN, Q., HUANG, Y., ZHONG, Z., et al., “Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification”, *arXiv preprint arXiv:1801.09927v1*, pp. 1–10, 2018.
- [215] DENG, L., “The MNIST Database of Handwritten Digit Images for Machine Learning Research”, *IEEE Signal Processing Magazine*, v. 29, pp. 141–142, 2012.
- [216] CASSEL, M., LIMA, F. K., “Evaluating One-Hot Encoding Finite State Machines for SEU Reliability in SRAM-based FPGAs”. In: *Proceedings of the 12th IEEE International On-Line Testing Symposium*, 2006.
- [217] TAN, S., “Neighbor-weighted K-nearest neighbor for unbalanced text corpus”, *Expert Systems with Applications*, v. 28, pp. 667–671, 2005.
- [218] POZZOLO, A. D., CAELEN†, O., JOHNSON, R. A., et al., “Calibrating Probability with Undersampling for Unbalanced Classification”. In: *Symposium Series on Computational Intelligence*, 2015.
- [219] CARTER, J. V., PAN, J., RAI, S. N., et al., “ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves”, *Surgery*, pp. 1638–1645, 2016.
- [220] KOHAVI, R., “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. pp. 1–7, 1995.
- [221] LI, R., MAGBOOL, N. M., HUANG, B., et al., “Constrained multimodal ensemble Kalman filter based on Kullback–Leibler (KL) divergence”, *Química Nova*, v. 70, pp. 16–28, 2019.

- [222] LUCIC, M., KURACH, K., MICHALSKI, M. BOUSQUET, O., et al., “Are GANs Created Equal? A Large-Scale Study”. pp. 1–10, 2018.
- [223] BARRAT, S., SHARMA, R., “A Note on the Inception Score”, *arXiv preprint arXiv:1801.01973v2*, pp. 1–9, 2018.
- [224] HEUSEL, M., RAMSAUER, H., UNTERTHINER, T., et al., “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”, *arXiv preprint arXiv:1706.08500v6*, pp. 1–38, 2018.
- [225] DOWSON, D. C., LANDAU, B. V., “The Fréchet Distance Between Multivariate Normal Distributions”, *Journal of Multivariate Analysis*, v. 12, pp. 450–455, 1982.
- [226] THADA, V., “Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm”, *International Journal of Innovations in Engineering and Technology (IJJET)*, v. 2, pp. 202–205, 2013.
- [227] BULAT, A., TZIMIROPOULOS, G., “Human Pose Estimation via Convolutional Part Heatmap Regression”. In: *Computer Vision – ECCV 2016*, pp. 717–732, 2016.

Apêndice A

Trabalhos Publicados

Seguem abaixo os resumos dos trabalhos científicos publicados com os resultados encontrados da pesquisa, que fundamentou este documento. Os trabalhos estão em ordem cronológica, iniciando pelo mais antigo.

A.1 Artigos aprovados em congressos

1. *Ricardo M. Prates, Eduardo F. de Simas Filho, Jês de Jesus Fiais Cerqueira e Rodrigo Pereira Ramos* **Desenvolvimento de Metodologia Inteligente para Classificação de Tipos de Isoladores em Redes de Distribuição**. *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBRT*. São Pedro - SP, Setembro de 2017.

Resumo:

O presente trabalho apresenta uma metodologia para classificação automatizada de imagens de três tipos de isoladores existentes em Redes de Distribuição de média tensão, conhecidos coloquialmente como Isolador de Pino, Isolador Polimérico e Isolador de Saia Baiana. O processo de classificação ocorre através do emprego de técnicas de processamento digital de imagens (PDI) e de inteligência computacional. Esta metodologia pode ser caracterizada pelas etapas de segmentação de imagens, extração de atributos dimensionais e cálculo de parâmetros característicos, além do desenvolvimento e implementação de uma Rede Neural Artificial (RNA). Em suma, as informações obtidas na etapa de PDI foram posteriormente tratadas para

o treinamento da RNA escolhida - uma Rede Perceptron Multicamadas. Ao final do artigo, foi realizado um estudo comparativo para identificar um número otimizado de neurônios que devem pertencer à camada oculta da Rede Neural, como também uma avaliação dos parâmetros de desempenho da RNA para o processo de classificação. Como resultado, o sistema mostrou-se satisfatório, obtendo precisão geral de 99% de acerto na identificação do tipo de componente presente na imagem.

2. *Andre P. Marotta, Eduardo F. de Simas Filho, Ricardo M. Prates, e Rodrigo P. Ramos* **Estudo da Influencia do Ângulo de Inclinação na Classificação de Isoladores Utilizando Redes Neurais e Processamento de Imagem.** *VIII Conferencia Nacional em Comunicações, Redes e Segurança da Informação, EN-COM 2018.* Salvador - BA, Outubro de 2018.

Resumo:

O presente trabalho apresenta uma metodologia para aquisição, processamento e classificação automatizada de imagens de cinco tipos de isoladores utilizados em redes de distribuição de média tensão simulando o que ocorre na prática com a aquisição de imagens usando um drone. Um banco de dados foi montado com diferentes ângulos de inclinação em relação ao eixo central dos isoladores. Foram utilizadas diferentes formas de extração de atributos das imagens para alimentar classificadores baseados em redes neurais artificiais. Por fim, foram realizados testes com a compressão por Análise de Componentes Principais (ACP) e Wavelet a fim de verificar a informação mínima necessária para classificação.

3. *Ricardo M. Prates, Jaime S. Cardoso, Eduardo F. Simas, Rodrigo P. Ramos* **Power Distribution Insulators Classification Using Image Hybrid Deep Learning.** *27th European Signal Processing Conference, EUSIPCO 2019.* Coruna - Espanha, Setembro de 2019 - QUALIS CONFERÊNCIAS - CIÊNCIA DA COMPUTAÇÃO: A2. DOI:10.23919/EUSIPCO.2019.8903139

Resumo:

The Overhead Power Distribution Lines present a wide range of insulator components, which have different shapes and types of building materials. These components are usually exposed to weather and operational conditions that may cause deviations in their shapes, colors or textures. These changes might hinder the development of automatic systems for visual inspection. In this perspective, the following work presents a robust methodology for image classification, which aims at the efficient distribution insulator class identification, regardless of its degradation level. This work can be characterized by the following steps: implementation of Convolutional Neural Network (CNN); transfer learning; attribute vector acquisition and design of hybrid classifier architectures to improve the discrimination efficiency. In summary, a previously trained CNN passes through a fine tuning stage for later use as a feature extractor for the training of a new set of classifiers. A comparative study was conducted to identify which classifier architecture obtained the best discrimination performance for non-conforming components. The proposed methodology showed a significant improvement in classification performance, obtaining 95% overall accuracy in identifying the classes of non-conforming components.

4. *Ricardo M. Prates, Eduardo F. de Simas Filho, Andre P. Marotta e Rodrigo P. Ramos* **Development of Statistical Model for Insulator Type Classification in Overhead Power Lines.** *IX Conferencia Nacional em Comunicações, Redes e Segurança da Informação, ENCOM 2019.* Petrolina - PE, Outubro de 2019.

Resumo:

This paper presents a statistical methodology for image classification of insulator types usually found in Overhead Power Distribution Lines (OPDLs). The classification process occurs through the use of deep learning and a statistical approach, known as the Hidden Markov Model (HMM). This methodology can be described by the following steps: image segmentation by a specific type of Convolutional Neural Network (CNN), called U-Net; dimensional attributes extraction; parameter vector design and development of an HMM chain for sequential data classification. The proposed model achieved 91% overall accuracy in insulator class identification, surpassing other available solutions, such as Random Forest and Support Vector

Machine (SVM).

5. *Andre P. Marotta, Eduardo F. de Simas Filho Ricardo M. Prates e Rodrigo P. Ramos* **Fusão Entre Técnicas de Extração De Atributos na Classificação de Isoladores de Redes de Distribuição Usando Redes Neurais.** *Simpósio Brasileiro de Automação Inteligente - SBAI 2019*). Ouro Preto - MG, Outubro de 2019.

Resumo:

O presente trabalho apresenta uma metodologia para processamento automatizado de imagens e classificação de quatro tipos de isoladores utilizados em redes de distribuição aéreas de média tensão (RDAs). Para isso, foi desenvolvida uma rede de distribuição didática para coletar fotos dos isoladores em um ambiente externo e também um estúdio para aquisição de imagens em ambiente controlado. Posteriormente, os atributos de imagem foram extraídos e utilizados como entrada para diferentes classificadores treinados, como redes neurais, máquina de vetor de suporte, árvore de decisão, n-vizinhos próximos, naive bayes e modelo oculto de Markov, para uma mistura gradual entre as bases de dados (estúdio e imagens externas). As eficiências de classificação foram comparadas considerando a aplicação da análise de componentes principais (ACP) para identificar as informações mínimas necessárias para a classificação.

6. *Ricardo Cruz, Ricardo M. Prates, Eduardo F. Simas Filho, Joaquim F. Pinto Costa e Jaime S. Cardoso* **Background Invariance by Adversarial Learning.** *International Conference on Pattern Recognition - ICPR 2020*). Itália, Janeiro de 2021 (Artigo aprovado para apresentação). - QUALIS CONFERÊNCIAS - CIÊNCIA DA COMPUTAÇÃO: A2.

Resumo:

Convolutional neural networks are shown to be vulnerable to changes in the background. The proposed method is an end-to-end method that augments the training

set by introducing new backgrounds during the training process. These backgrounds are created by a generative network that is trained as an adversary to the model. A case study is explored based on overhead power line insulators detection using a drone – a training set is prepared from photographs taken inside a laboratory and then evaluated using photographs that are harder to collect from outside the laboratory. The proposed method improves performance by over 20% for this case study

A.2 Artigos publicados em revista

7. Ricardo M. Prates, Ricardo Cruz, Andre P. Marotta, Rodrigo P. Ramos, Eduardo F. Simas, Jaime S. Cardoso **Insulator Visual Non-Conformity Detection in Overhead Power Distribution Lines using Deep Learning.** *Computers & Electrical Engineering, Elsevier* - Qualis A2 para Engenharias IV. **Submetido no dia 28/04/2019, Aceito no dia 01/08/2019.**

DOI:10.1016/j.compeleceng.2019.08.001

Resumo:

Overhead Power Distribution Lines (OPDLs) correspond to a large percentage of the medium-voltage electrical systems. In these networks, visual inspection activities are usually performed without resorting to automated systems, requiring a significant investment of time and human resources. We present a methodology to identify the defect and type of insulators using Convolutional Neural Networks (CNNs). More than 2,500 photographs were collected both from inside a studio and from a realistic OPDL. A classification model is proposed to automatically recognize the insulators conformity. This model is able to learn from indoors photographs by augmenting these images with realistic details such as top ties and real-world backgrounds. Furthermore, Multi-Task Learning (MTL) was used to improve performance of defect detection by also predicting the insulator class. The proposed methodology is able to achieve an accuracy of 92% for material classification and 85% for defect detection, with F1-score of 0.75, surpassing available solutions.

8. *Rodrigo P Ramos, Jéssica S Gomes, Ricardo M Prates, Eduardo F Simas Filho, Barbara J Teruel e Daniel dos Santos Costa* **Non-invasive setup for Grape Maturation Classification Using Deep Learning.** *Computers & Electrical Engineering, Elsevier* - Qualis A1 para CIÊNCIAS AGRÁRIAS I e Qualis A2 para INTERDISCIPLINAR. **Submetido no dia 06/05/2020, Aceito no dia 08/09/2020.** DOI 10.1002/jsfa.10824

Resumo:

Background: The San Francisco Valley region from Brazil is known worldwide for its fruit production and exportation, especially grapes and wines. The grapes have high quality not only due to the excellent morphological characteristics, but also to the pleasant taste of their fruits. Such features are obtained because of the climatic conditions present in the region. In addition to the favorable climate for grape cultivation, harvesting at the right time interferes with fruit properties.

Results: This work aims to define grape maturation stage of Syrah and Cabernet Sauvignon cultivars with the aid of deep learning models. The idea of working with these algorithms came from the fact that the techniques commonly used to find the ideal harvesting point are invasive, expensive, and take a long time to get their results. In this work, convolutional neural networks were used in an image classification system, in which grape images were acquired, preprocessed, and classified based on their maturation stage. Images were acquired with varying illuminants that were considered as parameters of the classification models, as well as the different post-harvesting weeks. The best models achieved maturation classification accuracy of 93.41% and 72.66% for Syrah and Cabernet Sauvignon respectively.

Conclusions: It was possible to correctly classify wine grapes using computational intelligent algorithms with high accuracy, regarding the harvesting time, corroborating chemometric results.

Apêndice B

Outras Informações Relevantes

B.1 Estágio em Instituição de Ensino Estrangeira

1. *Programa de Doutorado Sanduíche no Exterior – PDSE - PLANO DE ESTUDOS - EDITAL CAPES Nº 47 /2017 Desenvolvimento de metodologia inteligente para classificação e segmentação instanciada de imagens de Redes de Distribuição de Energia Elétrica. Universidade do Porto - INESC TEC / FEUP.* Coorientador estrangeiro: Prof. Dr. Jaime dos Santos Cardoso; **Período de vigência:** 6 meses - set/2018 até mar/2019.

B.2 Desenvolvimento de *dataset*

2. *Base de dados de imagens de componentes isoladores de distribuição - Link:*

http://www.dee.eng.ufba.br/dslab/index.php/opdl_dataset/.

Citado no artigo:.

Ricardo M. Prates, Ricardo Cruz, André P. Marotta, Rodrigo P. Ramos, Eduardo F. Simas Filho, Jaime S. Cardoso. **Insulator visual non-conformity detection in overhead power distribution lines using deep learning.** Computers and Electrical Engineering, ELSEVIER, 78 (2019) 343–355.

DOI: <https://doi.org/10.1016/j.compeleceng.2019.08.001>