

Detecção de Infração em faixa de pedestres sem semáforos utilizando visão computacional e redes neurais

Bruno Guimarães Costa Alves

Dissertação de mestrado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como parte dos requisitos para obtenção do grau de Mestre em Ciências no domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Antonio Cezar de Castro Lima, Ph.D.

Orientador

Salvador, Bahia, Brasil

©Bruno Guimarães Costa Alves, Novembro de 2007

Detecção de Infração em faixa de pedestres sem semáforos utilizando visão computacional e redes neurais

Bruno Guimarães Costa Alves

Dissertação de mestrado

Antonio Cezar de Castro Lima, Ph.D.

Orientador

Niraldo Roberto Ferreira Neto, Dr, UFBA

Componente da Banca

George Darmiton da Cunha Cavalcanti, Dr, UFPE

Componente da Banca

Salvador, Bahia, Brasil, Novembro de 2007

Agradecimentos

A primeira pessoa a quem devo agradecer é a minha querida mãe. Sem ela jamais estaria aqui. Meus amigos, Marcos Cícero, João Paulo e Ricardo Martinho, que tanto me influenciaram em sempre questionar tudo, não ter dogmas e estar ciente de que tudo muda, que o conhecimento é algo que é construído com o estudo e a dedicação.

Meu irmão Tássio, uma pessoa maravilhosa e que também tem sua contribuição para que chega-se aqui onde estou. São tantas as pessoas a agradecer. Meu tio e minha tia, que me acolheram na sua casa enquanto ainda fazia graduação. A Heberni, que me estimulou a gostar desta área. Ao meu professor Colbert São Paulo, que me ajudou a encontrar a engenharia elétrica e saber que este era o curso que tanto queria para mim.

A minha mulher, Camile, que tanto me ajudou na correção desta dissertação, que tanto teve paciência comigo durante as noites e os fins de semana no computador.

Sei que ainda deixo de citar muitos nomes aqui, mas agradeço a todas as pessoas que passaram em minha vida e contribuíram para minha formação como pessoa e profissional de alguma forma, pois tudo nos influencia, todas as pessoas, situações, imagens e sons.

Resumo

Neste trabalho foram empregadas Redes Neurais Artificiais (RNA) e técnicas de processamento de imagens com o objetivo de identificar corretamente infrações em faixas de pedestres sem semáforos, cometida por condutores de veículos que desrespeitam a parada obrigatória na existência de pessoas que desejem atravessá-las. Com este fim, foram aplicadas para o reconhecimento dos padrões, veículos e pessoas, técnicas básicas de detecção de bordas em conjunto com uma solução específica desenvolvida e apresentada neste trabalho. A combinação dessas técnicas melhorou o processo de detecção de contornos dos objetos, facilitando assim o reconhecimento de veículos na faixa de pedestres por padrões geométricos, através do fornecimento de apenas informações relevantes à RNA durante o processo de aprendizado dos padrões. A utilização de redes neurais para identificação de pessoas na faixa veio da grande dificuldade encontrada com utilização de outros métodos, da sua excelente capacidade de reconhecer padrões em tempo real, mesmo quando o espaço de pontos a ser utilizado é grande. Os resultados aqui apresentados demonstram a capacidade do algoritmo desenvolvido em reconhecer infrações com taxas de acerto superiores a 78%.

Abstract

In this work, Artificial Neural Networks (ANN) and image processing techniques have been applied to identify drivers who violate the law when they fail to stop their vehicles before a pedestrian crosswalk without traffic lights. Standards pattern recognition approaches based on edge detection tools, together with a specific solution provided here in this work, have been used to identify people and vehicles on the crosswalk. The combination of the standard and specific approaches improved significantly the edge detection of the objected involved, that allowed an easy recognition of geometric patterns related to the vehicles on the pedestrian crossing, for example, their wheels. The ANNs have been mainly employed in order to recognize the presence of people on the crosswalk due to several difficulties which cannot be overcome using traditional methods. The results presented here have shown the capability of the proposed algorithm to identify infraction correctly related to pedestrian crosswalk in 78% of the cases tested.

Conteúdo

1	Introdução	3
1.1	Detecção de infração em faixa de pedestre	3
1.2	Detecção de veículos na faixa de pedestre	4
1.3	Identificação de pessoas na faixa	4
1.4	Organização da Dissertação	5
2	Metodologia e Materiais Utilizados	7
2.1	Metodologia	7
2.2	Materiais	9
3	Ferramentas Matemáticas de Processamento de Imagem	11
3.1	Processamento Digital de Imagem	11
3.2	Relações entre os pixels	13
3.3	A segmentação da imagem	14
3.3.1	Derivadas em imagem digital	15
3.3.2	Máscaras de operação	16
3.3.3	Detecção de descontinuidades	16
4	Detecção de veículos na faixa de pedestre	19
4.1	A busca por características	19
4.2	A extração de características	20
4.3	Reconhecimento de roda	26
5	Identificação de Pessoas na Faixa de Pedestre	32
5.1	Identificação de Pessoas na Faixa através RNA	32
5.2	Identificação de Pessoas na Faixa através de Perceptron de Múltiplas Camadas .	33
5.3	Identificação de pessoas na faixa através Mapa Auto-Organizável	36

<i>CONTEÚDO</i>	5
6 Resultados	39
6.1 Resultados dos testes experimentais	39
6.2 Análise dos Resultados	40
7 Conclusão e Perspectivas Futuras	43

Lista de Figuras

2.1	Imagem adquirida pelo posicionamento correto da câmara.	8
2.2	Imagem adquirida pelo posicionamento incorreto da câmara.	9
3.1	Sensor tipo Matriz (Gonzalez & E.Woods, 2002).	12
3.2	Processo de digitalização de imagem (Gonzalez & E.Woods, 2002).	12
3.3	Imagem com 256 x 256 pixels. (a) 16 níveis de cinza (b) 8 níveis de cinza (c) 2 níveis de cinza (Gonzalez & E.Woods, 2002).	13
3.4	Diposição espacial de uma imagem digitalizada (Gonzalez & E.Woods, 2002).	14
3.5	Máscara matricial para operação em imagens digitais.	16
3.6	Processo de operação de imagem via máscara.	17
3.7	Máscara que implementa Prewit.	17
3.8	Máscara que implementa Sobel.	18
4.1	Imagem de faixa de pedestre.	21
4.2	Imagem de faixa processada corretamente com o operador de Sobel.	22
4.3	Imagem de faixa de pedestre.	23
4.4	Imagem de faixa processada incorretamente com o operador de Sobel.	24
4.5	Detecção de borda da roda feita corretamente pelo método de Canny.	24
4.6	Detecção de borda da roda feita de forma incorreta pelo método de Canny.	25
4.7	Imagem original.	27
4.8	Imagem resultante do processo de suavização.	27
4.9	Imagem resultante do processo de diferenciação.	28
4.10	Imagem resultante do processo de detecção de bordas.	28
4.11	Imagem sem processamento.	29
4.12	Imagem sendo processada com o algoritmo de identificação de círculo.	30
4.13	Imagem processada pelo algoritmo de identificação de círculo.	30
4.14	Imagem processada pelo algoritmo de detecção de veículo.	31
5.1	Perceptron com uma camada oculta.	33
5.2	Perceptron com duas camadas ocultas.	35

5.3	Perceptron com uma camada oculta contendo 8 neurônios.	36
5.4	Gráfico do número de iterações pelo erro de saída para o perceptron com uma camada oculta contendo oito neurônios.	37
5.5	Arquitetura da rede para o algoritmo SOM.	37

Lista de Símbolos e Abreviaturas

Símbolos

A_0 - area do objeto presente na imagem

C_0 - fator circular

$d_j(n)$ - saída desejada do neurônio j no passo n

DSP - digital signal processing

$e_j^L(n)$ - erro do neurônio j da camada de saída

$f(x, y)$ - função que representa uma imagem analógica

$G(x)$ -função gaussiana

$G'(x)$ - derivada da função gaussiana

G_x - gradiente da imagem digital na direção x

G_y - gradiente da imagem digital na direção y

$I(x, y)$ - função que representa uma imagem digital

$I'(x, y)$ - imagem suavizada com uma função gaussiana

$I''(x, y)$ - imagem diferenciada com a derivada da função gaussiana

JPG - Joint Photographic Experts Group

$MPEG$ - Moving Picture Experts Group

$M(x, y)$ - valor da média de uma região da imagem

$N_4(p)$ - relação entre pixels que tem disposição espacial vertical e horizontal

$N_D(p)$ - relação entre pixels que tem disposição espacial diagonal

$N_8(p)$ - relação entre pixels que tem disposição espacial vertical, horizontal e diagonal

P_0 - perimetro de uma objeto presente em uma imagem digital

S - caminho entre dois pixels

R - região de uma imagem digital

RNA - rede neural artificial

Y_i^{l-1} - resposta do neurônio i presente na camada $l - 1$

$\frac{\partial I}{\partial x}$ - derivada parcial na direção x de uma imagem digital

$\frac{\partial I}{\partial y}$ - derivada parcial na direção y de uma imagem digital

$\frac{\partial^2 I}{\partial x^2}$ - derivada parcial de segunda orden na direção x de uma imagem digital

$\frac{\partial^2 I}{\partial y^2}$ - derivada parcial de segunda ordem na direção y de uma imagem digital

∇I - gradiente de uma imagem digital

∇ - magnitude de uma vetor

∇I^2 - laplaciano de uma imagem digital

η - é a taxa de aprendizagem do algoritmo de retropropagação

$\delta_j^L(n)$ - gradiente do neurônio j da camada de saída L

$\delta_j^l(n)$ - gradiente do neurônio j da camada oculta l

φ_j - função de ativação do neurônio j

φ_j' - derivada da função de ativação do neurônio j

Capítulo 1

Introdução

1.1 Detecção de infração em faixa de pedestre

A criação da faixa de pedestres veio com a finalidade de melhorar o andamento do trânsito de veículos e pedestres, permitindo que estes pudessem existir de forma harmônica. Os artigos 70 e 71 do Código Nacional de Trânsito (Cardoso et al., 1997) dizem que: os pedestres que estiverem atravessando a via sobre as faixas delimitadas para este fim terão a prioridade de passagem, exceto nos locais com sinalização semafórica, onde deverão ser respeitadas as disposições deste Código. Como diz o Código citado acima, existem faixas que tem seu controle de preferência feito por semáforos e outras que não. Nas que não tem esse controle, o pedestre tem a prioridade. Assim com base em garantir o cumprimento da lei e devida punição para aqueles que venham a descumpri-la, nasceu à idéia de desenvolver um algoritmo e um software que realize o processo de detectar a infração dos veículos em faixas que não possuem semáforo, com um tempo de execução máximo de um segundo. O tempo de processamento definido anteriormente teve seu cálculo baseado em sempre permitir que cada veículo que passe pela faixa seja analisado pelo menos uma única vez, já que todas as faixas analisadas por este trabalho tinham como velocidade máxima permitida 20Km/h (quilômetros por hora) e um comprimento médio de 5 metros, o que implica que um veículo com uma velocidade de 20Km/h levaria aproximadamente 1 segundo para percorrer toda a faixa. Assim, este trabalho aqui proposto tem como objetivo o desenvolvimento de um algoritmo que seja capaz de realizar a detecção da infração pedestre e veículos presentes na faixa de pedestre sem semáforo para o seu controle de preferência, implementar este algoritmo através de um programa que seja capaz de abrir um arquivo de vídeo e imagem, realizando a detecção ou não da infração.

Para facilitar a detecção da infração, o processo foi dividido em duas partes distintas. A primeira é a identificação de veículos na faixa de pedestre, e a segunda é quando for detectada a presença de veículo, identificar a existência ou não de pessoas na faixa. Estes são os dois problemas que devem ser resolvidos para que o processo de detecção de infração seja executado

de forma correta.

1.2 Detecção de veículos na faixa de pedestre

A detecção de veículos sobre a faixa de pedestres é o primeiro problema que deve ser resolvido neste trabalho. Na literatura existe uma grande quantidade de trabalhos dedicados à detecção de veículos, sendo a maioria voltada para a finalidade de controle de fluxo de veículos em rodovias, ou seja, controle de tráfego em rodovias. Podem-se citar (A.Silva & Gonzaga, 2006),(Micoski & M.Hemerly, 2006),(F.F.S.Ribeiro & de C. Lima, 1999),(Gepperth et al., 2005) e (Goerick et al., 1996) como alguns dos exemplos de trabalhos nesta área. Todos estes trabalhos mostram diferentes métodos que podem ser utilizados para a identificação de veículos, sendo que todos eles têm em comum o passo mais básico dentro da área de processamento de imagens, a segmentação. No processamento de imagens com finalidade de identificação de padrões, o passo mais básico é extrair os dados de interesse a serem processados. Para isso é necessário realizar a segmentação da imagem. Com a segmentação é possível separar os objetos presentes na imagem, possibilitando assim excluir dados desnecessários e minimizar a quantidade de informações a ser processada. Existem vários métodos para segmentar imagem, sendo um dos mais comuns o de (Canny, 1986).

Uma boa segmentação é fundamental para que todo o restante do processamento funcione de forma correta. Com isso o processo de segmentação se tornou um dos primeiros desafios do processo de detecção de veículo sobre a faixa de pedestres, levando a seguinte questão: qual seria o melhor algoritmo para utilizar no processo de segmentação? Esta não é uma pergunta fácil, já que isso depende muito do problema a ser resolvido, pois não existe em processamento de imagens uma fórmula ou solução pronta que possa ser aplicada a qualquer situação. O outro desafio foi encontrar características que pudessem indicar a presença do padrão veículo na faixa. As características escolhidas influenciam diretamente no processo de segmentação, portanto, a escolha dessas características foi a primeira tarefa a ser resolvida nesta etapa. Com isso a detecção de veículos na faixa consistiu basicamente em buscar ou desenvolver o melhor método para realizar a segmentação da imagem, com o intuito de facilitar a identificação das características que permitem identificar o padrão veículo na faixa.

1.3 Identificação de pessoas na faixa

Uma vez resolvido o problema da detecção de presença de veículo sobre a faixa de pedestres, restava agora verificar a presença de pessoas sobre a faixa juntamente com os veículos. Este problema é bem mais complexo que o problema de identificação de carros. Existem alguns trabalhos na área de identificação de pessoas, tais como (Shashua et al., 2004), (Oren et al.,

1997), (Tabb et al., 1999) e (Zhao & Nevatia, 2003), mas nenhum trabalho falando sobre a identificação de pessoas com a presença de carros sobre a faixa de pedestres foi encontrado.

Como já foi dito na seção 1.1 deste capítulo, a solução para o problema da detecção de infração deve ter um tempo máximo de processamento inferior a um segundo. Logo, a solução proposta para o problema deste item deve atender a este critério. Com isso a utilização da imagem que é gerada através do processo de segmentação para a detecção de veículo deve ser utilizada para o processo de identificação da presença ou não de pedestres sobre a faixa. Isso evita que um novo processo de segmentação ou outro processo qualquer de extração de características seja realizado, diminuindo assim o custo computacional empregado. Em (Shashua et al., 2004), (Oren et al., 1997) e (Tabb et al., 1999) soluções interessantes são propostas. No entanto, nenhuma delas foi testada para este ambiente específico, onde a variação de formas devido à presença de carros dificulta muito a retirada do objeto pedestre da imagem. Além disso, todos têm um tempo de processamento superior ao estabelecido na seção 1.1. No entanto (Tabb et al., 1999) utiliza redes neurais para a classificação de formas humanas e não humanas em conjunto com modelos de Contorno Ativo (Kass et al., 1988). Deste trabalho nasceu a idéia de utilizar redes neurais para identificar a presença de pessoas sobre a faixa ou não, uma vez que a detecção da presença de veículo tenha sido feita.

A RNA é uma ferramenta muito eficiente, que já vem sendo bastante utilizada para a resolução de uma grande quantidade de problemas na área de reconhecimento de padrões (Haykin, 2001). Com os testes realizados neste trabalho de mestrado, a RNA se mostrou ser uma ferramenta bem interessante para a resolução do problema e com um custo computacional aceitável para manter a solução deste problema dentro do critério estabelecido na seção 1.1. Assim a identificação de pessoas na faixa consistiu em encontrar uma estrutura de rede neural que fosse mais adequada para essa tarefa e um método de treinamento que convergisse de maneira rápida e satisfatória.

1.4 Organização da Dissertação

Esta dissertação foi dividida em sete capítulos, contando com este. No capítulo 2, Metodologia e Materiais, é explicada toda a metodologia e os materiais utilizados para o desenvolvimento do algoritmo, tanto em nível de hardware como de software. No capítulo 3, Ferramentas Matemáticas de Processamento de Imagem, é apresentada a base matemática necessária para o desenvolvimento do algoritmo aqui proposto e o processamento de imagens digitais. No capítulo 4, Detecção de veículos na faixa de pedestre, são explicados todos os passos, as dificuldades e o algoritmo desenvolvido para a detecção do padrão veículo. No capítulo 5, Identificação de pessoas na faixa de pedestres, são mostrados todos os passos, dificuldades, algoritmo desenvolvido para solucionar este problema e o algoritmo final que realiza a detecção da infração, pedestre(s)

e veículo(s), sobre a faixa de pedestres. No capítulo 6, Resultados, são mostrados os resultados obtidos com os algoritmos propostos no capítulo 4 e 5 e uma análise dos mesmos. No capítulo 7, Conclusão e Perspectivas Futuras, são apresentadas as contribuições e perspectivas futuras.

Capítulo 2

Metodologia e Materiais Utilizados

2.1 Metodologia

Uma vez que a idéia de detecção de infração em faixa de pedestres foi proposta, o primeiro passo a ser tomado foi verificar a existência de trabalhos similares ou iguais a este aqui proposto. Foi feita uma extensa busca na Internet, pesquisando-se artigos, dissertações, teses, livros e empresas ou instituições que já tivessem desenvolvido uma solução para este problema. Vários artigos em revistas especializadas foram vistos e mesmo assim não se encontrou nada que pudesse ser similar a idéia aqui proposta, detecção de pessoas e veículos sobre a faixa de pedestres. Findado este passo, foi realizado um estudo sobre processamento de imagem e reconhecimento de padrões, buscando-se ter embasamento teórico para o desenvolvimento deste trabalho. Na etapa seguinte, foi realizada a escolha das ferramentas e materiais necessário para o desenvolvimento da dissertação. Decidiu-se fazer primeiramente a implementação e os testes do algoritmo em um ambiente de programação e simulação com mais recursos, para depois implementar cada função em linguagem de programação C, gerando uma biblioteca que permitisse desenvolver o programa final que executaria o algoritmo aqui desenvolvido.

Como o objetivo deste trabalho é reconhecer os padrões veículos e pessoas, este foi dividido em duas etapas. Na primeira etapa o programa que seria desenvolvido deveria identificar o padrão veículos, e uma vez que este padrão fosse identificado, a segunda etapa seria então realizar o reconhecimento do padrão pessoas. Esta separação facilita o processo de desenvolvimento do algoritmo, pois facilita a detecção de erros nas funções implementadas e os testes das idéias propostas. Cada uma dessas etapas foi dividida em sub tópicos que permitiu a resolução do problema de forma mais rápida.

Primeira etapa:

- Escolha de características a serem extraídas;
- Escolha e desenvolvimento do algoritmo de segmentação;
- Implementação do algoritmo de segmentação;

- Classificação da presença de veículo sobre a faixa.

Segunda etapa:

- Escolha da arquitetura da RNA e do algoritmo de treinamento;
- Teste da arquitetura;
- Teste de novas arquiteturas.

Os dados foram adquiridos por meio de uma filmagem, gerando assim vários arquivos de vídeo. Para a aquisição destes dados se utilizou uma câmera com alta resolução, 7.2Megapixels, configurando esta para gerar um arquivo MPEG com resolução de 240 por 320 pixels. No entanto, para que a infração fosse detectada de forma correta é necessário posicionar a câmera de uma maneira que apenas a(s) pessoa(s) e veículo(s) que estivessem sobre a faixa fossem filmadas. Então o primeiro passo para realizar a filmagem foi posicionar a câmera em uma altura que permitisse que a mesma enquadrasse apenas a faixa de pedestres. Na figura 2.1 é possível ver a imagem adequada adquirida pela câmera posicionada de forma correta, enquanto que na figura 2.2 é possível ver uma imagem considerada não adequada para o processamento, devido ao posicionamento incorreto da câmera.

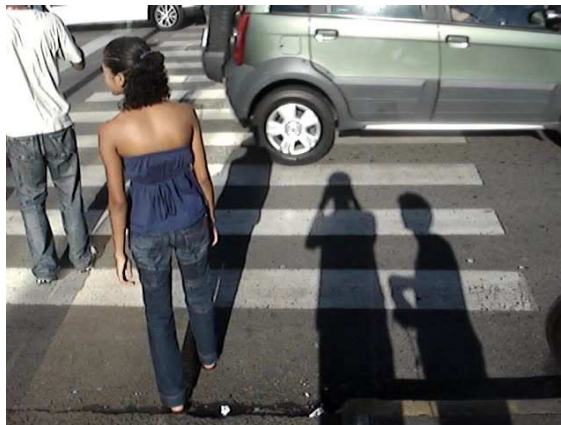


Figura 2.1: Imagem adquirida pelo posicionamento correto da câmera.

Além de posicionar a câmera em uma determinada altura, também foi decidido que a câmera sempre ficaria perpendicular à faixa, como mostrado na figura 2.1 e 2.2, devido ao tipo de característica escolhida para identificar o padrão veículo, a roda.

Foram escolhidas três faixa de pedestres, que foram filmadas em diferentes horários do dia. A escolha de apenas três faixas foi motivada pelo fato destas terem um alto índice de infração quando comparadas com qualquer outra faixa existente na cidade de Salvador. Por alto índice de infração deve-se entender a ocorrência de 10 infrações por minuto ou mais, enquanto que nas outras faixas ocorrem em média 1 infração por hora. As condições climáticas durante o processo de filmagem variaram desde céu limpo e claro a um céu nublado, sendo que nenhuma



Figura 2.2: Imagem adquirida pelo posicionamento incorreto da câmera.

das filmagens foi feita com a presença de chuva. Já a realização das filmagens em diferentes horários do dia teve a intenção de testar insensibilidade do algoritmo com relação às variações temporais e climáticas. Assim, foram adquiridos três arquivos em formato MPEG que foram separados em três pastas, foram chamadas de pasta 1, pasta 2 e pasta 3. Cada pasta se refere a uma determinada faixa. Com os arquivos em mão, decidiu-se transformar todos os seus quadros em imagens no formato JPG. Essas imagens são mais fáceis de trabalhar que um vídeo, pois isto facilita muito mais o trabalho de desenvolvimento e correção do código que implementa o algoritmo. Para cada arquivo de vídeo gravado gerou-se uma pasta contendo 5200 imagens no formato JPG.

2.2 Materiais

Para o desenvolvimento do algoritmo e criação do software que o implementa, os seguintes recursos de hardware e software foram utilizados.

- Matlab 7.1;
- C++ Builder 5;
- Câmera digital SonyCyber-shotDSC-W55;
- Breeze 5.4.1 Ligth;
- Computador com processador Celeron 1.3GHz e memória RAM de 512MB.

A idéia básica foi começar os estudos utilizando o MATLAB, já que este tem recursos de visualização e acesso a dados prontos, o que facilita toda a implementação e processo de correção. Assim foi desenvolvido um código no MATLAB e à medida que este código era testado e aprovado, este era implementado em C utilizando o C++ Builder 5. Todo o desenvolvimento do algoritmo e do software que o implementa foi feito deste jeito. O código no MATLAB

trabalhou apenas com figuras no formato JPG, enquanto que o software desenvolvido no Builder trabalhou com um arquivo de vídeo no formato MPG e imagens no formato JPG.

Capítulo 3

Ferramentas Matemáticas de Processamento de Imagem

3.1 Processamento Digital de Imagem

O processamento digital de imagens refere-se ao processamento de imagens em um sistema de hardware digital (Gonzalez & E.Woods, 2002). Podem-se citar como exemplos o computador e plataformas dedicados com DSP. No entanto, para que a imagem possa ser manipulada por essas plataformas, as imagens devem ser digitalizadas. O processo de digitalização é a transformação de uma grandeza que é contínua em discreta. Pode-se então dizer que este processo tem seu início com a utilização de um sensor de imagem, que tem a capacidade de transformar a energia luminosa em um sinal de tensão (Gonzalez & E.Woods, 2002). Os sensores de imagem utilizados na maioria das câmeras digitais têm o formato de uma matriz bidimensional como é mostrado na figura 3.1. Cada ponto da matriz é um sensor que gera um sinal de tensão proporcional à energia luminosa que incide sobre ele. Uma vez que esse sinal de tensão é gerado, então ele é amostrado e quantizado, formando assim uma imagem digital.

A imagem é um sinal contínuo e pode ser modelada como uma função $f(x,y)$, onde x e y são suas coordenadas espaciais, de amplitude finita e pertencentes ao conjunto dos números reais. Assim o processo de amostragem de uma imagem consiste em tornar discretos os valores de suas coordenadas, fazendo com que elas passem a serem números inteiros. Uma vez feita esta separação, cada um dos valores de tensão $f(x,y)$ é digitalizado, realizando assim o processo de quantização. Uma imagem digital pode então ser definida como uma função $I(x,y)$, onde x e y são suas coordenadas espaciais pertencentes ao conjunto dos números inteiros positivos, de valores e amplitude finita. Na figura 3.2 é ilustrado o processo de digitalização da imagem.

Os elementos que compõem a imagem digital são chamados de pixels. Cada pixel pode ser representado por uma determinada quantidade de bits. No caso do trabalho desenvolvido aqui, cada valor de pixel é representado por 8bits, o que permite que 256 valores diferentes possam

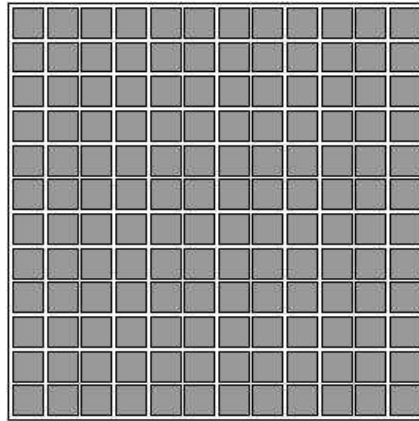


Figura 3.1: Sensor tipo Matriz (Gonzalez & E.Woods, 2002).

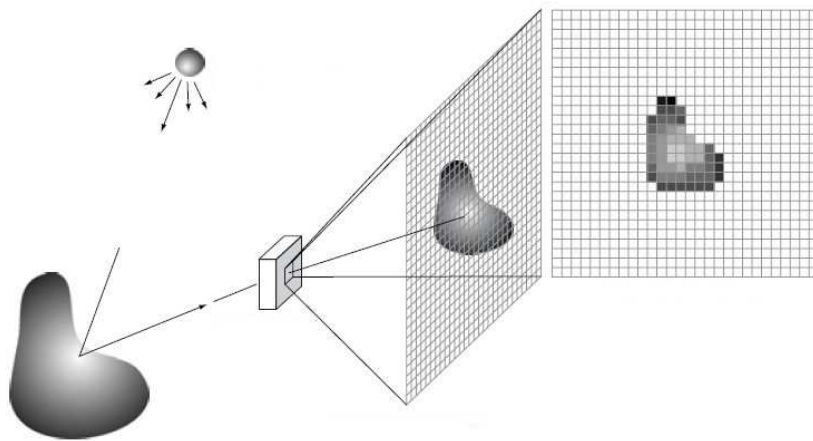


Figura 3.2: Processo de digitalização de imagem (Gonzalez & E.Woods, 2002).

ser assumidos pelo pixel.

As imagens que serão processadas são monocromáticas, ou seja, imagens que têm sua variação de intensidade do preto ao branco por variações de tons de cinza. Na figura 3.3 pode-se ver como a quantidade de níveis de cinza que cada pixel pode ter afeta a imagem.

Como visto anteriormente uma imagem digital não é nada mais que uma matriz com M linhas e N colunas, onde os valores contidos nela são chamados de pixels. Aqui nesta dissertação será usada a seguinte notação: a imagem digitalizada com M linhas e N colunas será representada por $I(x,y)$, sendo que suas coordenadas de origem são $(0,0)$ e a sua disposição espacial é como mostrada na figura 3.4.

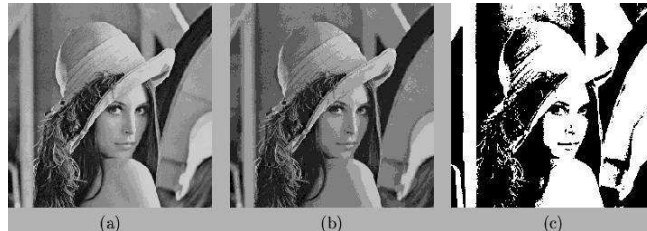


Figura 3.3: Imagem com 256 x 256 pixels. (a) 16 níveis de cinza (b) 8 níveis de cinza (c) 2 níveis de cinza (Gonzalez & E.Woods, 2002).

3.2 Relações entre os pixels

Para falar sobre o desenvolvimento do algoritmo, algumas relações entre pixels tem de ser definidas. Essas relações são definidas segundo (Gonzalez & E.Woods, 2002). A primeira relação entre pixels é a vizinhança. Com foi dito na seção 3.1, a imagem digital aqui será representada por $I(x,y)$ onde (x,y) são suas coordenadas espaciais. Assim sendo, um pixel p , com coordenadas (x,y) , tem dois pixels como vizinhos na horizontal e dois na vertical. Estes pixels são chamados de vizinhos do tipo 4, para o qual usa-se a notação de $N_4(p)$, sendo suas coordenadas são dados por $(x+1,y), (x-1,y), (x,y+1), (x,y-1)$.

Já os pixels que são vizinhos em relação a diagonal de p , são chamados de vizinhos do tipo D . A notação usadas para eles é $N_D(p)$.

A união dos pixels tipo $N_D(p)$ com os do tipo $N_4(p)$ é chamada de vizinhos do tipo 8, para o qual usa-se a notação $N_8(p)$.

A segunda relação entre pixels é adjacência. Neste trabalho serão apresentados apenas dois tipos de adjacência. Uma é a adjacência do tipo 4 e a outra é a do tipo 8.

- Dois pixels q e p têm adjacência do tipo 4 se eles têm o mesmo valor z e se $q \in N_4(p)$.
- Dois pixels q e p têm adjacência do tipo 8 se eles têm o mesmo valor z e se $q \in N_8(p)$.

A terceira relação entre pixels é a conectividade. Mas para definir conectividade entre pixels é necessário fazer a definição de “caminho”. Sejam dois pixels p e q , com coordenadas (x,y) e (s,t) respectivamente, defini-se caminho como uma seqüência de diferentes pixels com coordenadas $(x_0,y_0), (x_1,y_1), \dots, (x_n,y_n)$, onde $(x_0,y_0) = (x,y)$, $(x_n,y_n) = (s,t)$, e os pixels (x_i,y_i) e (x_{i-1},y_{i-1}) são adjacentes, e ligam estes dois pixels entre si. Assim o tipo de adjacência influencia diretamente no caminho que liga dois pixels entre si. Agora seja um grupo de pixels

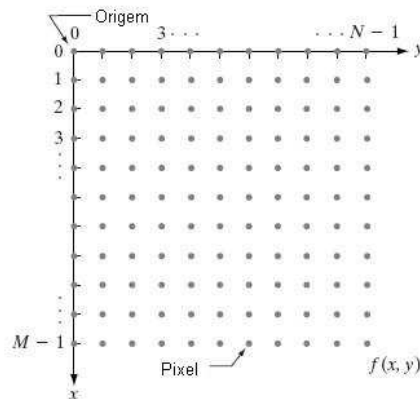


Figura 3.4: Disposição espacial de uma imagem digitalizada (Gonzalez & E.Woods, 2002).

S em uma imagem. Dois pixels p e q são conectados em S se existe um caminho de pixels que pertence a S e ligue p a q .

A quarta relação entre pixels é a região. Pode-se definir região como um conjunto de pontos R , onde cada pixel pertencente a R está conectado a qualquer outro pixel pertencente a R , por meio de um caminho contido em R .

A quinta relação entre pixels é o contorno. O contorno de uma região R é um conjunto de pixels onde cada um tem pelo menos um pixel na sua vizinhança que não pertence a R .

A sexta e ultima relação é o que em inglês chama-se de Edge. O edge é a medida da descontinuidade nos níveis de cinza presente em uma imagem e pode ser quantificada através do processo de derivação aplicada à imagem.

3.3 A segmentação da imagem

Com já foi dito no capítulo introdutório desta dissertação, a segmentação é uma das tarefas mais difíceis e importantes do processamento de imagem. Uma segmentação inadequada pode levar a resultados insatisfatórios. Sendo assim, a segmentação mostrou-se ser um ponto crucial para o desenvolvimento do algoritmo aqui proposto. Segmentar uma imagem é extrair desta, regiões que tenham entre si um certo grau de similaridade. No nosso caso a similaridade é com a relação ao nível de cinza presente na região. Assim, as regiões retiradas pelo processo de segmentação têm entre si um nível de cinza muito parecido, igual ou com uma margem de diferença bem próxima. Esta margem depende muito da informação contida na imagem e do que se quer retirar dela.

Nas subseções desta seção será mostrada a base matemática necessária para implementar o processo de segmentação, os algoritmos aqui utilizados para a tentativa de segmentar de forma

mais eficiente a imagem e o algoritmo criado para solucionar o problema de uma forma mais eficiente.

3.3.1 Derivadas em imagem digital

Segundo (Gonzalez & E.Woods, 2002), a derivada de uma imagem digital é dada pela diferença entre os pontos presentes na imagem. Assim a derivada de primeira-ordem em relação à x e à y são como expressas em 3.1 e 3.2.

$$\frac{\partial I}{\partial x} = I(x+1, y) - I(x, y) \quad (3.1)$$

$$\frac{\partial I}{\partial y} = I(x, y+1) - I(x, y) \quad (3.2)$$

Já a derivada de segunda-ordem em relação à x e à y são como expressas em 3.1 e 3.2.

$$\frac{\partial^2 I}{\partial x^2} = I(x+1, y) + I(x-1, y) - 2 \cdot I(x, y) \quad (3.3)$$

$$\frac{\partial^2 I}{\partial y^2} = I(x, y+1) + I(x, y-1) - 2 \cdot I(x, y) \quad (3.4)$$

Com as definições de derivada de primeira e de segunda ordem dadas acima, pode-se agora calcular o gradiente e o laplaciano. O gradiente é definido através de 3.5,

$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \quad (3.5)$$

onde ∇I é um vetor de duas linhas por uma coluna e representa a direção onde ocorre a taxa de variação de $I(x, y)$ em relação à coordenada (x, y) . A magnitude do vetor gradiente é dada por 3.6,

$$\nabla = \sqrt{G_x^2 + G_y^2} \quad (3.6)$$

sendo que a magnitude do vetor gradiente representa a taxa de crescimento de $I(x, y)$ por unidade de distância na direção de ∇I .

O laplaciano é como mostrado através de 3.7,

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (3.7)$$

logo,

$$\nabla^2 I = [I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1)] - 4 \cdot I(x, y) \quad (3.8)$$

3.3.2 Máscaras de operação

O interesse aqui reside em mostrar como implementar as operações que serão utilizadas no processamento de imagens desta dissertação. A implementação será feita por meio de máscaras, que terão dimensões de M por N , onde $M = N$, sendo que M será sempre um número ímpar (Gonzalez & E.Woods, 2002). Estas máscaras são compostas por coeficientes cujos valores dependem do tipo de operação que se deseja implementar. Assim, existem máscaras específicas para implementar cada uma das operações desejadas. A figura 3.5 mostra a máscara com seus coeficientes.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 3.5: Máscara matricial para operação em imagens digitais.

Com essa máscara é possível implementar o gradiente, laplaciano, entre outros operadores. A máscara mostrada na figura 3.5 é passada pela imagem da esquerda para a direita, de cima para baixo, sempre tendo como ponto de operação o pixel central. Assim, a resposta da máscara no ponto (x, y) da imagem é dada por 3.9,

$$\begin{aligned}
 g(x, y) = & w_1 \cdot I(x-1, y-1) + w_2 \cdot I(x-1, y) + w_3 \cdot I(x-1, y+1) + \\
 & + w_4 \cdot I(x, y-1) + w_5 \cdot I(x, y) + w_6 \cdot I(x, y+1) + \\
 & + w_7 \cdot I(x+1, y-1) + w_8 \cdot I(x+1, y) + w_9 \cdot I(x+1, y+1)
 \end{aligned} \tag{3.9}$$

onde $g(x, y)$ é a imagem fruto da operação feita sobre a imagem $I(x, y)$. Todo esse processo pode ser visto na figura 3.6.

3.3.3 Detecção de descontinuidades

A detecção de descontinuidades foi a primeira tarefa a ser realizada no processo de segmentação. A detecção das descontinuidades presentes na imagem consiste em detectar as variações do

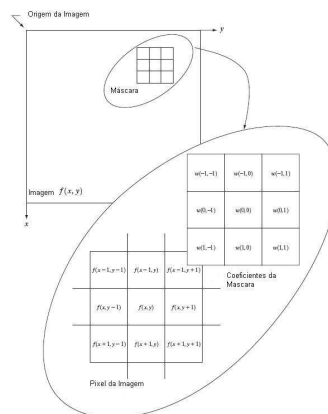


Figura 3.6: Processo de operação de imagem via máscara.

nível de cinza da imagem, à medida que se desloca espacialmente nela. Para calcular as descontinuidades, faz-se o uso da derivada. No caso de imagens bidimensionais as descontinuidades são detectadas tanto na vertical quanto na horizontal. A detecção de descontinuidades pode ser feita por meio da primeira e segunda derivada. Sendo assim, a primeira derivada de uma imagem digital é o seu gradiente, já a segunda é o seu laplaciano. Existem várias maneiras de se implementar o operador gradiente. As aproximações mais comuns para o gradiente são de Prewitt e Sobel (Gonzalez & E.Woods, 2002). As máscaras que implementam essas duas aproximações para o gradiente são mostradas logo a seguir. Na figura 3.7 é mostrada a máscara de Prewit e na figura 3.8 é mostrada a máscara de Sobel.

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Figura 3.7: Máscara que implementa Prewit.

Um outro detector muito utilizado para realizar a detecção de *edges* e bordas é o detector de Canny (Gonzalez & E.Woods, 2002). Ele suaviza o ruído presente na imagem e localiza a borda dos objetos presentes.

O algoritmo que implementa o operador de Canny pode ser dividido em 4 partes:

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Figura 3.8: Máscara que implementa Sobel.

- Uniformização da imagem - a imagem é uniformizada por meio de uma função gaussiana bidimensional;
- Diferenciação - a imagem é diferenciada utilizando um operador ótimo, que pode ser aproximado pela derivada da função gaussiana;
- Supressão não Máxima - é feita a omissão dos pontos que não máximos locais em relação a direção transversal da borda;
- Limiarização de borda - nesta etapa é feita a limiarização baseada em dois limiares, T_1 e T_2 , que neste trabalho tiveram seus valores escolhidos de forma empírica.

A função gaussiana bidimensional e sua derivada são mostradas em 3.10 e 3.11.

$$G(x) = e^{\left(\frac{-x^2}{2 \cdot \sigma^2}\right)} \quad (3.10)$$

$$G'(x) = \frac{x}{\sigma^2} e^{\left(\frac{-x^2}{2 \cdot \sigma^2}\right)} \quad (3.11)$$

As etapas de suavização e diferenciação são feitas utilizando máscaras que implementam essas funções.

Capítulo 4

Detecção de veículos na faixa de pedestre

4.1 A busca por características

O processo de detecção de veículos sobre a faixa de pedestres foi organizada como mostrado abaixo:

1. Antes de realizar qualquer processamento de imagens e/ou reconhecimento de padrão, é necessário encontrar características que possam ser utilizadas para realizar o reconhecimento do padrão veículo;
2. Uma vez que essa(s) característica(s) seja(m) escolhida(s), deve-se escolher ou desenvolver o melhor método para realizar a extração desta(s) característica(s).

Assim se buscou características que poderiam ser utilizadas para a identificação do padrão veículo. Existem várias características que podem ser extraídas, como pode ser visto nos trabalhos de (Schneiderman & Kanade, 2000) e (A.Silva & Gonzaga, 2006). Foram escolhidas duas possíveis características, com base no trabalho de (Leung, 2004). Uma está associada à placa do veículo e a outra à roda. A utilização destas duas características ao mesmo tempo não é desejada, já que isso implicaria no dobro do custo computacional, pois seria necessária à aquisição e processamento de duas situações distintas, quando comparado com a extração de apenas uma. Fora o problema de dobrar o custo computacional, ainda seria necessário a utilização de duas câmeras diferentes, gerando a necessidade de sincronismo na aquisição dos dados, o que também implicaria no aumento da dificuldade de aquisição das imagens. Sendo assim, chegou-se a conclusão de que apenas uma das características poderia ser utilizada, já que o baixo custo computacional é uma das prioridades deste trabalho. A característica escolhida foi a roda. A justificativa para a escolha da roda foi feita com base nos argumentos colocados abaixo:

1. Na prática a aquisição das imagens contendo rodas é bem mais fácil do que aquelas contendo a(s) placa(s);

2. A dificuldade de se realizar a filmagem dos carros de frente, implica que a forma geométrica da placa varia com o deslocamento do carro sobre a faixa.

Como a forma geométrica da roda não varia significativamente com o deslocamento dos veículos, optou-se pela utilização da roda como característica a ser extraída. Assim, a primeira etapa para a identificação de veículos já estava resolvida. A próxima etapa agora seria realizar a extração desta característica de forma que ela pudesse ser reconhecida como círculo e assim identificar a presença de veículo(s) sobre a faixa. Para realizar a extração desta característica foi necessário segmentar a imagem, onde vários métodos foram testados e comparados.

4.2 A extração de características

A extração de características é uma etapa crucial para a identificação de veículo sobre a faixa. A sua importância vem do fato que, se o objeto desejado não for extraído corretamente, a identificação de veículo se tornará falha. A extração ou separação de objetos ou características presentes em uma imagem é feita através do processo de segmentação. O processo de segmentação foi dividido em duas etapas:

1. Detecção de borda ou contorno dos objetos presentes na imagem;
2. Filtragem e preenchimento dos objetos.

Como foi dito, para realizar a detecção de borda existem vários métodos que podem ser utilizados. Aqui procurou-se utilizar o método que fosse mais fácil de implementar, tivesse o menor custo computacional e desse como retorno uma boa resposta. No capítulo 3, vários métodos foram apresentados. Os mais comuns para realizar a detecção de bordas são Prewitt, Sobel e Canny. Mas além deles existem métodos como dilatação, erosão e limiar adaptativo entre outros. O primeiro método a ser utilizado foi o método de Sobel, cuja máscara utilizada para realizar sua implementação já foi descrita no capítulo 3. Sobel foi preferido em relação ao Prewitt, por causa de sua melhor resposta quanto a presença de ruídos (Gonzalez & E.Woods, 2002). O operador de Sobel realiza o cálculo do gradiente da imagem $I(x,y)$.

Para testar a eficiência do operador de Sobel, foi escrita uma função em C e na linguagem do MATLAB. Essa função realiza os seguintes passos:

1. Varrer a imagem da esquerda para a direita, de cima para baixo, de forma que a máscara sempre encontra-se dentro da imagem;
2. Em cada pixel (x,y) , calcula-se o componente do vetor gradiente na direção x por meio da equação 4.1 e na direção y por meio da equação 4.2;
3. Calcula-se o módulo do gradiente através da equação 4.3;
4. Compara-se se o valor do módulo do gradiente para cada pixel (x,y) com um certo

valor, chamado valor de limiar, que foi calculado de forma experimental. Se o gradiente do pixel (x, y) estiver acima do valor de limiar então o pixel recebe o valor 255, senão recebe o valor 0.

$$G_x(x, y) = w_3 \cdot I(x-1, y-1) + 2 \cdot w_6 \cdot I(x-1, y) + w_9 \cdot I(x-1, y+1) - \quad (4.1)$$

$$-w_1 \cdot I(x+1, y-1) - 2 \cdot w_4 \cdot I(x+1, y) - w_7 \cdot I(x+1, y+1)$$

$$G_y(x, y) = w_7 \cdot I(x-1, y-1) + 2 \cdot w_8 \cdot I(x-1, y) + w_9 \cdot I(x-1, y+1) - \quad (4.2)$$

$$-w_1 \cdot I(x+1, y-1) - 2 \cdot w_2 \cdot I(x+1, y) - w_9 \cdot I(x+1, y+1)$$

$$\nabla I(x, y) = \|G_x\| + \|G_y\| \quad (4.3)$$

O problema deste algoritmo está no fato de a detecção das bordas serem muito dependentes do valor de limiar e este, por sua vez, tem que ser ajustado à medida que as variações climáticas ocorrem ou quando se muda de uma faixa para outra. Essa necessidade de ajuste constante do valor de limiar não é aceitável, pois se deseja encontrar um método que, uma vez que ajustado, não precise ser reajustado. Nas figuras 4.1, 4.2, 4.3 e 4.4 é possível ver dois instantes diferentes da mesma faixa de pedestre, onde foi utilizado o mesmo valor de limiar. Nas figuras 4.1 e 4.2 o contorno externo da roda é retirado corretamente, enquanto que, nas figuras 4.3 e 4.4 a extração da roda não ocorre corretamente.



Figura 4.1: Imagem de faixa de pedestre.

Devido a essa deficiência do método de Sobel, partiu-se para utilização do método de Canny, que foi construído para ser um ótimo detector de bordas. Buscou-se então através de Canny



Figura 4.2: Imagem de faixa processada corretamente com o operador de Sobel.

suprimir a dependência com variações de luminosidade, produzidas por fatores diversos, tais como variações climáticas. Para realizar a sua implementação foi desenvolvida uma função em C e na linguagem MATLAB. A função que implementava o método de Canny seguiu os seguintes passos:

1. Suavização da imagem $I(x,y)$ na direção x por meio de uma máscara 1×5 , gerada através de uma função gaussiana com desvio padrão fornecido pelo usuário, resultando na imagem I_x e na direção y por meio de uma máscara 5×1 , gerada através de uma função gaussiana com desvio padrão fornecido pelo usuário, resultando na imagem I_y ;

2. Encontra-se a imagem suavizada através de 4.4;

$$I'(x,y) = \|I_x\| + \|I_y\| \quad (4.4)$$

3. Diferenciação da imagem suavizada, tanto na direção x como na direção y , por meio de uma máscara 1×5 e 5×1 respectivamente, obtida através da derivada da função gaussiana, gerando assim uma imagem Id_x e outra Id_y ;

4. Encontra a imagem diferenciada através de 4.5.

$$I''(x,y) = \|Id_x\| + \|Id_y\| \quad (4.5)$$

5. São suprimidos todos os pixels que não são máximos locais na direção perpendicular a borda;

6. Faz o processo de limiarização adaptativa por meio de $T1$ e $T2$, que neste trabalho foi calculado de forma experimental.

O método de Canny demonstrou os mesmos problemas que o operador de Sobel, entre outros, como descritos a seguir:

- Necessidade de ajustes constante em seus parâmetros;



Figura 4.3: Imagem de faixa de pedestre.

- Dificuldade de encontrar os valores mais adequados para os parâmetros do método;
- Dificuldade ou impossibilidade de se encontrar valores adequados durante o processo de reajuste dos parâmetros.

Com isso necessitou-se escolher um outro método que fosse capaz de realizar o processo de detecção de bordas das rodas de forma adequada e ainda ter um baixo custo computacional. Na figura 4.5 é possível ver a borda da roda sendo detectada de forma correta, enquanto na figura 4.6 é possível ver a borda da roda não sendo detectada de forma correta, para o método de Canny.

A solução encontrada para realizar a detecção das bordas das rodas de forma mais correta foi possível através do desenvolvimento de um novo algoritmo, que é descrito abaixo:

1. Suavização da imagem $I(x,y)$ na direção x por meio de uma máscara 1x5 gerada de uma função gaussiana com desvio padrão fornecido pelo usuário, resultando na imagem I_x . Enquanto que na direção y por meio de uma máscara 5x1, gerada através de uma função gaussiana com desvio padrão fornecido pelo usuário, resultando na imagem I_y ;
2. Encontra-se a imagem suavizada através de 4.4;
3. Diferenciação da imagem suavizada, tanto na direção x como na direção y , por meio de uma máscara 1x5 e 5x1 respectivamente, obtida através da derivada da função gaussiana, gerando assim uma imagem I_{dx} e outra I_{dy} ;
4. Encontra a imagem diferenciada através de 4.5;
5. Varre-se a imagem diferenciada da esquerda para a direita, de cima para baixo com uma máscara quadrada e de tamanho ímpar, que é fornecida pelo usuário aplicando 4.6;



Figura 4.4: Imagem de faixa processada incorretamente com o operador de Sobel.



Figura 4.5: Detecção de borda da roda feita corretamente pelo método de Canny.

$$M(x, y) = \frac{\sum_{i=-\frac{n-1}{2}}^{\frac{n-1}{2}} \sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} I''(x+i, y+j)}{n^2} \quad (4.6)$$

6. Compara-se o valor obtido com o parâmetro T , que é fornecido pelo usuário, utilizando 4.7;

$$\left\| \frac{I''(x, y) - M(x, y)}{M(x, y)} \right\| \geq 0 \quad (4.7)$$

7. Se a condição imposta pela equação 4.7 ou pela equação 4.8 for satisfeita, então o pixel $I(x, y)$ assumirá o valor 255, senão assumirá o valor 0.

$$\left\| I''(x, y) - M(x, y) \right\| \leq T \quad (4.8)$$

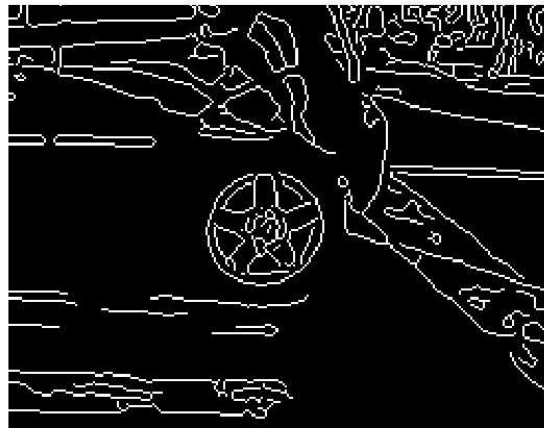


Figura 4.6: Detecção de borda da roda feita de forma incorreta pelo método de Canny.

O grande segredo deste algoritmo está basicamente na diferenciação com a máscara originada da derivada da função gaussiana, juntamente com a utilização das equações 4.6 e 4.7. Através do processo de diferenciação, as regiões da imagem onde não existem variações de tons de cinza têm seus valores aproximadamente zerados, enquanto que regiões de transição são enfatizadas. Ao olhar com mais atenção para a equação 4.6, pode-se notar que ela é a média para uma certa região R da imagem $I(x,y)$, onde o tamanho desta região é dado pelo valor do parâmetro n . A aplicação das equações 4.6 e 4.7 à imagem, sem esta sofrer um processo de diferenciação, nada mais é do que uma filtragem por média que tende a suavizar a imagem original, funcionando como um filtro passa-baixas (Gonzalez & E.Woods, 2002). No entanto, quando aplicadas a uma imagem que foi diferenciada, a equação 4.6 e 4.7 zeram valores que estejam percentualmente abaixo da média em um certo valor T . Isto significa que pixels pertencentes às bordas terão seus valores iguais a 255, enquanto que os outros terão valores iguais a zero. Com isso, é apenas necessário encontrar os valores adequados para os parâmetros n e T . O parâmetro T informa em percentual, quanto valor do pixel (x,y) pode estar abaixo do valor médio calculado com a equação 4.6.

Para ficar clara a influência de n no processo de detecção de bordas, tem-se que dividir sua análise em duas etapas. Uma é associada ao aumento do valor de n e outra a decréscimo do mesmo. Dentro de cada uma dessas duas etapas existem ainda outras duas situações, como mostradas abaixo:

1. O pixel em análise está na região próxima da borda ou na região de transição entre borda e fundo da imagem;
2. O pixel em análise está na região de borda.

Vamos considerar primeiramente o caso em que o valor de n está sendo aumentado. A medida em que n tem seu valor aumentado, mais valores de pixel estão sendo acrescentados

no cálculo da média. Se o pixel em análise se encaixa na situação 1, então, a depender do tamanho da máscara, valores bem maiores que o pixel (x,y) serão introduzidos no cálculo da média, o que acarreta em uma aumento desta e conseqüentemente na exclusão deste pixel em relação a pertencer a borda. Assim, ao processar uma imagem $I(x,y)$ com um tamanho n e depois processar a mesma imagem com um valor n maior que o anterior, tem-se uma imagem com menos ruídos e uma borda mais fina. Para o caso em que se aumenta o valor de n e o pixel (x,y) em análise se encaixa na situação 2, isto implicar em diminuir o valor da média a medida que o valor de n é aumentado. Assim, (x,y) sempre será incluído como um pixel pertencente a uma região de transição. Já para o caso em que o valor de n vai sendo diminuído gradativamente, o processo inverso ao mostrado no caso anterior, ocorre com o pixel que se encontra na situação 1. Enquanto que, no caso do pixel que se encontra na situação 2, ele mantém o mesmo comportamento que o pixel que se encontra na situação 1. Este método aqui proposto teve bons resultados, uma vez que ajustando adequadamente seus parâmetros n e T , ele mostrou boa insensibilidade as variações de luminosidade, detectando corretamente as bordas das rodas dos veículos, sem necessidade de reajuste, como pode ser visto no capítulo 6, onde os resultados experimentais deste método são mostrados. As vantagens deste método são apresentadas abaixo:

1. Facilidade em encontrar valores adequados para seus parâmetros de ajuste;
2. Excelente insensibilidade às variações climáticas e de luminosidade;
3. Grande taxa de acerto para a detecção de bordas dos objetos presentes na imagem;
4. Custo computacional inferior a 1 segundo.

Devido a todas estas características citadas acima, este método, que foi chamado de detecção de borda por média e diferenciação, foi escolhido em detrimento dos outros aqui citados anteriormente. As figuras 4.7, 4.8, 4.9 e 4.10 mostram a imagem original e as etapas de suavização, diferenciação e detecção de bordas consecutivamente.

4.3 Reconhecimento de roda

Uma vez que a imagem já foi segmentada pelo algoritmo descrito na seção anterior, o próximo passo é realizar o reconhecimento do padrão roda. Este padrão pode ser classificado através de uma forma geométrica muito comum, o círculo. A forma geométrica circular pode ser facilmente identificada através apenas de sua área e seu perímetro (K. Pratt, 2001). Para isso usa-se a equação 4.9, cujo o valor é chamado de circularidade do objeto.

$$C_0 = \frac{4 \cdot \pi \cdot A_0}{P_0^2} \quad (4.9)$$

Na equação 4.9, A_0 representa a área do objeto que se deseja identificar e P_0 o seu perímetro. Todo objeto que estiver presente na imagem e tiver forma circular irá fazer com que o uso



Figura 4.7: Imagem original.



Figura 4.8: Imagem resultante do processo de suavização.

da equação 4.9 resulte no valor um. No entanto, as rodas não são 100% uniformes em sua composição, tendo buracos dentro do seu contorno circular, o que impede que a equação 4.9 seja usada diretamente. Para resolver este problema, utilizou-se um algoritmo que faz a identificação de cada um dos objetos presentes na imagem, marcando o conjunto de pixels que compõem cada objeto (Gose et al., 1996). Este algoritmo é descrito abaixo:

1. Varre-se a imagem, processando-a com o algoritmo descrito na subseção 4.2, da esquerda para a direita, de cima para baixo;
2. Se um pixel com valor igual a 255 é encontrado, então ele é marcado como lido e seu índice (x,y) é colocado em uma lista;
3. Retira-se o índice do último pixel colocado na lista e testa, no sentido horário, se seus vizinhos do tipo 4 têm valor igual a 255. Quem tiver valor igual a 255 é marcado e tem seu

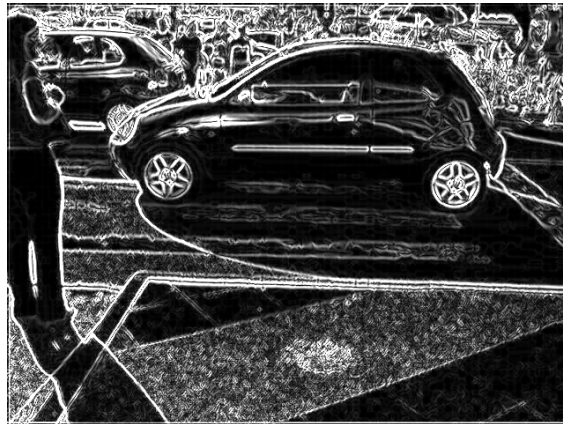


Figura 4.9: Imagem resultante do processo de diferenciação.

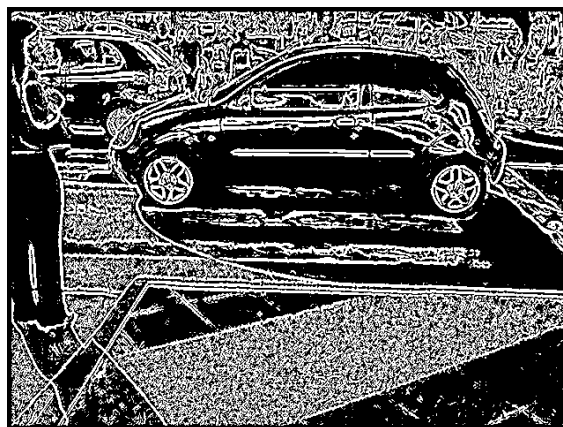


Figura 4.10: Imagem resultante do processo de detecção de bordas.

índice colocado na lista;

4. Se nenhum dos vizinhos analisados tiver valor igual a 255, verifica-se se a lista está vazia. Se estiver, volta-se para o passo 1, senão volta-se para o passo 3.

Este processo é repetido até não restar mais nenhum pixel sem ter sido analisado na imagem.

No entanto, este algoritmo só permite identificar que pixels pertencem a cada objeto presente na imagem. Para poder identificar se o objeto encontrado por este algoritmo é ou não uma roda, foi então desenvolvido um outro algoritmo que realiza o preenchimento dos buracos presente na figura em análise, calcula a sua área total, seu perímetro, suas dimensões horizontal e vertical, e o seu fator geométrico, identificando se o objeto em análise é ou não um círculo. Este algoritmo tem seus passos mostrados seguir:

1. Varre-se a imagem segmentada da esquerda para a direita, marca cada pixel que

pertence à borda como um pixel que compõe o perímetro e cada pixel que pertence ao objeto como um pixel de área;

2. Calcula-se as dimensões horizontal e vertical do objeto;
3. Testa se suas dimensões horizontal e vertical têm tamanhos próximos. Se sim, vai para o passo 4, senão volta para o passo 1;
4. Preenche-se os espaços vazios presentes dentro do objeto em análise;
5. Calcula-se o seu fator geométrico utilizando-se 4.9.
6. Teste se o valor do fator geométrico está dentro da região $0,9 < C_0 < 1,4$. Se sim, é classificado como círculo, caso contrário não.

A unidade mínima utilizada no cálculo da área e do perímetro é 1, que corresponde a um pixel. Nas figuras 4.11, 4.12, 4.13 e 4.14 é possível ver a cada uma das etapas do algoritmo aqui descrito. A escolha da região de valores para C_0 veio dos testes experimentais que foram feitos com este método e do fato que ao posicionar a câmera a uma certa altura isto implica em uma distorção da forma circular da roda, tornando sua forma geométrica um pouco oval.



Figura 4.11: Imagem sem processamento.



Figura 4.12: Imagem sendo processada com o algoritmo de identificação de círculo.

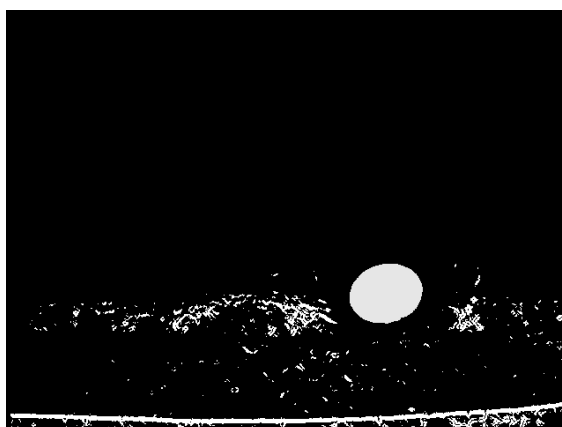


Figura 4.13: Imagem processada pelo algoritmo de identificação de círculo.

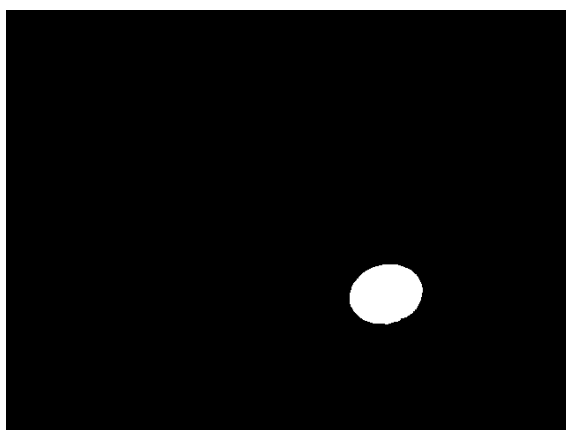


Figura 4.14: Imagem processada pelo algoritmo de detecção de veículo.

Capítulo 5

Identificação de Pessoas na Faixa de Pedestre

5.1 Identificação de Pessoas na Faixa através RNA

Com foi exposto na seção 1.3 do capítulo 1, a utilização de redes neurais se mostrou mais viável para o reconhecimento do padrão pessoas sobre a faixa de pedestre. A justificativa da utilização de RNA's é mostrada nos pontos abaixo:

1. Grande variação das formas das pessoas presentes nesta situação, dificultando a identificação de uma característica que pudesse ser utilizada para reconhecer o padrão pessoas;
2. Os outros algoritmos que poderiam ser utilizados para realizar o processo de identificação, (Shashua et al., 2004), (Oren et al., 1997) , tinham um custo computacional superior ao que foi estabelecido no capítulo 1;
3. Extensa utilização para resolver problemas de reconhecimento de padrões.

Em comparação ao padrão veículos, que têm uma característica que não se altera muito no tempo, o padrão pessoas é mais complexo, gerando uma grande gama de formas. O reconhecimento deste padrão depende da quantidade de pessoas, da posição delas, das roupas e quantidade de carros presentes na imagem. Os trabalhos que foram encontrados para o reconhecimento de pessoas, (Shashua et al., 2004), (Oren et al., 1997), (Zhao & Nevatia, 2003) e (Tabb et al., 1999), não tinham sido aplicados neste contexto. Portanto, não ofereciam garantia de sucesso, e ainda tinham um custo computacional superior ao que foi estabelecido no capítulo 1. Como foi dito no início deste trabalho, o objetivo é encontrar uma solução que atenda ao tempo de processamento estabelecido no capítulo 1, de fácil implementação e que consiga realizar a detecção da infração. Então, baseado nas argumentos acima citados, se optou pela utilização de redes neurais.

5.2 Identificação de Pessoas na Faixa através de Perceptron de Múltiplas Camadas

Partindo sempre do critério de tentar utilizar a solução mais simples para a resolução do problema, montou-se a rede contendo uma camada de entrada, uma camada oculta e uma camada de saída, como mostrado na figura 5.1.

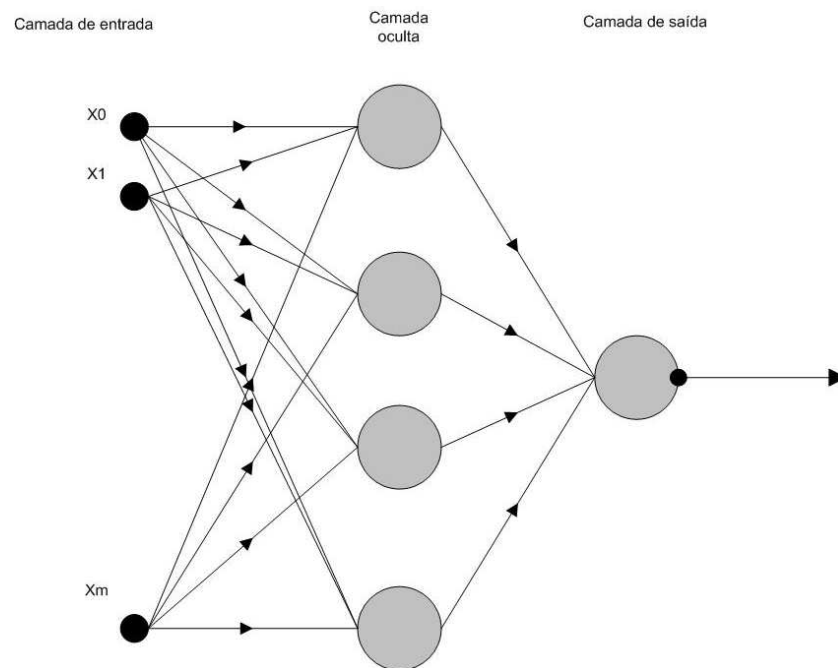


Figura 5.1: Perceptron com uma camada oculta.

Na figura 5.1, pode-se ver que na arquitetura de rede proposta, a camada oculta é composta de 4 neurônios e a camada de saída apenas por um neurônio. A utilização de apenas um neurônio de saída se dá pelo fato de que a rede só precisa dar uma resposta positiva ou negativa, indicando a presença ou não de pessoas na faixa.

Para realizar o treinamento da rede foi escolhido o algoritmo de retropropagação de erro, por este já ter sido aplicado com grande sucesso para resolver diversos problemas de grande dificuldade em conjunto com redes neurais de múltiplas camadas alimentadas adiante (Haykin, 2001). Este algoritmo de treinamento se encontra na categoria de métodos de treinamento supervisionado e tem seu funcionamento aplicado a este problema com descrito abaixo:

1. Um lote contendo m imagens em formato JPG de 240x320 pixels é apresentado à rede, juntamente com a sua indicação de ter presença ou não de pessoa;
2. A imagem m , no passo n , é convertida em um vetor linha x de 1x76800, que é aplicado à camada de entrada;
3. Calcula-se o campo local induzido para os neurônios da camada 1, que é a camada oculta, usando a equação 5.1 e a equação 5.6.

4. Passa-se a resposta de saída de cada neurônio da camada oculta para a entrada do neurônio da camada de saída;
5. Calcula-se a resposta do neurônio da camada de saída através de 5.1 e 5.6, e calcula-se o sinal de erro através de 5.2.
6. Calcula-se o gradiente do neurônio da camada de saída através de 5.3 e faz-se a correção dos seus pesos sinápticos através de 5.4;
7. Calcula-se o gradiente de cada neurônio da camada oculta através de 5.5 e faz-se a correção de seus pesos sinápticos através de 5.4;
8. Repete-se os passos de dois a sete até que a rede chegue ao valor de erro quadrático mínimo pré-estabelecido.

O valor de erro desejado, neste caso, foi 0,1 do erro máximo possível, que é 2,7159. Este valor de erro foi obtido a partir da utilização de uma função de ativação anti-simétrica mostrada em 5.6, onde os valores ótimos para a e b são 1,7159 e $2/3$ consecutivamente (Haykin, 2001).

$$v_j^l(n) = \sum_{i=0}^M w_{ji}^l(n) \cdot y_i^{l-1} \quad (5.1)$$

Na equação 5.1 l é a camada, M a quantidade de sinais que estão conectado ao neurônio j , i o neurônio ou sinal de entrada se a camada i for a camada de entrada, y o sinal de saída do neurônio i ou sinal de entrada se a camada for de entrada e w o valor da peso sináptico que conecta o neurônio i ao j .

$$e_j^L(n) = d_j(n) - y_j^L \quad (5.2)$$

Na equação 5.2 L indica o a última camada e a profundidade da rede também e $d_j(n)$ o valor esperado para a saída do neurônio j .

$$\delta_j^L(n) = e_j^L(n) \cdot \varphi_j'(v_j^L(n)) \quad (5.3)$$

Na equação 5.3 φ_j' é a derivada da função de ativação.

$$w_{ji}^l(n+1) = w_{ji}^l(n) + \eta \cdot \delta_j^l(n) \cdot y_i^{l-1}(n) \quad (5.4)$$

Na equação 5.4 η é a de taxa de aprendizagem.

$$\delta_j^l(n) = \varphi_j'(v_j^l(n)) \cdot \sum_k \delta_k^{l+1}(n) \cdot w_{kj}^{l+1}(n) \quad (5.5)$$

Na equação 5.5, k é o número de neurônios da camada à frente conectados ao neurônio da camada anterior j .

$$\varphi_j(n) = a \cdot \tanh(b \cdot v_j(n)) \quad (5.6)$$

Esta arquitetura usada se mostrou bastante eficiente, conseguindo convergir de forma rápida e identificando de forma correta o padrão desejado. Sua taxa de acerto mostrou-se dependente da diversidade de situações possíveis que fossem apresentadas à rede durante o processo de treinamento. Esta dependência já era esperada, sendo comentada em (Haykin, 2001) e não representa um problema para este algoritmo. Assim, para se ter uma alta taxa de acerto para esta arquitetura de rede, treinada com o algoritmo de retropropagação, deve-se utilizar a maior quantidade possível de situações diversas que os arquivos de vídeo das faixas de pedestres filmadas possam apresentar. Com isso, a taxa de acerto pode ser aumentada.

Visando melhorar a taxa de acerto na identificação de pessoas sobre a faixa, adicionou-se mais uma camada oculta a arquitetura de rede já existente. Na figura 5.2 é possível ver esta nova arquitetura. O algoritmo de treinamento utilizado foi o mesmo utilizado para a arquitetura proposta na figura 5.1.

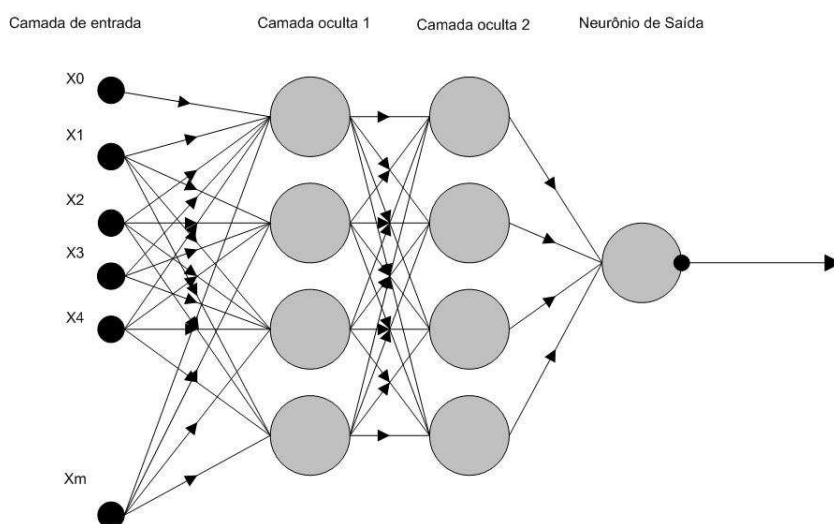


Figura 5.2: Perceptron com duas camadas ocultas.

O algoritmo associado à arquitetura proposta na figura 5.2 não conseguiu convergir. Várias iterações foram processadas e nenhuma tendência a convergência ocorreu. A não convergência deste algoritmo deve estar associada a grande quantidade de pontos de entrada que tem o padrão que se deseja identificar. Com base nesta análise foi descartado o uso deste tipo de arquitetura para realizar a detecção do padrão pessoa sobre a faixa.

Assim como esta arquitetura com duas camadas ocultas foi descartada, a utilização de um número maior de camadas ocultas também foi. Restava agora a opção de aumentar a quantidade de neurônios que compõem a arquitetura com uma única camada oculta. Resolveu-se então dobrar a quantidade de neurônios que compunham a camada oculta, visando uma maior eficiência no processo de detecção de infração. Logo, a nova arquitetura proposta ficou como mostrada na figura 5.3.

Essa nova arquitetura convergiu com quase o mesmo número de iterações que a arquitetura

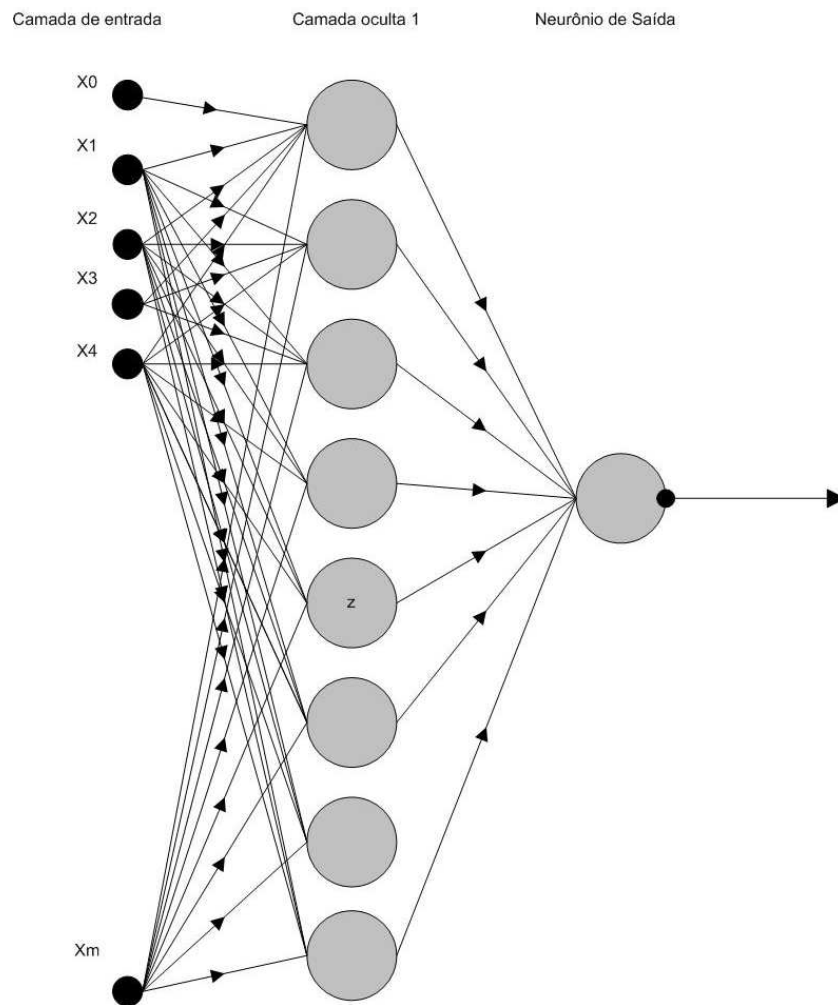


Figura 5.3: Perceptron com uma camada oculta contendo 8 neurônios.

contendo apenas 4 neurônios. Isso mostra que o aumento do número de neurônios na arquitetura contendo apenas uma camada oculta não influencia no processo de convergência, pelo menos para um aumento no número de neurônios até oito. O gráfico contendo o número de iterações versus erro de saída, para a arquitetura contendo oito neurônios na camada oculta, pode ser visto na figura 5.4.

5.3 Identificação de pessoas na faixa através Mapa Auto-Organizável

Com a boa resposta da arquitetura proposta na seção 5.2, restava agora tentar encontrar uma outra arquitetura que pudesse ser utilizada para o processo de detecção de infração. A rede proposta anteriormente utilizava um algoritmo supervisionado para realizar o processo de convergência do algoritmo de treinamento. A idéia agora era utilizar o algoritmo não-supervisionado, que aprendesse por si só. Assim, poder-se-ia comparar a eficiência entre um

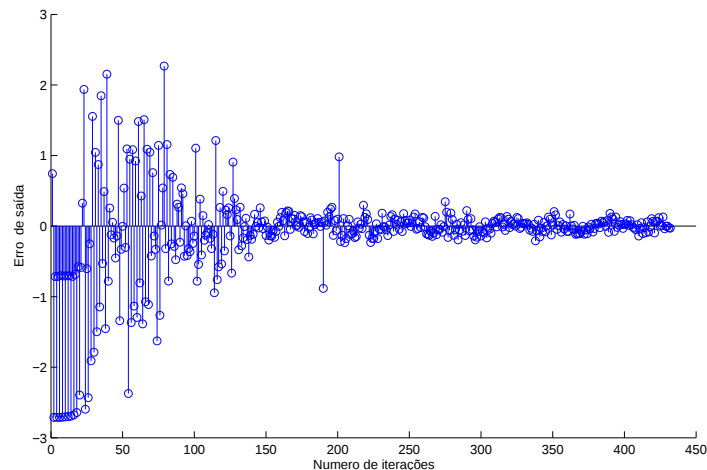


Figura 5.4: Gráfico do número de iterações pelo erro de saída para o perceptron com uma camada oculta contendo oito neurônios.

algoritmo supervisionado e não supervisionado para realizar o processo de detecção de infração. Além de se ter uma outra opção de algoritmo para identificação de pessoas sobre a faixa. O algoritmo escolhido para realizar este processo foi a rede SOM ou Self-Organizing Map. O SOM tem a capacidade de realizar a transformação de um padrão de entrada em um mapa discreto, formado por uma quantidade de neurônios, que adquire um padrão topológico em resposta a este padrão. Na figura 5.5, pode-se ver a arquitetura da rede utilizada pelo algoritmo SOM. A grade que é mostrada na figura 5.5 pode ser unidimensional também.

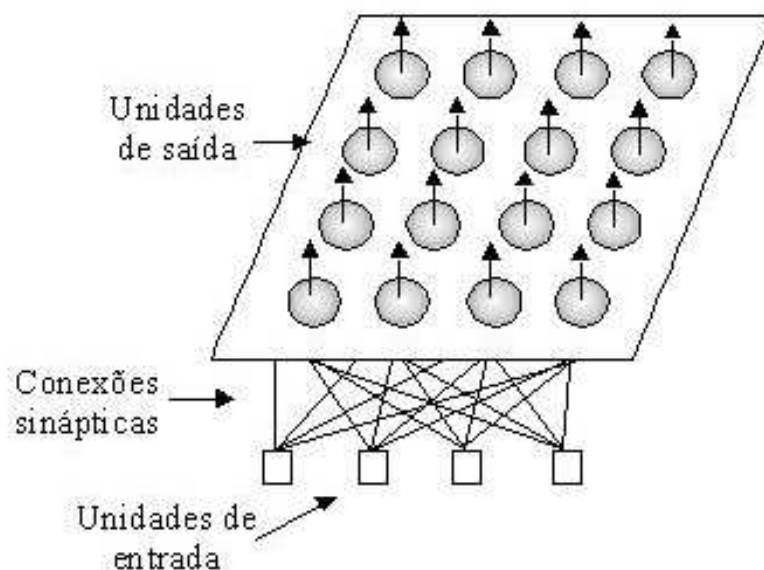


Figura 5.5: Arquitetura da rede para o algoritmo SOM.

Para a implementação do SOM foi escolhido um mapa contendo dezesseis neurônios. Tendo

este mapa o formato bidimensional de uma matriz 4x4, como mostrado na figura 5.5. O funcionamento do algoritmo SOM aplicado ao problema de detecção de pessoas sobre a faixa de pedestres é mostrado logo a seguir:

1. São escolhidos os valores iniciais dos pesos sinápticos de cada neurônio $w_j(0)$ de forma aleatória;
2. Utiliza a imagem segmentada para calcular qual neurônio que compõe a grade tem o melhor casamento com essa imagem, usando 5.7;

$$i(x) = \arg \min_j \|x(n) - w_j(n)\| \quad (5.7)$$

3. Faz o ajuste dos pesos sinápticos de cada neurônio da grade através de 5.7, onde $\eta(n)$ é dado por 5.9 e $h_{j,i(x)}(n)$ é dado por 5.10;

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n) (x(n) - w_j(n)) \quad (5.8)$$

$$\eta(n) = 0.1 \cdot \exp\left(-\frac{n}{1000}\right) \quad (5.9)$$

$$h_{j,i(x)}(n) = \exp\left(-\frac{d_{ij}^2}{2\sigma^2(n)}\right) \quad (5.10)$$

$$\sigma(n) = \sigma_0 \cdot \exp\left(-\frac{n}{\frac{1000}{\log \sigma_0}}\right) \quad (5.11)$$

4. Volta-se para o passo 2 enquanto o mapa de neurônios não formar uma topologia que varie muito pouco com a presença de pessoas sobre a faixa.

O valor de σ_0 deve assumir o valor do raio da grade utilizada (Haykin, 2001). No entanto, a tentativa de uso do algoritmo SOM não obteve sucesso. A grade não conseguiu convergir, ou seja, estabelecer um padrão topológico definido sempre que existisse sobre a faixa de pedestres a presença de pessoa(s) e veículo(s), muito provavelmente devido a grande quantidade de pontos que o vetor de entrada possui. Assim a solução encontrada para este problema foi utilizar uma arquitetura de rede de perceptron de múltiplas camadas alimentadas adiante, com o algoritmo de retropropagação para treinamento da rede, sendo esta arquitetura mostrada na figura 5.1.

Capítulo 6

Resultados

6.1 Resultados dos testes experimentais

Para realizar os testes do algoritmo proposto neste trabalho, foram realizadas três filmagens, sendo que cada uma delas foi feita em diferentes horários do dia e em três faixas diferentes. Os arquivos de vídeo foram gerados em formato MPEG, com uma resolução de 240 por 320 pixels. Todos os testes foram realizados com a ferramenta de software Matlab e com as funções implementadas ao longo deste trabalho. Com os arquivos em mãos, o primeiro teste a ser feito foi o de detecção de carro na faixa de pedestre, uma vez que esta é a primeira parte a ser executada para a identificação da infração. Nas tabelas 6.1, 6.2 e 6.3 encontram-se os resultados de detecção de veículo na faixa, para o algoritmo de Sobel, Canny e o algoritmo de detecção de borda por média respectivamente. Neste teste foi utilizado para todos os algoritmos de detecção de borda 200 imagens, cada uma contendo 100 imagens com presença de veículo(s) e 100 contendo presença apenas de pessoas. Os valores de limiar de Sobel e Canny foram escolhidos com base nos testes experimentais, sendo que estes valores deram os melhores resultados.

Tabela 6.1: Resultados do algoritmo de detecção de veículos utilizando o algoritmo de Sobel.

Faixa	Nº de Imagens	Valor Limiar	Acerto sem carro	Acerto com carro
Faixa 1	200	153	100%	52%
Faixa 2	200	153	99%	64%%
Faixa 3	200	153	92%	52%

Como é possível ver nas tabelas 6.1, 6.2 e 6.3 o algoritmo de detecção de bordas por média proporciona uma taxa de acerto maior que os algoritmos de Sobel e Canny, sendo por isso escolhido como o algoritmo a ser utilizado na detecção de bordas.

O segundo teste a ser feito foi de convergência da rede neural. Os resultados deste teste podem ser vistos na tabela 6.4. Neste teste foram utilizadas 100 imagens para realizar o

Tabela 6.2: Resultados do algoritmo de detecção de veículos utilizando o algoritmo de Canny.

Faixa	Nº de Imagens	Limiar 1	Limiar 2	Acerto sem carro	Acerto com carro
Faixa 1	200	0,08	0,3	100%	10%
Faixa 2	200	0,08	0,3	100%	11%
Faixa 3	200	0,08	0,3	100%	14%

Tabela 6.3: Resultados do algoritmo de detecção de veículos com o algoritmo de detecção de borda por média.

Faixa	Nº de Imagens	Tamanho da Máscara	Acerto sem carro	Acerto com carro
Faixa 1	200	7	100%	85%
Faixa 2	200	7	92%	90%
Faixa 3	200	7	93%	90%

treinamento da rede neural escolhida no capítulo 5. Destas 100 imagens, 50 possuem infração e as outras 50 possuem apenas a presença de veículo(s). Para realizar este treinamento foram utilizadas as imagens processadas pelo algoritmo de Canny, uma vez que este gera imagens contendo as formas humanas com muito menos ruído que a do algoritmo de detecção de borda por média.

Tabela 6.4: Resultados do teste de convergência para a rede neural.

Faixa	Nº de Imagens	Nº de iterações	Limiar 1	Limiar 2
Faixa 1	100	700	0,08	0,3
Faixa 2	100	600	0,08	0,3
Faixa 3	100	900	0,08	0,3

O terceiro e último teste a ser realizado foi o de detecção da infração. Este foi feito com os valores dos pesos sinápticos obtidos nos testes de convergência da rede neural que se encontra na tabela 6.4. Em cada uma das linhas tem-se o número de amostras utilizadas no teste e a taxa de acertos. Com este teste é possível avaliar a eficiência do algoritmo. Os resultados são mostrados na tabela 6.5.

6.2 Análise dos Resultados

Uma análise da eficiência do algoritmo aqui proposto pode ser feita através dos resultados mostrados na seção 6.1 deste capítulo. Cada uma das tabelas mostradas descreve a eficiência do algoritmo em cada uma das etapas que este segue até o passo final, que é a identificação da infração caso ela exista. Assim o algoritmo realiza primeiro a identificação de veículo na faixa

Tabela 6.5: Resultados do teste de detecção de infração.

Faixa	N° de amostras	Acerto sem infração	Acerto com infração
Faixa 1	200	100%	80%
Faixa 2	200	90%	84%
Faixa 3	200	100%	78%

e depois a existência de pessoa(s) na faixa, caso a resposta para a existência de veículo seja positiva. No entanto, se for identificada a presença de veículos, sem que existam veículos na faixa, ainda resta a análise da RNA para rejeitar essa situação. Analisando as tabelas 6.1, 6.2 e 6.3 pode-se ver que o algoritmo de detecção de borda por média tem uma maior eficiência que os algoritmos de Sobel e Canny para realizar a identificação de veículos. Da tabela 6.3 pode-se ver que para todas as faixas testadas a taxa de acerto do algoritmo de detecção de veículos por média se manteve acima de 85%.

Com foi dito ao longo dessa dissertação, o tamanho da máscara influencia diretamente na eficiência da detecção de carros, pois à medida que a máscara aumenta de tamanho, as bordas dos objetos presentes na imagem tendem a ficarem mais finas, podendo não ser identificada. Já quando o tamanho da máscara diminui as bordas dos objetos presentes na imagem tendem a ficarem mais grossas e a quantidade de ruído na imagem aumenta. De forma experimental chegou-se a conclusão de que para valores de máscara de tamanho sete tem-se a melhor taxa de acerto para a menor quantidade de ruído sendo gerado. No entanto, para veículos do tipo moto, existe uma dificuldade maior em realizar esta detecção, pois a roda do mesmo não consegue ser segmentada de forma satisfatória pelo algoritmo aqui proposto e pelos outros que foram utilizados ao longo do seu desenvolvimento. Contudo, a detecção para os outros tipos de veículos se dá de forma satisfatória.

Antes de se falar na detecção da infração, tem-se que falar do processo de treinamento da rede, que é de fundamental importância para uma boa taxa de acertos na detecção da infração. Na tabela 6.4, se pode ver os resultados do processo de treinamento da RNA. Para todas as faixas em que a rede foi submetida ao processo de treinamento, a rede convergiu. Deve-se notar que o número de iterações também é influenciado pela quantidade de diferentes situações que são apresentadas à rede. Em geral, quanto mais situações diferentes, maior a quantidade de iterações para a convergência.

Com os valores de pesos sinápticos encontrados para cada uma das faixas fez-se o teste de detecção de infração. Pode-se ver o resultado deste teste na tabela 6.5. Através destas tabelas, também se pode ver que o algoritmo aqui proposto teve taxas de acerto acima de 78%, o que pode ser considerado uma boa taxa de acerto. Assim, a utilização de rede neurais do tipo perceptrons de múltiplas camadas alimentada adiante se mostrou uma solução promissora para a resolução do problema aqui proposto.

Existem também algumas considerações a serem feitas. A primeira é que a detecção de veículos depende do bom posicionamento da câmera que capta as imagens. Se esta estiver posicionada em uma altura adequada que permita a câmera pegar apenas a faixa, e ficar centralizada para enquadrar apenas a faixa, implicará então em uma boa taxa de acerto. A segunda observação é que carros que não possuam calota ou jante de uma cor diferente da cor do pneu, muito provavelmente não serão detectados pelo algoritmo proposto neste trabalho. No entanto a grande maioria dos veículos usa jantes ou calotas que permitem este destaque em relação ao pneu. Com base em tudo o que foi apresentado aqui se pode dizer que o algoritmo aqui proposto consegue realizar a detecção de forma satisfatória, tendo a sua limitação apenas com veículos do tipo motocicleta e veículos que não tenham calota ou jante que se destaque do pneu.

Capítulo 7

Conclusão e Perspectivas Futuras

Com relação à solução desenvolvida neste trabalho de dissertação chega-se a conclusão de que esta é satisfatória para a resolução do problema aqui proposto. A sua boa taxa de acerto e boa insensibilidade a variações climáticas serve como base para a afirmação feita anteriormente, sendo a sua insensibilidade as variações climáticas na extração dos dados a serem analisados pelo algoritmo de reconhecimento do padrão veículo e do padrão pessoas, a principal contribuição deste trabalho. No entanto, a solução aqui proposta tem sua limitação, não conseguindo realizar a detecção de veículos do tipo moto. A arquitetura de RNA aqui proposta para a solução da detecção de pessoas provavelmente não é a única arquitetura capaz de resolver este problema, mas foi entre as arquiteturas aqui testadas a que apresentou os melhores resultados. Outras arquiteturas e algoritmos de treinamento devem ser testados em trabalho futuros, já que existem vários modelos de RNA que não foram avaliadas neste trabalho.

Como fruto deste trabalho foi gerado um artigo que foi publicado no III Workshop de Visão Computacional WVC'2007(Alves & de C. Lima, 2007) e um programa que executa o algoritmo aqui proposto tanto através de um arquivo de imagem do tipo JPG, como de um arquivo de vídeo MPEG.

Bibliografia

- Alves, B. G. & de C. Lima, A. (2007). Detecção de infração em faixas de pedestres sem semáforos utilizando visão computacional de redes neurais, *III WVC - Workshop de Visão Computacional*.
- A.Silva, E. & Gonzaga, A. (2006). Detecção de veículos em movimento usando modelos de misturas gaussianas e rnas, *Workshop de Visão Computacional* pp. 245–248.
- Canny, J. (1986). A computational approach to edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **8**(6): 679–714.
- Cardoso, F. H., Rezende, I. & Padilha, E. (1997). *Lei 9.503 de 23 de setembro de 1997*, Diário Oficial da União.
- F.F.S.Ribeiro & de C. Lima, A. (1999). Detecção de volume de tráfego de veículos proporcionado por visão computacional via redes neurais, *IV Congresso Brasileiro de Redes Neurais* pp. 096–101.
- Gepperth, A., Edelbrunner, J. & Bucher, T. (2005). Real-time detection and classification of cars in video sequences, *Intelligent Vehicles Symposium. Proceedings. IEEE* pp. 625–631.
- Goerick, C., Noll, D. & Werner, M. (1996). Artificial neural networks in real-time car detection and tracking applications, *Pattern Recognition Lett* (17): 335–343.
- Gonzalez, R. C. & E.Woods, R. (2002). *Digital Image Processing*, Prentice Hall.
- Gose, E., Jost, S. & Johnsonbaugh, R. (1996). *Pattern Recongniton and Image Analysis*, Prentice Hall.
- Haykin, S. (2001). *Redes Neurais: princípios e prática*, Bookman.
- K. Pratt, W. (2001). *Digital Image Processing: PIKS Inside*, New York : John Wiley and Sons.
- Kass, M., Witkin, A. & Terzopoulos, D. (1988). Snakes: active contour models, *In International Journal of Computer Vision* pp. 321–331.

- Leung, B. (2004). Component-based car detection in street scene images, *Technical Report. Massachusetts Inst. of Technology* .
- Micoski, M. & M.Hemerly, E. (2006). Detecção de tráfego por vídeo: Um estudo experimental, *XVI Congresso Brasileiro de Automática* .
- Oren, M., Papageorgiou, C., Sinha, P., Osuna, E. & Poggio, T. (1997). Pedestrian detection using wavelet templates, *Computer Vision and Pattern Recognition* pp. 193–199.
- Schneiderman, H. & Kanade, T. (2000). A statistical method for 3d object detection applied to faces and cars, *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on* **1**: 746–75.
- Shashua, A., Gdalyahu, Y. & Hayun, G. (2004). Pedestrian detection for driving assistance systems: Single-frame classification and system level performance, *IEEE Intelligent Vehicle Symposium* pp. 1–6.
- Tabb, K., George, S., Adams, R. & Davey, N. (1999). Human shape recognition from snakes using neural networks, *Computational Intelligence and Multimedia Applications, 1999. ICCIMA '99. Proceedings. Third International Conference on* pp. 292–296.
- Zhao, T. & Nevatia, R. (2003). Bayesian human segmentation in crowded situations, *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* pp. 59–66.