

Mathias Silva da Rosa

**RECONHECIMENTO DE PLACAS DE
SINALIZAÇÃO DE TRÂNSITO VIA
PROCESSAMENTO DIGITAL DE IMAGEM E
APRENDIZADO DE MÁQUINA**

São José

2017

Mathias Silva da Rosa

RECONHECIMENTO DE PLACAS DE SINALIZAÇÃO DE TRÂNSITO VIA PROCESSAMENTO DIGITAL DE IMAGEM E APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Santa Catarina - Campus São José para a obtenção dos créditos na disciplina Trabalho de Conclusão do curso de Engenharia de Telecomunicações — Bacharelado.

Instituto Federal de Santa Catarina

Campus São José

Curso de Engenharia de Telecomunicações - Bacharelado

Orientadores:

Diego da Silva de Medeiros

Deise Monquellate Arndt

São José

2017

Mathias Silva da Rosa

RECONHECIMENTO DE PLACAS DE SINALIZAÇÃO DE TRÂNSITO
VIA PROCESSAMENTO DIGITAL DE IMAGEM E APRENDIZADO DE MÁ-
QUINA/ Mathias Silva da Rosa. – São José, 2017-

73 p. : il. (algumas color.) ; 30 cm.

Orientadores:

Diego da Silva de Medeiros

Deise Monquelate Arndt

Tese (Graduação) – Instituto Federal de Santa Catarina

Campus São José

Curso de Engenharia de Telecomunicações - Bacharelado, 2017.

Mathias Silva da Rosa

RECONHECIMENTO DE PLACAS DE SINALIZAÇÃO DE TRÂNSITO VIA PROCESSAMENTO DIGITAL DE IMAGEM E APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Santa Catarina - Campus São José para a obtenção dos créditos na disciplina Trabalho de Conclusão do curso de Engenharia de Telecomunicações — Bacharelado.

Trabalho aprovado. São José, 19 de fevereiro de 2018:

Prof. Diego da Silva de Medeiros
Orientador - IFSC

Prof. Deise Monquelate Arndt
Orientadora - IFSC

Prof. Ramon Mayor Martins
Membro 1 - IFSC

Prof. Marcos Moecke
Membro 2 - IFSC

São José
19 de fevereiro de 2018

Agradecimentos

Agradeço primeiramente a Deus por me guiar pelo caminho certo e conceder esta oportunidade tão rara.

Agradeço aos meus pais, Estelina e Ivandel, pelo apoio incondicional em toda esta caminhada, pelo amor e carinho insubstituíveis e pelo incentivo para superar todas as dificuldades nos momentos mais difíceis. Vocês são minha base.

Agradeço ao meu orientador Diego Medeiros e à minha coorientadora Deise Arndt pela paciência e dedicação em responder todas minhas questões, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos.

Agradeço ao instituto, seu corpo docente, direção e administração que ofereceram toda a estrutura essencial para meu crescimento profissional e pessoal.

Agradeço à minha namorada Lucilene, por enfrentar comigo todos os finais de semana de estudo e mesmo assim me dar todo o carinho e amor. Sem você essa conquista não teria o mesmo gosto.

Por fim, agradeço a todas as pessoas que de alguma forma contribuíram para a conclusão deste trabalho, o meu muitíssimo obrigado.

“A coisa boa sobre padrões é que há tantos deles para escolher!” Andrew S. Tanenbaum

Resumo

Uma das principais abordagens do reconhecimento automático de placas de sinalização de trânsito é realizar a integração entre técnicas de processamento digital de imagens e métodos de aprendizado de máquina objetivando determinar as informações importantes apresentadas por cada imagem capturada em tempo real. Estas informações podem ser utilizadas como fatores que podem determinar o comportamento de mecanismos como os Sistemas Autônomos de Condução (ADS) e Sistemas Avançados de Apoio ao Motorista (ADAS). Este trabalho tem como propósito apresentar o estudo sobre um sistema de reconhecimento automático de placas de sinalização de trânsito dividido em cinco etapas: aquisição, pré-processamento com o uso do modelo HSV e técnicas como limiarização, morfologia e detectores de borda; segmentação através do uso da transformada Hough generalizada; descrição através do uso do método Histograma de Gradientes Orientados (HOG) e por fim, classificação com o uso de Máquinas de Vetores de Suporte (SVM's). Primeiramente o trabalho proposto aborda o desenvolvimento dos processos de treinamento e teste implementados através da plataforma MATLAB[®] e a utilização de imagens estáticas contendo placas de sinalização de trânsito. Em seguida, uma câmera é acoplada ao painel de um carro em movimento obtendo fotos sequenciais de placas de sinalização através do trajeto percorrido. O algoritmo proposto alcançou taxas de acerto superiores à 70% para o reconhecimento das placas de sinalizações para primeira etapa de testes. Para a segunda etapa, apesar das características diferentes apresentadas pelas placas de sinalização, as taxas de acertos alcançaram 90% para o reconhecimento das placas.

Palavras-chaves: HOG. SVM.

Abstract

One of the main approaches for the automatic recognition of traffic signs is to realize the integration between techniques of digital image processing and machine learning to identify the important informations presented by each image captured in real-time. These informations can be used as factors to determine the behavior of mechanisms like Autonomous Driver System (ADS) or Advanced Driver Assistant System (ADAS). This job has as objective to introduce a study of automatic traffic sign recognition system divided into five steps: acquisition, preprocessing with the use of HSV model and techniques such as limiarization, morphology and edge detection; segmentation with the use of generalized Hough transform; description with the use of Histograms of Oriented Gradients (HOG) and finally, classification with the use of Support Vectors Machine (SVM). First, the proposed work approaches the development of the processes of training and test implemented with MATLAB[®] platform and the use of static images containing traffic signs. Then, a camera is coupled to the panel of a moving car, getting sequential pictures of traffic signs through the course. The proposed algorithm reached hit rate over 70% for the recognition of traffic signs in the first part of tests. In the second part, despite of traffic signs which had some different features, the hit rate reached 90% for the recognition of traffic signs.

Key-words: HOG. SVM.

Lista de ilustrações

| | |
|---|----|
| Figura 1.1 –Distribuição mundial de mortes por acidentes de trânsito. | 15 |
| Figura 1.2 –Sistema autônomo de transporte nos modelos automotivos da Tesla. . . | 16 |
| Figura 2.1 –Imagem digital em escala de cinza representada através de um plano cartesiano. | 21 |
| Figura 2.2 –Principais etapas de uma sistema de visão computacional. | 22 |
| Figura 2.3 –Modelo RGB representado por um cubo 3D. | 24 |
| Figura 2.4 –Modelo HSV representado por um cone 3D. | 25 |
| Figura 2.5 –Limiarização de uma imagem em escala de cinza (a) Imagem original; (b) Imagem limiarizada. | 27 |
| Figura 2.6 –Limiarização de uma imagem colorida (a) Imagem original; (b) Imagem limiarizada. | 27 |
| Figura 2.7 –Processo de erosão na primeira linha e dilatação na segunda linha. . . . | 28 |
| Figura 2.8 –Método de Sobel em uma imagem em escala de cinza (a) Imagem original; (b) Imagem de bordas. | 29 |
| Figura 2.9 –Filtros Gaussianos utilizados pelo método de Canny. | 30 |
| Figura 2.10 –Método de Canny aplicado a uma imagem em escala de cinza (a) Imagem original; (b) Imagem de bordas. | 30 |
| Figura 2.11 –Detecção de círculos e triângulos com a transformada Hough generalizada. | 32 |
| Figura 2.12 –Etapas do método HOG. | 35 |
| Figura 2.13 –Descriptor HOG de uma imagem em escala de cinza. | 36 |
| Figura 2.14 –Hierarquia do aprendizado de máquina. | 38 |
| Figura 2.15 –Classificadores segundo três hipóteses. | 39 |
| Figura 2.16 –Classificador linear à esquerda e não linear à direita. | 42 |
| Figura 2.17 –Cálculo da margem máxima. | 43 |
| Figura 2.18 –(a) Dados não separáveis linearmente em um plano bidimensional (b) Dados não separáveis linearmente em um plano tridimensional | 43 |
| Figura 3.1 –Diagrama da arquitetura geral | 44 |
| Figura 3.2 –Amostras de treinamento GTSDB | 45 |
| Figura 3.3 –Amostra de teste GTSDB | 46 |
| Figura 3.4 –Diagrama do fluxo da etapa de pré-processamento. | 47 |
| Figura 3.5 –Processo de limiarização da matiz (a) Imagem original; (b) Imagem binária; | 48 |
| Figura 3.6 –Histograma da saturação. | 48 |
| Figura 3.7 –Histograma da saturação equalizado. | 49 |

| | |
|--|----|
| Figura 3.8 –Processo de limiarização da saturação (a) Imagem original; (b) Imagem binária; | 49 |
| Figura 3.9 –Multiplicação das limiarizações da matiz e saturação. | 50 |
| Figura 3.10 –Processo de enchimento da imagem. | 50 |
| Figura 3.11 –Processo de erosão da imagem. | 51 |
| Figura 3.12 –Remoção das áreas. | 52 |
| Figura 3.13 –Processo de dilatação da imagem. | 53 |
| Figura 3.14 –Aplicação do método de Canny para detecção de bordas. | 53 |
| Figura 3.15 –Remoção das bordas através do ponto de massa. | 54 |
| Figura 3.16 –Segmentação da placa de sinalização. | 55 |
| Figura 3.17 –Descriptor HOG sobre a imagem analisada. | 55 |
| Figura 3.19 –Suporte desenvolvido para acoplar a câmera ao painel do veículo. | 57 |
| Figura 3.20 –Câmera acoplada ao painel frontal do veículo. | 57 |
| Figura 3.21 –Amostra das imagens obtidas pela câmera Intelbras VIP 4120. | 57 |
| Figura 3.22 –Resultados da limiarização da matiz. | 59 |
| Figura 3.23 –Resultados da limiarização da saturação. | 60 |

Lista de tabelas

| | |
|---|----|
| Tabela 3.1 –Parâmetros de definição das SVM's | 56 |
| Tabela 3.2 –Parâmetros de definição do algoritmo | 58 |
| Tabela 3.3 –Inclinação dos limiars da transformada Hough generalizada. | 61 |
| Tabela 3.4 –Resultados da etapa de segmentação. | 62 |
| Tabela 3.5 –Resultados da etapa de classificação | 63 |
| Tabela 3.6 –Resultados do algoritmo aplicado a placas de sinalização brasileiras. . . | 64 |
| Tabela A.1 –Configuração do sistema. | 73 |
| Tabela A.2 –Quantidade de imagens de treinamento | 73 |

Lista de abreviaturas e siglas

| | |
|------|--|
| ADAS | Advanced Driver Assitant System (Sistema Avançado de Apoio ao Motorista) |
| RI | Região de Interesse |
| HOG | Histogram of Oriented Gradients (Histogramas de Gradientes Orientados) |
| SVM | <i>Support Vector Machines</i> (Máquinas de Vetores de Suporte) |
| HSV | Hue Saturation and Value (Matiz, Saturação e Valor) |
| VC | Vapnik–Chervonenkis |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 15 |
| 1.1 | Metodologia | 18 |
| 1.2 | Estrutura do Trabalho | 19 |
| 2 | Fundamentação Teórica | 20 |
| 2.1 | Imagem Digital | 20 |
| 2.2 | Sistema de Visão Computacional | 21 |
| 2.3 | Pré-processamento | 23 |
| 2.3.1 | Modelo de Cores | 23 |
| 2.3.2 | Limiarização | 26 |
| 2.3.3 | Morfologia | 27 |
| 2.3.4 | Detectores de Borda | 28 |
| 2.4 | Segmentação | 30 |
| 2.4.1 | Transformada Hough | 31 |
| 2.4.2 | Transformada Hough Generalizada | 32 |
| 2.5 | Descrição | 33 |
| 2.5.1 | Gradiente | 34 |
| 2.5.2 | Histograma de Gradientes Orientados (HOG) | 35 |
| 2.6 | Classificação | 36 |
| 2.6.1 | Aprendizado de Máquina | 36 |
| 2.6.2 | Teoria do Aprendizado Estatístico | 38 |
| 2.6.3 | Máquina de Vetores de Suporte | 41 |
| 3 | Desenvolvimento | 44 |
| 3.1 | Arquitetura Geral | 44 |
| 3.2 | Banco de Imagens | 45 |
| 3.3 | Implementação | 46 |
| 3.3.1 | Pré-processamento | 47 |
| 3.3.2 | Segmentação | 53 |
| 3.3.3 | Descrição | 55 |
| 3.3.4 | Classificação | 56 |
| 3.4 | Cenário de Testes | 56 |
| 3.5 | Resultados | 58 |
| 3.5.1 | Pré-processamento | 58 |
| 3.5.2 | Segmentação | 61 |
| 3.5.3 | Classificação | 62 |

| | |
|--|---------------|
| 3.5.4 Placas de Sinalização Brasileiras | 63 |
| Conclusão | 65 |
| Referências | 68 |
| Apêndices | 72 |
| APÊNDICE A Configuração Utilizada para a Implementação do Sistema . . | 73 |

1 Introdução

A sinalização de trânsito exerce um papel fundamental de auxiliar na manutenção da fluidez das vias de trânsito e de proporcionar uma maior segurança dos condutores e dos pedestres que nelas circulam. Para isso, é necessário o uso de uma linguagem visual, constituída de diversos tipos de sinais e dispositivos que transmitam informações das quais são utilizadas para orientar os condutores e advertir sobre possíveis situações de perigo (CONTRAN, 2007, p.21).

Apesar do uso consolidado da sinalização de trânsito, o fator humano influencia diretamente na eficácia deste tipo de mecanismo e o torna, em muitos casos, uma solução incompleta. O condutor, ao não dar a devida atenção à sinalização de trânsito, pode provocar diversas situações de risco, como velocidade fora do limite permitido, cruzamentos indevidos, uso incorreto das vias, etc. Em consequência, são os inúmeros acidentes de trânsito que acarretam em mortes, invalidez, pagamento de indenizações entre outras situações que em sua maioria poderiam ser evitadas (CASTRO; HORBERRY, 2004, p.4).

Segundo WHO (2017a), é estimado que os acidentes de trânsito ocupam o nono lugar nas causas que mais matam pessoas independentemente da faixa etária e o primeiro lugar nas causas que mais matam pessoas entre 15 e 29 anos. Conforme a Figura 1.1, todos os anos morrem 17,5 pessoas a cada 100 mil indivíduos, vítimas de acidentes de trânsito em todo o mundo, onde também segundo WHO (2017a) quase metade deste número (49%) é constituído por pedestres, ciclistas e motociclistas. Além disso, estima-se que 50 milhões de pessoas sofrem lesões ou invalidez relacionadas aos acidentes de trânsito a cada ano.

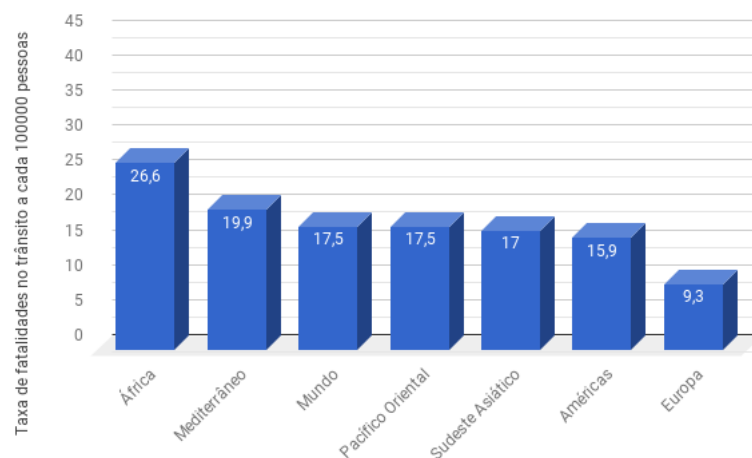


Figura 1.1: Distribuição mundial de mortes por acidentes de trânsito.

Adaptado de (WHO, 2017b, p.2)

Além do considerável dano causado às vítimas e suas famílias, os acidentes de trânsito implicam em uma considerável perda socioeconômica. Segundo dados da [Seguradora Líder \(2015\)](#), em 2015 aproximadamente 600 mil indenizações foram pagas devido aos acidentes de trânsito em território brasileiro, que em sua maioria foram resultados de casos de invalidez permanente e despesas médicas.

Na tentativa de diminuir estes números, diversas instituições e empresas vêm investindo no desenvolvimento de mecanismos que ampliem os sentidos do motorista e garantem uma maior segurança no trânsito. Para isso, muitos sistemas utilizam a informação transmitida por sinalizações de trânsito como um fator determinante de comportamento. Entre os principais sistemas que possuem esta abordagem estão os Sistemas ADS e ADAS ([LIN et al., 2016](#), p.2247).

Os modelos automotivos Model S, Model X e Model 3 da [Tesla \(2017\)](#), como é possível observar através da [Figura 1.2](#), possuem sistemas autônomos de transporte que fazem do reconhecimento de placas de sinalização de trânsito um relevante fator para determinar o movimento do carro. Na [Figura 1.2](#), a placa de advertência de curva à esquerda é identificada e destacada no sistema através da cor roxa, possibilitando ao sistema prever que o veículo deverá fazer este movimento no caminho a ser percorrido.



Figura 1.2: Sistema autônomo de transporte nos modelos automotivos da Tesla.

Fonte: ([TESLA, 2017](#))

Para determinar condições da pista ou situações potencialmente perigosas, os sistemas de apoio, como os desenvolvidos por ([AUDI, 2017](#)), ([BOSCH, 2017](#)) e ([HYUNDAI, 2017](#)) utilizam as placas de sinalização com o objetivo de alertar o motorista e mantê-lo atento a estas condições. Câmeras acopladas à parte dianteira do veículo, em conjunto com um sistema computacional integrado, fazem o reconhecimento da placa e apresentam as informações para o motorista através de alertas em periféricos de saída como monitores e alto-falantes.

Em diversos trabalhos, as técnicas utilizadas para a concepção destes sistemas se baseiam em métodos inspirados em mecanismos biológicos como a visão e o próprio reconhecimento de padrões do cérebro humano. Mais especificamente, os métodos envolvem o processamento digital de imagens e o uso de algoritmos de aprendizado de máquina que apresentam boa performance ao serem utilizadas em sistemas deste tipo (GONZALEZ; WOODS, 2010, p.22).

Chen, Yang e Kong (2011) propõem um sistema de reconhecimento automático de placas de sinalização de trânsito baseado no uso de métodos de limiarização de cores e Transformada Hough para etapa de detecção, no uso de momentos Pseudo-Zernike para a etapa de descrição e no uso de SVM para o processo de classificação. Para a implementação do sistema foi utilizada a biblioteca OpenCV para C++, muito utilizada em aplicações no campo da visão computacional. Para a execução dos testes foram utilizadas 2600 imagens coletadas em diversas condições de clima e angulação. Os testes apresentaram taxas de acerto superiores à 90% no reconhecimento de todas as classes.

Măriuț et al. (2011) propõem a utilização das propriedades do espaço HSV (Matiz, Saturação e Valor) juntamente com a correlação das imagens de entrada com um banco predefinido para a etapa de detecção. A descrição das regiões de interesse (RI) é feita através da aplicação de filtros de Gabor, enquanto que o cálculo da mínima distância Euclidiana é utilizado para a etapa de classificação. Com o objetivo de analisar a influência da iluminação sobre o processo de reconhecimento, os níveis de brilho foram alterados resultando em um conjunto maior de imagens de entrada. Os resultados apresentaram taxas de erros elevados, principalmente em placas com detalhes complexos, evidenciando a fragilidade do sistema em condições adversas.

Greenhalgh e Mirmehdi (2012) propõem a utilização de métodos de limiarização de intensidade e a definição das Regiões de Máxima Estabilidade (MSER) para a etapa de detecção, HOG para a etapa de descrição e uma cascata de SVM's para a etapa de classificação. O sistema foi testado em uma faixa específica de velocidades do veículo sobre uma variedade de luz e condições de clima. Apesar da grande variedade de fatores, foram obtidas taxas de acerto de 89.2% para placas de fundo branco e 92.1% para placas de fundo colorido.

Gao et al. (2014) (2014) propõem uma abordagem focada na extração de características das RI's, especialmente na otimização desta etapa utilizando o algoritmo *Scale-Invariant Feature Transform* (SIFT), que possui como principal característica a capacidade de invariância a mudanças de iluminação, ruído, rotação e escala; em conjunto com o uso do Procedimento de Análise de Componentes Principais (PCA) para uma melhor representação dos dados. Na etapa de classificação é utilizada SVM. O enfoque dos resultados foi a latência do sistema, onde o uso do PCA proporcionou a redução de aproximadamente 50% da latência no processo de descrição.

Romdhane, Mliki e Hammami (2016) propõem um sistema de reconhecimento automático de placas de sinalização baseado no uso das técnicas HOG e SVM para descrição e classificação, respectivamente. Para a etapa de detecção, um sistema de busca localiza a região onde se encontra a placa de sinalização utilizando como parâmetro as faixas de delimitação de via, eliminando grande parte da imagem. Métodos de limiarização de intensidade também são aplicados e classificadores simples segmentam a RI das demais regiões. Os resultados obtidos apresentaram taxas de acerto superiores à 80% neste caso.

A partir da leitura dos trabalhos acima, é possível observar a diversificação dos algoritmos aplicados no reconhecimento automático de placas de sinalização em diferentes tipos de ambiente. Dentro deste contexto, este trabalho busca o estudo e o desenvolvimento computacional de um sistema que procura reconhecer automaticamente placas de sinalização de trânsito via técnicas de processamento de imagem e aprendizado de máquina. Para isso, é proposto um rearranjo das principais técnicas analisadas e o estudo dos resultados deste algoritmo diante de um conjunto de dados de entrada obtidos através de um ambiente real.

1.1 Metodologia

Em muitos dos trabalhos estudados, os algoritmos para o reconhecimento automático de placas de sinalização trânsito são divididos em etapas de aquisição, pré-processamento, segmentação, descrição e classificação. Por isso, neste trabalho, propõe-se utilizar uma série de mecanismos para cada etapa em conformidade com literatura analisada e com as técnicas apresentadas mais recentemente.

Para a etapa de pré-processamento, são utilizados para o realce das regiões de interesse, métodos de limiarização de imagem no modelo de cores HSV, em conjunto de operações morfológicas e detectores de borda. Na etapa de segmentação, a transformada Hough generalizada é utilizada para localizar o centro da RI e segmentá-la através da aproximação de área com templates pré-definidos. Para a etapa de descrição são utilizados os descritores obtidos através do método HOG. Por fim, para a etapa de classificação, SVM's são aplicadas em paralelo aos descritores e apresentam como resultado a identificação de cada placa de sinalização de entrada.

Os códigos relacionados as etapas de pré-processamento, segmentação e descrição são implementados através do uso do *software* MATLAB[®]. A etapa de classificação também é realizada através desta plataforma, porém utilizando os métodos encontrados no pacote *Support Vector Machine Classification* encontrado no *toolbox* de Estatística e Classificação das recentes versões do MATLAB[®].

Os procedimentos de validação são realizados através de uma máquina com Intel(R) Core(TM) i7-4500U 1.80GHz e 8GB de RAM. Para o treinamento do sistema, são

utilizadas imagens estáticas contendo placas de sinalização do sistema alemão de trânsito disponibilizadas por [Houben et al.\(2013\)](#). Deste banco de imagens são utilizadas para treinamento e testes, imagens contendo placas de sinalização de trânsito pertencentes a 11 Classes, das quais possuem características como borda vermelha e formato circular.

Por fim, são analisadas as taxas de acerto do sistema, decorrente de testes realizados em imagens obtidas através de uma câmera do modelo Intelbras VIP 4120 acoplada ao painel dianteiro de um veículo em movimento. Estes testes são realizados percorrendo-se vias públicas que possuem placas do sistema brasileiro de trânsito semelhantes as placas analisadas do processo anterior.

1.2 Estrutura do Trabalho

O presente trabalho é estruturado em 4 capítulos. O primeiro capítulo referente à introdução, apresenta uma breve explicação sobre a proposta do trabalho, as justificativas, os conceitos relacionados e a metodologia envolvida. O capítulo de fundamentação teórica explana com mais detalhes os conceitos relacionados a um sistema de visão computacional, como Aquisição da Imagem, Técnicas de Pré-processamento, Segmentação, Descrição e Classificação. O capítulo de desenvolvimento, descreve detalhadamente a implementação do sistema, a execução dos experimentos e os resultados obtidos. Por fim, o último capítulo apresenta as conclusões obtidas e as sugestões para os próximos trabalhos.

2 Fundamentação Teórica

Este capítulo aborda a teoria utilizada para concepção do trabalho, bem como os processos necessários para o desenvolvimento de um sistema de reconhecimento automático de placas de sinalização de trânsito. A seção 2.1 aborda o conceito de imagem digital, principal dado de entrada do sistema. A seguir, a seção 2.2 apresenta um sistema de visão computacional e seus componentes, introduzindo a etapa de pré-processamento, segmentação, descrição e classificação, apresentadas pelas seções 2.3, 2.4, 2.5 e 2.6, respectivamente.

2.1 Imagem Digital

Há várias maneiras de definir o conceito de imagem dependendo do foco do estudo ou do cenário onde a imagem é analisada. Segundo [Silveira e Rita \(2010\)](#) uma imagem, em termos de linguagem e comunicação, é a representação de um objeto através de seus elementos gráficos que destacam certas características deste objeto, como a cor, o tamanho, o contraste, etc.

No campo da visão computacional, uma imagem é interpretada de maneira que seus elementos gráficos são representações numéricas e discretizadas, passando a ser denominada como imagem digital ([SOLOMON; BRECKON, 2011](#), p.1). A conversão de uma imagem real em uma imagem digitalizada é obtida através de sensores de aquisição de imagens que adquirem a imagem digital através da absorção da energia refletida em forma de luz do objeto ou cena em questão ([GONZALEZ; WOODS, 2010](#), p.29).

Uma imagem digital é descrita em coordenadas espaciais conforme é apresentado pela [Figura 2.1](#). A função discreta $I(x,y)$ representa a intensidade (imagens em escala de cinza) ou a cor (imagens coloridas) em pontos fixos (*pixels*) do plano cartesiano bidimensional, definidos pelo número de linhas x e o número de colunas y ($x = 1,2,\dots,M$; $y = 1,2,\dots,N$). É possível observar também através da [Figura 2.1](#) a aplicação do plano cartesiano com eixo vertical invertido para representar uma imagem digital de duas dimensões ([HORNBERG, 2007](#), p.485).

Em termos de hardware, a intensidade ou a cor de uma imagem é discretizada em níveis, que geralmente são representados por valores definidos entre 0 e 2^{k-1} , sendo k um número inteiro. O processo de discretização dos valores de intensidade ou dos valores de cor é denominado de quantização e é este processo que permite que uma imagem possa ser processada na forma digital ([VERNON, 1991](#), p.28).

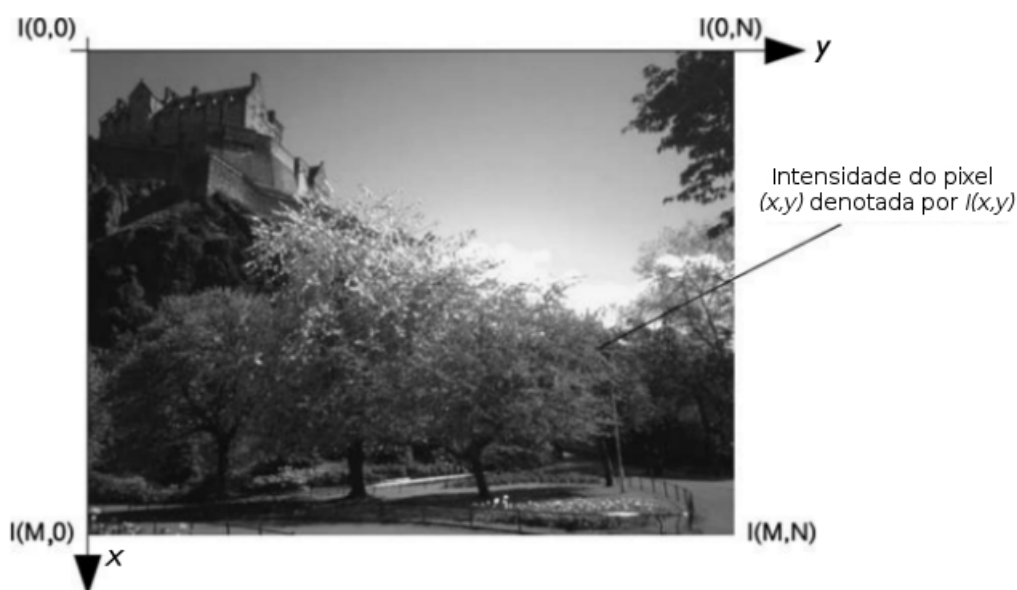


Figura 2.1: Imagem digital em escala de cinza representada através de um plano cartesiano.

Adaptado de (SOLOMON; BRECKON, 2011, p.2)

2.2 Sistema de Visão Computacional

O tratamento de dados provenientes de imagens é denominado de processamento digital de imagens e é uma das principais etapas de um sistema de visão computacional, uma vez que este permite a concepção de um grande número de aplicações, principalmente nas áreas de medicina, comunicação, transporte, meteorologia, espacial, etc (VERNON, 1991, p.2).

O processamento digital de imagens pode ser dividido em três etapas: aquisição de imagens, pré-processamento e segmentação; que são apresentadas no topo da Figura 2.2. Esta figura apresenta os componentes básicos de um sistema de visão computacional, onde as etapas referentes ao processamento digital são precedentes às etapas de descrição e classificação (FILHO; NETO, 1999, p.1).

Para realizar o processamento digital de uma imagem, é necessário primeiramente obtê-las no formato digital (GONZALEZ; WOODS, 2010, p.29). Em muitos sistemas, a captura de imagens e a digitalização destas é de responsabilidade da etapa de aquisição, da qual é comumente executada através de sensores de aquisição de imagens (câmeras CCD, scanners, etc) que convertem a luz emitida pelos objetos em sinais digitais (JAHNE, 2005, p.23).

Em muitos casos, a imagem obtida do processo de aquisição pode apresentar imperfeições como *pixels* ruidosos, contraste ou brilho inadequado, ou ainda muitas informações desnecessárias (HORNBERG, 2007, p.516). Uma das formas de evitar que esses fatores influenciam nos resultados do sistema é aplicar uma etapa de pré-processamento respon-

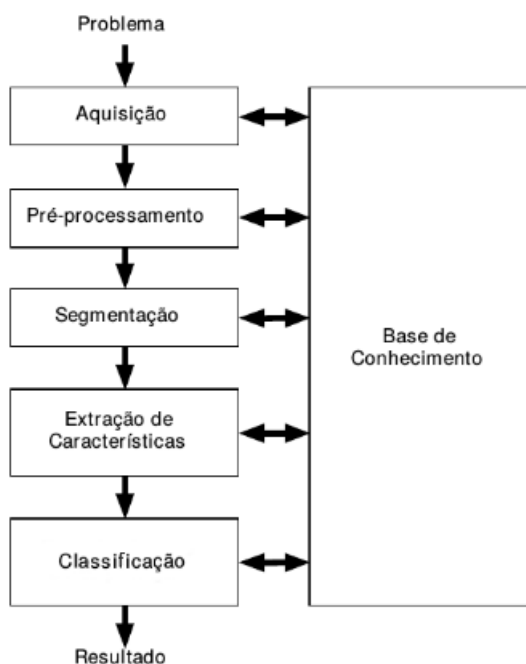


Figura 2.2: Principais etapas de uma sistema de visão computacional.

Adaptado de (FILHO; NETO, 1999, p.9)

sável pelo realce e aprimoramento da imagem ou objeto (GONZALEZ; WOODS, 2010, p.15).

O objetivo da etapa de segmentação é separar as regiões de interesse das demais regiões da imagem, tarefa essa que pode apresentar alto nível de complexidade dependendo das técnicas empregadas e do nível de precisão exigido (HORNBERG, 2007, p.545). Várias técnicas foram desenvolvidas para a segmentação de objetos em uma imagem, porém este processo não possui uma solução geral. Em muitos casos é necessário a junção de várias técnicas para se atingir a precisão necessária para o domínio do problema (JAHNE, 2005, p.449).

A etapa de extração das características é responsável por formar os descritores que permitem caracterizar as regiões segmentadas adequadamente para o posterior processamento computacional, em outras palavras, este processo reduz significativamente a massa de dados, mensurando certas propriedades da região, antes de serem enviados para a etapa de classificação (DOUGHERTY, 2009, p.340).

A última etapa envolve metodologias de classificação de padrões que, basicamente, separam as regiões descritas em diferentes classes (TANIMOTO, 2012, p.393). Técnicas de classificação possuem três principais modelos: estatístico, estrutural e híbrido. O modelo estatístico lida com objetos e padrões que possuem base estatística e são descritos por características quantitativas como comprimento, área e textura. O modelo estrutural lida com objetos com estruturas qualitativas ou relações sintáticas inerentes ao objeto. Em

muitos casos, as duas abordagens são combinadas em um modelo híbrido, onde técnicas de aprendizado de máquina são frequentemente utilizadas (DOUGHERTY, 2009, p.340).

É possível observar ainda através da [Figura 2.2](#) que cada etapa do processo necessita de um conhecimento prévio sobre o problema a ser resolvido, armazenado em uma base de conhecimento, cujo tamanho e complexidade podem variar em relação ao contexto da aplicação. Outro papel importante da base de conhecimento é apresentar entre as etapas um caminho de realimentação, onde informações obtidas são reutilizadas com o objetivo de realizar uma análise mais criteriosa dos dados (FILHO; NETO, 1999, p.10).

2.3 Pré-processamento

As características de uma imagem ou de um objeto contido em uma imagem podem ser destacados através da manipulação de seus elementos gráficos. Dois dos principais elementos que são utilizados para processos de realce são a cor e o formato (GONZALEZ; WOODS, 2010). Nas próximas subseções são explanados a representação de imagens digitais em modelos de cores e métodos de realce utilizando a cor e o formato dos objetos da imagem dentro do contexto deste trabalho.

2.3.1 Modelo de Cores

Em várias circunstâncias, manipular uma imagem colorida ou adicionar cores a uma imagem proporciona diversas vantagens, uma vez que a cor é um poderoso descritor que pode simplificar a identificação dos objetos e sua extração (GONZALEZ; WOODS, 2010, p.259). Para representar todas as cores em uma imagem, é necessário utilizar a combinação de um ou mais canais primários, onde a proporção de cada canal especifica o tipo das cores secundárias geradas (SOLOMON; BRECKON, 2011, p.9).

Para facilitar as especificações das cores de uma forma padronizada, foram criados os modelos de cores (também denominados espaço de cores ou sistema de cores). Um modelo de cores especifica um espaço de coordenadas na qual a cor é representada através de um único ponto deste espaço (DOUGHERTY, 2009, p.40). Outra vantagem do uso de um modelo de cores, é o seu fácil tratamento computacional, uma vez que as cores podem ser representadas numericamente e assim, facilmente convertidas para a linguagem de máquina (SOLOMON; BRECKON, 2011, p.10).

Entre os modelos de cores mais utilizados estão os modelos RGB, HSV, HSI, LAB, YCbCr, CMYK, etc, cada uma com suas respectivas vantagens e desvantagens. O modelo RGB (*red*, *green*, *blue*), muito utilizado para aquisição e exibição de imagens digitais, é formado pela combinação proporcional das três cores primárias vermelho, verde e azul. Computacionalmente, cada elemento é representado através de um conjunto de bits de

tamanho variável conforme a profundidade das cores utilizadas. (DOUGHERTY, 2009, p.41).

Uma imagem representada no modelo RGB consiste de três componentes de imagem, uma para cada cor primária e com isso o número de possíveis cores do modelo RGB é a multiplicação dos bytes de cada subimagem, resultando em um número de aproximadamente 16 milhões de cores (HORNBERG, 2007, p.485). A Figura 2.3 demonstra o cubo tridimensional que representa graficamente o modelo RGB.

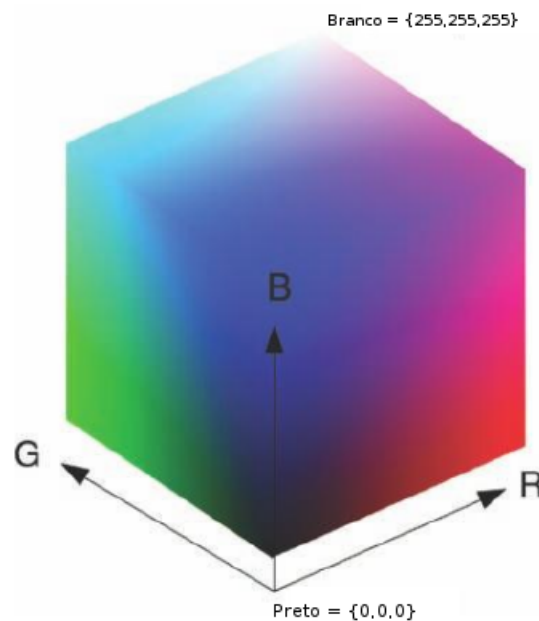


Figura 2.3: Modelo RGB representado por um cubo 3D.

Fonte: (SOLOMON; BRECKON, 2011, p.330)

Apesar de seu uso consolidado, a aplicação do espaço RGB em métodos de processamento digital implica, em muitas situações, resultados inadequados. Para alguns cenários, como por exemplo o realce de borda, é mais eficiente trabalhar em um espaço que separa a intensidade dos *pixels* da informação de crominância (DOUGHERTY, 2009, p.41). Neste contexto, muitas outras alternativas foram criadas, entre elas se encontra o modelo HSV (Matiz, Saturação e Valor) (SOLOMON; BRECKON, 2011, p.12).

O matiz é um atributo que descreve uma cor pura, como por exemplo as cores “azul”, “vermelho”, etc; ao passo que a saturação apresenta a informação da quantidade de cor que um *pixel* possui. Por fim, o valor é a quantidade de brilho que uma cor possui e é o elemento utilizado para realizar a separação entre as informações de intensidade e cor (GONZALEZ; WOODS, 2010, p.268).

A Figura 2.4 apresenta o cone tridimensional que representa graficamente o modelo HSV. Numericamente, os elementos do modelo HSV são definidos através do eixo vertical de valor, dos valores ao longo do comprimento do vetor de saturação e dos valores de

angulação obtidos através da rotação do matiz (TANIMOTO, 2012, p.78).

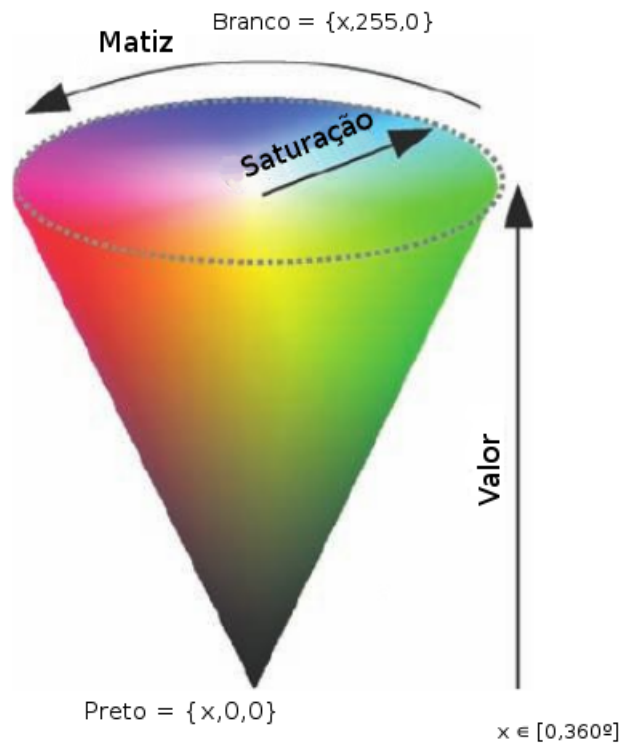


Figura 2.4: Modelo HSV representado por um cone 3D.

Fonte: (SOLOMON; BRECKON, 2011, p.231)

Quando os processos de aquisição de imagens coloridas e processamento digital são feitos em modelos de cores diferentes, é necessário realizar a conversão entres modelos. A conversão dos valores do modelo RGB para o modelo HSV é realizado através das equações 2.1, 2.2 e 2.3 (SHAIK et al., 2015, p.43).

$$H = \arccos \frac{\frac{1}{2}(2R - G - B)}{\sqrt{(R - G)^2 - (R - B)(G - B)}} \quad (2.1)$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (2.2)$$

$$V = \max(R, G, B) \quad (2.3)$$

O espaço HSV é muito utilizado no realce de regiões baseado na cor. Ao separar uma porção do elemento matiz como uma faixa de cores de interesse, é possível detectar objetos dentro da imagem que possuem estas cores (SOLOMON; BRECKON, 2011, p.14). A Figura 2.5 apresenta um possível processo de realce através do modelo HSV.

2.3.2 Limiarização

Em muitas aplicações, para destacar uma região de interesse contida em uma imagem, é conveniente separar esta das demais regiões da imagem. A limiarização provê um dos meios mais simples para separar digitalmente as informações de interesse baseado nos diferentes níveis de intensidade ou cor de cada *pixel* da imagem (DOUGHERTY, 2009, p.137).

A entrada do processo de limiarização é uma imagem em escala de cinza ou colorida e a saída é uma imagem binária contendo apenas o valor 0 para os *pixels* do fundo da imagem e o valor 1 para os *pixels* do objeto em questão. Usualmente, os bits 0's correspondem à informação não usual, enquanto que os bits 1's representam as informações da região de interesse (SOLOMON; BRECKON, 2011, p.56).

Em implementações simples, o processo de limiarização envolve parâmetros singulares conhecidos como limiares. Em cada *pixel* da imagem o valor de intensidade, por exemplo, é comparado com o limiar. O *pixel* que possui nível de intensidade superior ao limiar, tem seu valor convertido para 1, enquanto que o *pixel* que possui nível de intensidade inferior ao limiar, tem seu valor convertido para 0 (CHAKI; SHAIKH; SAEED, 2014, p.1). Resumidamente, o processo de limiarização simples é definido pela Equação 2.4.

$$\begin{aligned} B(x, y) &= 1, & \text{se } I(x, y) \geq T \\ &= 0, & \text{caso contrário.} \end{aligned} \tag{2.4}$$

Uma imagem ao ser limiarizada, possui somente valores dentro do conjunto $A = [0,1]$ e por isso permite que cada *pixel* seja representado por um único bit. Isto faz com que posteriores processos de segmentação e descrição necessitem de um menor recurso de processamento (CHAKI; SHAIKH; SAEED, 2014, p.2).

A complexidade de um processo de limiarização depende fortemente da obtenção dos limiares. Existem duas abordagens: utilizar um limiar global, obtido através dos valores de todos os *pixels* da imagem ou utilizar diversos limiares locais cada um correspondente à uma região da imagem. A escolha de qual abordagem utilizar depende da precisão necessária entre a região de interesse e o segundo plano da imagem (KRIG, 2014, p.77). A Figura 2.5 apresenta uma imagem binária resultante do processo de limiarização utilizando um limiar global.

Semelhantemente, o processo de binarização pode ser aplicada nas propriedades de uma imagem colorida, sendo também denominado de fatiamento de cores. Um limiar pode ser calculado para definir quais propriedades de cor como cores específicas, matiz, saturação, etc; devem ser destacados da imagem. Para isso, aplica-se este processo sobre os valores de cada componente do modelo de cores utilizado para representar a imagem



Figura 2.5: Limiarização de uma imagem em escala de cinza (a) Imagem original; (b) Imagem limiarizada.

Fonte: Elaborado pelo Autor.

(GONZALEZ; WOODS, 2010, p. 284). A Figura 2.6 apresenta o processo de limiarização em uma imagem colorida, representada no modelo RGB, do qual é destaca a cor vermelha.



Figura 2.6: Limiarização de uma imagem colorida (a) Imagem original; (b) Imagem limiarizada.

Fonte: Elaborado pelo Autor.

2.3.3 Morfologia

A morfologia, na área de visão computacional, trata de uma variedade de operações que são aplicadas à imagem durante a etapa de pré-processamento (GOYAL, 2011, o. 161). Para isso, um determinado conjunto de *pixels* relevantes para a aplicação em questão sofrem transformações através do uso de outros conjuntos binários bem definidos, chamados de elementos estruturantes. Portanto, a base da morfologia é a teoria matemática dos conjuntos (FILHO; NETO, 1999, p. 1390). Os elementos estruturantes são filtros binários com diversos formatos e que são aplicados em todos os *pixels*.

O processo morfológico de erosão permite transformações que apresentam resultados como a diminuição de objetos, quebra de ligações, aumento de orifícios na imagem e a remoção de ruído, conforme é apresentado na parte superior da Figura 2.7. Os *pixels* da imagem que se encontram dentro do conjunto definido pelo tamanho do elemento estruturante são aplicados à Equação 2.5, onde A representa o valor binário do *pixels* da imagem e B representa o valor binário do elemento estruturante, ambos na mesma posição. Os *pixels* que possuem valor 1 e que possuem a posição do elemento estruturante correspondente também com valor 1, são abstraídos da imagem, ou seja, convertidos para o valor 0 (SOLOMON; BRECKON, 2011, p. 200).



Figura 2.7: Processo de erosão na primeira linha e dilatação na segunda linha.

Fonte: (SOLOMON; BRECKON, 2011, p.200)

$$Y = A.B' \quad (2.5)$$

Por outro lado, o processo morfológico de dilatação apresenta o resultado inverso do processo de erosão. Este permite transformações que apresentam resultados como a evidenciação de objetos, remoção de serrilhamentos e remoção de descontinuidades, conforme parte inferior da [Figura 2.7](#). No caso do processo de dilatação, os *pixels* que se encontram dentro do conjunto definido pelo tamanho do elemento estruturante são aplicados à [Equação 2.6](#). Os *pixels* que possuem valor 0 e que possuem a posição do elemento estruturante correspondente com valor 1, são convertidos para o valor 1, retornando ao primeiro plano da imagem (GOYAL, 2011, p.162).

$$Y = A'.B \quad (2.6)$$

2.3.4 Detectores de Borda

É possível definir uma borda de uma imagem em níveis de cinza como a fronteira entre duas regiões cujos níveis de intensidade predominantes são razoavelmente diferentes. Analogamente, pode-se definir uma borda em uma imagem colorida onde há alterações abruptas de cor em determinadas regiões da imagem. Quando deseja-se segmentar uma região da imagem com base em suas variações de intensidade ou de cor, são comumente utilizados métodos denominados detectores de borda (FILHO; NETO, 1999, p. 37).

Detectores de borda são operadores que transformam uma dada imagem binária em uma imagem de borda. Cada *pixel* de uma imagem de borda tipicamente representa a força da borda que passa através de uma determinada região da imagem original. Em algumas aplicações isto pode representar a direção da borda, com valores cujo o ângulo se encontra entre 0 à 2π (TANIMOTO, 2012, p.368).

Entre os métodos de detecção de borda, está o método de Sobel, do qual utiliza as regiões vizinhas de cada *pixel* situadas dentro de uma máscara dimensionada em 3x3 *pixels* para obter os valores correspondentes à borda. Este método computa estimativas dos gradientes na direção vertical e horizontal, utilizando cálculos individuais para cada direção. Na prática, são realizadas convoluções nas direções verticais e horizontais com os filtros abaixo, respectivamente (HORNBERG, 2007, p. 598).

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.8)$$

A grande vantagem da utilização do método de detecção de borda de Sobel é a sua baixa sensibilidade aos ruídos na imagem, ou em outras palavras, possui uma maior capacidade de suavização. Entretanto, este método não é apropriado para detectar bordas na diagonal, orientadas em aproximadamente $\pm 45^\circ$ (GONZALEZ; WOODS, 2010, p.468.). A Figura 2.8 apresenta o método de Sobel aplicado sobre uma imagem cinza.



Figura 2.8: Método de Sobel em uma imagem em escala de cinza (a) Imagem original; (b) Imagem de bordas.

Fonte: Elaborado pelo Autor

Outro método muito utilizado para detecção de bordas é o método de Canny. O método de Canny é considerado como um aperfeiçoamento de métodos como o de Sobel citado anteriormente, uma vez que sua principal operação é remover qualquer tipo de ruído contido na imagem antes de realçar suas bordas (TANIMOTO, 2012, p. 371). Para

isso, este método realiza a convolução da imagem com filtros Gaussianos apresentados pela Figura 2.9.

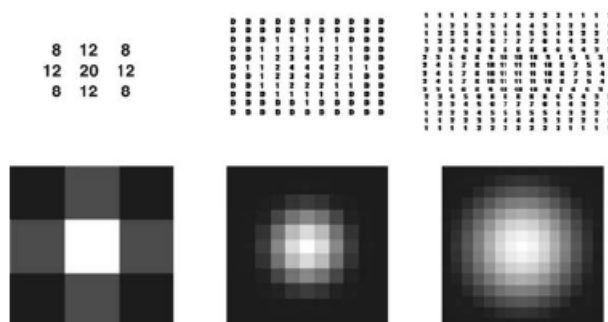


Figura 2.9: Filtros Gaussianos utilizados pelo método de Canny.

Fonte: (SOLOMON; BRECKON, 2011, p.96)

Quanto maior for o tamanho dos filtros Gaussianos, menor será a sensibilidade do detector e conseqüentemente maior será a taxa de erro na localização das bordas. Por fim, o método de Canny é concretizado utilizando outros métodos de detecção para encontrar a magnitude e a direção dos gradientes em cada *pixel*, conforme mencionado anteriormente (KRIG, 2014, p. 66). A Figura 2.10 apresenta a aplicação do método de Canny também em uma imagem cinza.

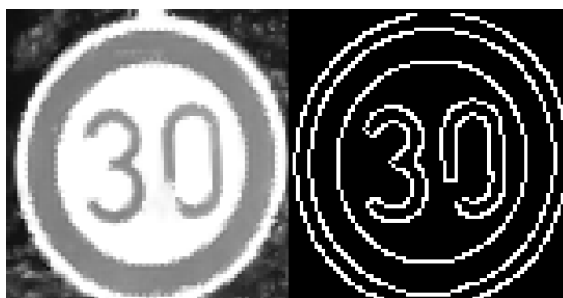


Figura 2.10: Método de Canny aplicado a uma imagem em escala de cinza (a) Imagem original; (b) Imagem de bordas.

Fonte: Elaborado pelo Autor.

2.4 Segmentação

Para realizar a segmentação da região de interesse, é necessário definir sua localização exata em todo o espaço da imagem, uma vez que não se sabe nada sobre onde possam estar os pixels pertencentes à esta. Nessas condições, todos os *pixels* são candidatos a pertencerem a região de interesse e por isso devem ser aceitos ou eliminados com base em propriedades globais predefinidas (GONZALEZ; WOODS, 2010, p.455). Uma das principais abordagens é a varredura global percorrendo toda a imagem em busca de

possíveis *pixels* pertencentes às regiões de interesse, onde se destacam técnicas como a transformada Hough.

2.4.1 Transformada Hough

Entre as técnicas de detecção global está a transformada Hough, muito utilizada para a localização de linhas, formas circulares e elípticas contidas em uma imagem digital. Proposta por Hough (1962), esta transformada possui a capacidade de detectar com alto nível de precisão objetos que possuem falhas em seu contorno, oclusões ou que estão em imagens ruidosas (TANIMOTO, 2012, p.375).

Para isso a transformada Hough utiliza como principal ferramenta equações que definem o formato do qual deseja-se localizar sobre uma determinada imagem. Considere o conjunto de pontos (x_m, y_n) de uma imagem digital. É possível obter todos os *pixels* (x_i, y_i) , onde $x_i \in x_n$ e $y_i \in y_n$, que pertencem a uma determinada reta através da equação geral de uma reta $y_i = ax_i + b$ (GONZALEZ; WOODS, 2010, p.483).

É possível ainda representar as características da reta através dos parâmetros a e b , isolando-os da equação anterior. Com isso, obtemos um novo espaço de representação denominado de espaço de parâmetros, do qual define a reta anterior através de um único ponto. Entretanto, um determinado ponto no espaço de parâmetros representa toda e qualquer reta que intercepta este através dos valores de x e y , conforme a equação $b_i = xa_i + y$ (JAHNE, 2005, p.459).

Por fim, é possível determinar os pontos pertencentes a uma mesma reta ao convertê-los para o espaço de parâmetros e observando que estes indicam a mesma posição no espaço. Através da Figura 2.12 é possível observar que uma reta que passa por dois pontos é caracterizada através de um ponto (a', b') , onde a' é a inclinação e b' é a interseção da reta contendo os dois pontos (ILLINGWORTH; KITTLER, 1988, p. 88).

A transformada Hough percorre todos os *pixels* da imagem e identifica uma reta através de um sistema de votação, observando quais *pixels* realizam mais votações no espaço de parâmetros. Através de um limiar, é possível encontrar pontos no espaço de parâmetros que possuam a maior probabilidade de indicar a existência e a localização das retas no plano cartesiano (GONZALEZ; WOODS, 2010, p.483).

Do mesmo modo, a transformada Hough pode ser utilizada para obter a localização de outras formas como círculos e elipses, utilizando a equação correspondente a cada formato. A Equação 2.12 define a detecção de um círculo de raio r através das coordenadas tridimensionais (a, b, r) . Na prática, muitos sistemas fixam ou delimitam o valor de r em um faixa predefinida para diminuir o número de execução necessárias para completar o

algoritmo (VERNON, 1991, p.131).

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.9)$$

2.4.2 Transformada Hough Generalizada

A transformada Hough generalizada é uma extensão da transformada Hough básica que permite a detecção de formatos arbitrários que não podem ser representados por equações analíticas. Para isso, o algoritmo necessita de templates de entrada que direcionarão a detecção conforme a correspondência dos *pixels* pertencentes à imagem e os *pixels* pertencentes à cada template (BALLARD, 1981, p.111).

Um template é um conjunto de *pixels* previamente definido e que deve representar de forma exata o padrão do objeto a ser detectado (VERNON, 1991, p.70). Ao transladar o centro do template por toda a imagem e obter, da mesma forma que a Transformada Hough, as votações de cada *pixel* é possível localizar os objetos semelhantes ao template na imagem (BALLARD, 1981, p.111). A Figura 2.11 demonstra o resultado apresentado pela Transformada Hough Generalizada para detecção de círculos e triângulos.

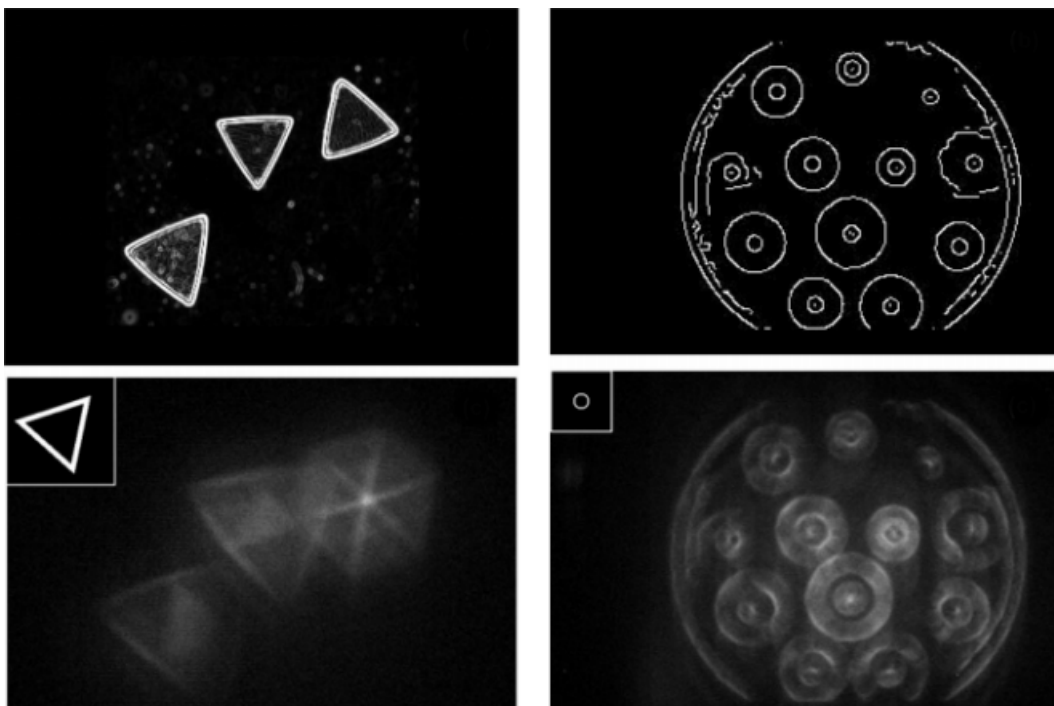


Figura 2.11: Detecção de círculos e triângulos com a transformada Hough generalizada.

Fonte: (FERNÁNDEZ et al., 2015, p.10589)

Apesar das vantagens obtidas através da Transformada Hough Generalizada, sua execução pode se tornar complexa a medida que o número de templates necessários aumenta significativamente em decorrência ao nível de detecção requerido. Se o objetivo for

detectar objetos que podem estar rotacionados ou escalonados em tamanho, é necessário um número relativamente grande de templates para alcançar um resultado satisfatório para o processo de segmentação. Outro ponto negativo é a grande quantidade de memória para acumular o sistema de votação desta técnica (HORNBERG, 2007, p.645).

2.5 Descrição

Uma das principais formas de aumentar a performance da etapa de classificação é realizar, após o processo de segmentação, a representação da região de interesse conforme suas características. Este processo envolve definir de forma única uma mesma característica em diversos cenários diferentes (GONZALEZ; WOODS, 2010, p.523)

Uma das principais abordagens é a representação através de descritores de imagem, onde a proposta é descrever as características globais e locais de maneira distinta dos demais componentes. Características globais indicam propriedades de imagem como um todo, envolvendo todos os *pixels*, enquanto características locais apontam para propriedades de pontos chaves ou regiões de interesse da imagem (CORSO; HAGER, 2009, p.448).

Em geral, descritores são vetores numéricos finitos que são extraídos da imagem através de operações baseadas em modelos estatísticos, modelos paramétricos, coeficientes obtidos através de transformações, ou até mesmo uma combinação dessas medidas (HAS-SABALLAH; ABDELMGEID; ALSHAZLY, 2016, p.1). O objetivo é obter vetores que representam adequadamente as características e que sejam invariantes à transformações na imagem como rotações ou alterações de iluminação, por exemplo.

Uma importante vantagem do uso de descritores, principalmente ao descrever características locais, é a redução da quantidade de dados definida pela diferença entre o tamanho da imagem ou região a ser descrita e o tamanho do vetor obtido. Isso faz com que descritores de imagem sejam utilizados em diversas aplicações que possuem pouco poder de processamento, como por exemplo aplicações mobile e sensores (CHANDRASEKHAR et al., 2009, p.1).

Descritores que não são afetados significativamente por transformações na imagem, como rotações, achatamentos, iluminação, etc; são denominados de descritores topológicos. Um exemplo de um descritor topológico simples seria o número de obstruções em uma imagem, do qual não deve ser afetado por transformações de alongamento ou de rotação (GONZALEZ; WOODS, 2010, p.542).

Nesse contexto, foram desenvolvidos diversos algoritmos para a concepção de descritores de imagens, para diversas condições e aplicações. Entre os algoritmos com uso consolidado estão o *Scale Invariant Features Transform* (SIFT) (LOWE, 1999), *Speeded-*

up *Robust Features* (SURF) (BAY et al., 2008) e *Histogram of Oriented Gradients* (HOG) (DALAL; DALAL, 2005). Em cada um destes algoritmos os descritores são obtidos através da utilização de filtros de imagem e operações não lineares.

O Histograma de Gradientes Orientados (*Histogram of Oriented Gradients* - HOG) é um método de descrição topológica para características locais com crescente uso em sistemas de visão artificial. Seu funcionamento é baseado no comportamento do gradiente de cada *pixel* da imagem e na formulação de histogramas a partir destes gradientes (DALAL; DALAL, 2005, p.1). Focaremos na utilização deste método para a etapa de descrição, apresentando as definições e os principais conceitos relacionados à este método.

2.5.1 Gradiente

O gradiente é uma ferramenta matemática que permite representar os níveis de intensidade ou a taxa de variação de intensidade em uma dada região da imagem. Como mencionado na seção 2.3.4, métodos de detecção de bordas utilizam os gradientes para detectar as bordas de uma imagem. Entretanto, os gradientes também podem ser utilizados para concepção de escritores internos (GONZALEZ; WOODS, 2010, p.543). Para cada *pixel* (x_m, y_n) de uma imagem f , o vetor gradiente pode ser obtido através da Equação 2.13.

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.10)$$

O vetor obtido possui a importante propriedade geométrica de apontar no sentido de maior taxa de variação intensidade de f no ponto (x,y) e sua magnitude (tamanho) quantifica em valores numéricos esta taxa de variação. A magnitude e a direção do vetor gradiente são obtidos através das equações 2.15 e 2.14, respectivamente (HORNBERG, 2007, p.589). Onde g_x e g_y são os valores verticais e horizontais dos gradientes de cada *pixel*. O ângulo de direção do vetor gradiente é sempre medido em relação ao eixo x e perpendicular a taxa de variação de intensidade na região do *pixel* (x,y) (JAHNE, 2005, p.333).

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \quad (2.11)$$

$$\alpha(x, y) = \tan^{-1} \begin{bmatrix} g_y \\ g_x \end{bmatrix} \quad (2.12)$$

2.5.2 Histograma de Gradientes Orientados (HOG)

Um descritor de um objeto deve ser topológico e não permitir variações com transformações afins. Entre os descritores que possuem avançado nível de topologia se encontra o HOG (SANTOS, 2012, p.2). Este método relaciona-se ao fato de que características como aparência e forma de objetos em uma imagem podem ser descritos através da distribuição de gradientes de intensidade dos *pixels* da imagem.

Como é possível observar através da Figura 2.12, para descrever uma imagem através do método HOG é necessário a execução de 5 passos. Antes de iniciar o processo é necessário que a imagem de entrada seja redimensionada conforme os valores escolhidos para os parâmetros de configuração (tamanho de célula, tamanho de blocos, número de classes no histograma, etc) (SULEIMAN; SZE, 2014, p.1).

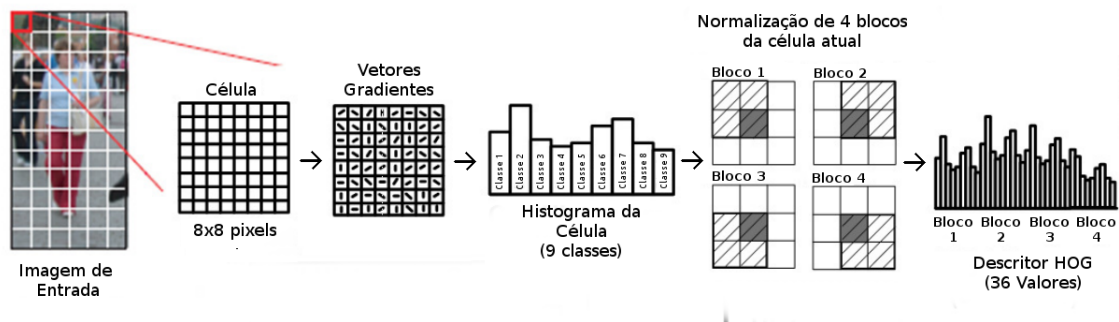


Figura 2.12: Etapas do método HOG.

Fonte: (SULEIMAN; SZE, 2014)

Primeiramente, a imagem de entrada é dividida em células de tamanho $n \times n$ *pixel*. O valor de n deve ser divisor da quantidade total de *pixels* do eixo vertical e horizontal da imagem redimensionada para que esta tenha uma quantidade exata de células. Para cada célula é computado os vetores gradientes de cada *pixel* através da convolução da célula com filtros Gaussianos, conforme mencionado na seção 2.3.4 (DALAL; DALAL, 2005, p.2).

Cada vetor gradiente é definido por sua magnitude e por sua direção. Para cada célula é obtido um histograma unidimensional, do qual cada uma de suas classes representam um valor de direção. Com isso, os valores de magnitude de cada gradiente são distribuídos entre as classes conforme a correspondência ou aproximação entre o valor da classe e o valor de direção do gradiente (SANTOS, 2012, p.2).

Em seguida, as células são ordenadas em grupos denominados de blocos que podem se sobrepor conforme configuração do HOG. Os histogramas obtidos das células pertencentes a cada bloco são normalizados através de uma modelo Euclidiano, diminuindo principalmente variações do descritor decorrente à mudanças de iluminação de uma mesma imagem (TOMASI, 2015, p.2).

Os histogramas normalizados são transpostos em um vetor linha e concatenados aos resultados das normalizações dos próximos blocos. Ao final deste processo, é formado um vetor numérico denominado de descritor HOG e o resultado gráfico sobre uma imagem em escala de cinza é apresentado pela [Figura 2.13](#) (DALAL; DALAL, 2005, p.2).



Figura 2.13: Descritor HOG de uma imagem em escala de cinza.

Fonte: (BALALI; GOLPARVAR-FARD, 2015, p.11)

Segundo Dalal e Dalal 2005 os parâmetros recomendados para uma melhor descrição topológica da imagem com o método HOG são células de tamanho 8×8 *pixels*, blocos de tamanho 16×16 , histogramas de 9 classes e uma imagem de entrada redimensionada em 64×128 *pixels*, do quais serão utilizados para a concepção dos descritores HOG da etapa de descrição..

2.6 Classificação

A etapa de classificação geralmente é o último passo para o reconhecimento de um objeto contido em uma imagem digital. Este processo envolve separar os objetos segmentados e descritos em diferentes classes, onde cada classe representa uma informação de interesse (DOUGHERTY, 2009, p.339). Como mencionando anteriormente na seção 2.2, técnicas de classificação podem ser divididas em três principais modelos: estatístico, estrutural e híbrido. Entre os mecanismos de classificação estatística está a técnica denominada de Máquina de Vetores de Suporte (SVM). Nas próximas subseções, serão apresentados os conceitos e as ferramentas matemáticas essenciais para o completo entendimento desta técnica.

2.6.1 Aprendizado de Máquina

Basicamente, o aprendizado de máquina busca extrair conhecimento através de dados. Também denominada como análise preditiva ou aprendizado estatístico, este campo de conhecimento converge de uma intersecção entre estatística, inteligência artificial e ciência da computação (MÜLLER; GUIDO, 2016, p.1). Por meio da concepção de técnicas

que desenvolvem o aprendizado a partir de sistemas computacionais capazes de adquirir conhecimento de forma automática, o aprendizado de máquina proporciona o nascimento de diversas aplicações em diversas áreas tecnológicas e científicas (ALPAYDIN, 2014, p.3).

Na sua essência, o aprendizado de máquina se baseia principalmente no conceito de indução, ou em outras palavras, na inferência lógica que permite obter conclusões genéricas sobre um conjunto particular de exemplos. É através da indução que o cérebro humano pode derivar conhecimento novo sobre situações anteriormente apresentadas e para isso é necessário que se tenha um conjunto de exemplos de treinamento que deverá ser a fonte do aprendizado (MONARD; BARANAUSKAS, 2003, p.39).

Apesar disso, muitas vezes as hipóteses levantadas através da indução não necessariamente preservam a verdade. Um conjunto de exemplos insuficientes ou exemplos que não foram bem escolhidos podem formar hipóteses inválidas, influenciando na eficácia do sistema. Neste contexto, compreender o poder e a limitação de cada técnica utilizando uma metodologia que permite avaliar os conceitos induzidos por essas técnicas em determinadas situações, é a principal atividade a ser realizada antes do desenvolvimento do sistema em si (MONARD; BARANAUSKAS, 2003, p.39). O aprendizado indutivo pode ser dividido em supervisionado e não supervisionado dependendo do modo em que é obtido o conhecimento.

No aprendizado supervisionado é fornecido ao sistema um conjunto de exemplos de treinamento que já possuem a informação dos atributos que devem ser posteriormente reconhecidos pelo sistema. Um exemplo nada mais é do que um tupla de atributos passado ao algoritmo com o objetivo de construir um bom classificador que possa determinar corretamente a classe dos novos exemplos que ainda não foram rotulados, ou seja, exemplos dos quais os atributos são desconhecidos (CAMASTRA; VINCIARELLI, 2008, p.86).

Após a concepção do classificador, é possível avaliá-lo em relação ao seu grau de precisão, velocidade de aprendizado, compreensibilidade ou qualquer outro parâmetro que defina a qualidade deste classificador mediante a tarefa em questão (MONARD; BARANAUSKAS, 2003, p.43). Um dos principais fatores limitantes da qualidade de um classificador é a sua capacidade de trabalhar com dados imperfeitos que podem ser derivados do próprio processo que gerou os dados, do processo de aquisição, das classes rotuladas incorretamente, etc (ALPAYDIN, 2014, p.25).

No aprendizado supervisionado, podemos ainda categorizá-lo em classificação e regressão. A classificação tem como objetivo determinar a classe em que um exemplo de entrada pertence. Para isso, o modelo analisa os outros exemplos fornecidos anteriormente e que já possuem indicação de classe, a fim de aprender como classificar o novo registro. Já a regressão é utilizada quando um exemplo é identificado por um valor numérico e não um valor categórico. Assim, pode-se determinar o valor de uma determinada variável analisando-se os valores das demais (CAMILO; SILVA, 2009, p.9).

No aprendizado não supervisionado, o indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou *clusters*. Após a determinação dos agrupamentos, normalmente, é necessária uma análise para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado (ALPAYDIN, 2014, p.11). A Figura 2.14 representa a hierarquia completa do aprendizado de máquina.

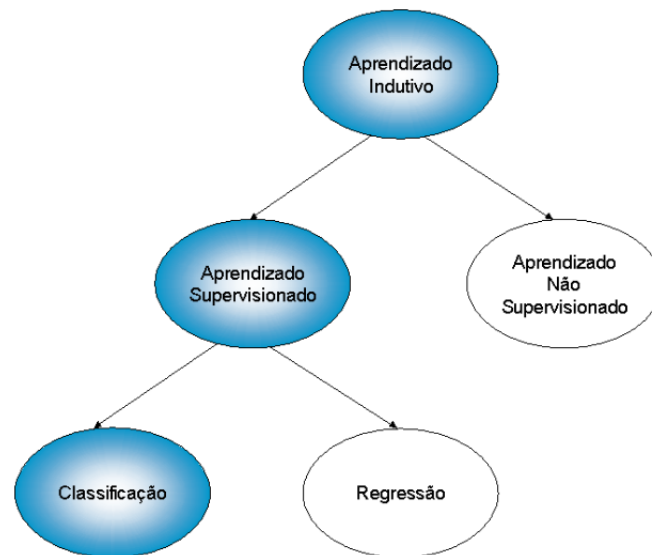


Figura 2.14: Hierarquia do aprendizado de máquina.

Fonte: (MONARD; BARANAUSKAS, 2003)

Existem diversos paradigmas de aprendizado que estabelecem a forma em que o conhecimento é adquirido por um sistema baseado em aprendizado de máquina. Entre os paradigmas estão o simbólico, o baseado em eventos, o conexionista, o genético e o estatístico (MONARD; BARANAUSKAS, 2003, p.41). No nosso contexto, vamos focar apenas no paradigma de aprendizado estatístico que foi a base para o desenvolvimento das SVM's.

2.6.2 Teoria do Aprendizado Estatístico

Um mesmo algoritmo de aprendizado, pode ser treinado utilizando diferentes parâmetros e diferentes conjuntos de treinamento, conseqüentemente, este mesmo algoritmo pode conceber diferentes classificadores para um mesmo cenário. O problema de selecionar o classificador dentro do conjunto de classificadores gerados pelo algoritmo é denominado seleção de modelo e a solução para este problema é a determinação de performance destes classificadores através da Teoria do Aprendizado Estatístico (CAMASTRA; VINCIA-RELLI, 2008, p.149).

Considere um conjunto treinamento T composto por duas classes representadas por círculos e triângulos, onde os elementos de cada classe são definidos através de tu-

plas de exemplos (x_i, y_i) . Através do processo de aprendizagem um algoritmo fornece 3 classificadores f_a , f_b e f_c pertencentes ao conjunto F de todos os possíveis classificadores gerados por este mesmo algoritmo (ALPAYDIN, 2014, p.17). Este cenário é apresentado pela Figura 2.15.

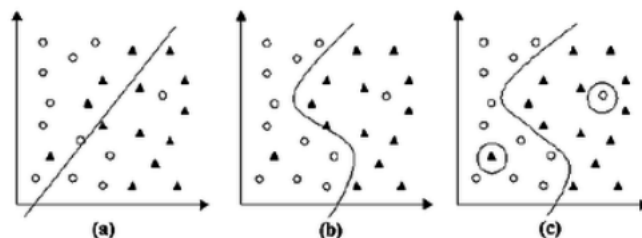


Figura 2.15: Classificadores segundo três hipóteses.

Fonte: (LORENA; CARVALHO, 2007, P.46)

O classificador f_a utiliza uma fronteira de decisão linear sobre o conjunto onde os pontos extremos das duas classes são muito próximos entre si, por isso são considerados não confiáveis. Neste caso, o classificador comete muitos erros, mesmo para casos muito simples. Pode-se considerar que houve um subajuste do classificador, uma vez que este não é capaz de se ajustar às instâncias de treinamento (SMOLA; SCHÖLKOPF, 1998, p.7).

Em contrapartida, o classificador f_c utiliza fronteiras de decisão complexas e difíceis de serem definidas matematicamente, possuindo a capacidade alta de determinar exemplos do conjunto de treinamento, mas com grandes possibilidades de cometer erros quando confrontado com novos exemplos. Pode-se considerar que houve um sobreajuste, onde considera-se que o classificador memorizou os dados do conjunto de treinamento (SMOLA; SCHÖLKOPF, 1998, p.7).

O classificador f_b seria considerado um intermediário entre as duas propostas anteriormente mencionadas. Nesta hipótese, o classificador utiliza uma fronteira não linear com baixa complexidade de definição e por isso consegue determinar com sucesso quase todas as classes do conjunto de treinamento. Sendo assim, o problema da seleção de modelo é resolvido quando o melhor desempenho de generalização é atingido através do equilíbrio entre a precisão alcançada em um conjunto de treinamento particular e capacidade do sistema de aprender qualquer conjunto de treinamento sem erro (SMOLA; SCHÖLKOPF, 1998, p.20).

Para isso, busca-se a aproximação entre uma determinada função alvo $f(x)$ e a saída real do sistema, onde esta aproximação pode ser mensurada através da estimação do erro mínimo (também denominado risco mínimo), do qual mede a capacidade de generalização de f (HAYKIN, 2007, p.115).

Assumindo-se que os exemplos apresentados pelo conjunto de treinamento são

independentes e identicamente distribuídos de acordo com uma distribuição de probabilidade $P(x,y)$, o risco esperado é definido pela [Equação 2.13](#). Onde, $c(f(x), y)$ é uma função de custo relacionando a previsão $f(x)$ quando a saída desejada é y ([VAPNIK, 2013](#), p.18). No caso de uma classificação binária, uma função comumente usada é a $c(f(x), y) = \frac{1}{2} |yf(x)|$, que retorna o valor 0 se x é classificado corretamente e 1 caso contrário.

$$R(f) = \int c(f(x), y)dP(x, y) \quad (2.13)$$

Entretanto, a minimização do risco esperado não pode ser realizada diretamente com esta equação, uma vez que em geral a distribuição de probabilidade $P(x, y)$ é desconhecida. No processo de aprendizagem, a única informação disponível está contida no conjunto de treinamento que também é amostrado de $P(x, y)$ ([HAYKIN, 2007](#), p.116).

Para contornar esta dificuldade matemática, é utilizado o princípio indutivo da minimização do risco empírico. O risco empírico, apresentado pela [Equação 2.14](#), mede o desempenho do classificador f nos dados de treinamento, por meio da taxa de classificações incorretas obtidas no conjunto de treinamento y ([ALPAYDIN, 2014](#), p.24).

$$R_{emp}(f) = 1/n \sum_{i=1}^n L(y, f(x, \alpha)) \quad (2.14)$$

Nesta equação, Com $n \rightarrow \infty$, é possível estabelecer um critério de aceitação para qual a performance do classificador em questão é apropriada para o problema de classificação específico, ou seja, é possível obter classificadores cujos valores de risco empírico convergem para o risco esperado ([VAPNIK, 2013](#), p.20). Entretanto, para conjuntos de treinamento menores, geralmente não é possível determinar esse tipo de garantia.

Neste contexto, dentro de um conjunto amplo de classificadores F é sempre possível encontrar um classificador com risco empírico pequeno. Deve-se então restringir os classificadores através de limites do risco esperado e com isso a Teoria do Aprendizado Estatístico fornece ferramentas para mensurar a capacidade de cada classificador. A ferramenta mais utilizada é a Dimensão VC, que é componente importante para o limite de generalização da SVM ([ALPAYDIN, 2014](#), p.28).

A dimensão VC (Vapnik-Chervonenkis) é uma medida de capacidade de classificação sobre a família de funções geradas por um algoritmo de aprendizado, ou em outras palavras, é o valor que representa o número de exemplos necessários para se aprender uma classe de interesse de maneira confiável ([HAYKIN, 2007](#), p.119). Quando maior for a dimensão VC, maior será o número de exemplos necessários para aprender a classe.

O erro de generalização, diferentemente do erro de treinamento (risco empírico), é a taxa de erros de um determinado classificador aplicado a um conjunto de exemplos que

ainda não foram vistos. Para calcular a capacidade de generalização, é necessário mensurar o limitante da respectiva taxa de erro através da soma dos dois elementos anteriormente mencionados: o risco empírico e a dimensão VC (HAYKIN, 2007, p.119).

As máquinas de aprendizado que buscam minimizar o limitante superior são ditas seguir o princípio estatístico da minimização de risco estrutural. Entre estas máquinas está a Máquina de Vetores de Suporte (do inglês, Support Vector Machine - SVM) que objetivam aumentar a capacidade de generalização através da minimização do risco empírico e da dimensão VC (VAPNIK, 2013, p.31).

2.6.3 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (do inglês *Support Vector Machine* - SVM) é um método de aprendizado de máquina supervisionado para classificação de padrões baseado na teoria de aprendizado estatístico criada por Vapnik (2013) e colaboradores. As SVM's se tornaram famosas nas últimas décadas por possuírem, em determinadas situações, performance comparável e muitas vezes superior às obtidas por outros algoritmos de aprendizado (CAMPBELL; YING, 2011, p.1).

Alguns dos motivos para isso estão fundamentados no bom desempenho desta técnica na generalização de bases de reais, nos recursos teóricos bastante consolidados, no processo de treinamento que elimina a possibilidade de mínimos locais e na existência de poucos parâmetros livres para ajustá-lo. Por isso, esta técnica vem sendo empregada em diversas áreas como categorização de texto, reconhecimento de escrita e voz, classificação de imagens, análise de bio-sequências, etc (BEGG; PALANISWAMI, 2006, p.244).

Na sua forma mais simples, o objetivo de uma SVM é a obtenção de fronteiras lineares para separação dos dados em duas classes (classificação binária). A classificação linear consegue apenas delimitar classes que pertencem a um conjunto linearmente separável, enquanto a classificação não linear consegue delimitar classes que pertencem a um conjunto não linearmente separável (CAMPBELL; YING, 2011, p.1).

Sendo assim, são poucas as aplicações que podem utilizar a classificação linear, uma vez que aplicações reais necessitam de fronteiras de complexa definição aumentando consequentemente a complexidade do sistema classificador (CAMPBELL; YING, 2011, p.6). Apesar de sua limitação, a classificação linear apresenta propriedades importantes para o embasamento inicial da SVM's.

No caso da minimização do risco estrutural, uma SVM Linear produz um valor zero para o primeiro termo e minimiza o segundo termo para definir os limites (HAYKIN, 2007, p.349). A Figura 2.16 apresenta em (a) um conjunto de treinamento linearmente separável em um espaço bidimensional, enquanto (b) apresenta um conjunto não linearmente separável. As linhas apresentadas em ambas as situações são denominadas de

superfície de decisão. No caso da (a), devido a linearidade da superfície de decisão, esta também é denominada de hiperplano (LORENA; CARVALHO, 2007, p.52).

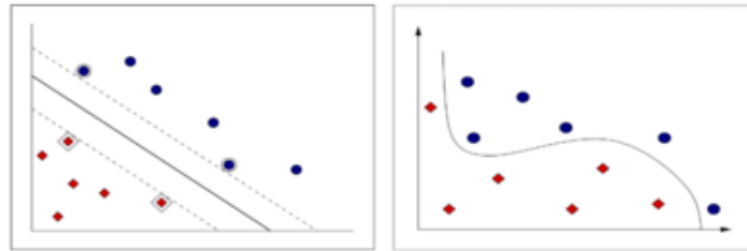


Figura 2.16: Classificador linear à esquerda e não linear à direita.

Fonte: (SANTOS, 2002, p.36)

No contexto de aprendizado de máquina, um hiperplano é um espaço de separação n dimensional. Em um espaço bidimensional, um hiperplano é definido através de uma reta, enquanto em um espaço tridimensional, um hiperplano é definido através de um plano (SLOUGHTER, 2001).

O algoritmo de um SVM tem como principal objetivo encontrar um hiperplano de margem máxima, do qual separa corretamente todas as classes de um determinado conjunto e mantém a maior distância entre os vetores mais próximos a este hiperplano (BEGG; PALANISWAMI, 2006, p.249). O hiperplano que detém destas características é denominado hiperplano ótimo e os vetores que mais se aproximam deste hiperplano são denominados de vetores de suporte (HAYKIN, 2007, p.351). Para a definição de um hiperplano aplicado à classificação linear, utiliza-se a equação da reta no formato como é apresentado pela Equação 2.15:

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b = 0 \quad (2.15)$$

Essa equação, também denominada de *kernel*, segmenta o conjunto de dados em duas regiões: $\mathbf{w} \cdot \mathbf{x} + b > 0$ e $\mathbf{w} \cdot \mathbf{x} + b < 0$. Seja x_1 um ponto no hiperplano de forma que $\mathbf{w} \cdot \mathbf{x} + b \geq 1$ e x_2 um ponto no hiperplano de forma que $\mathbf{w} \cdot \mathbf{x} + b \leq -1$, é possível estabelecer a distância d através dos vetores normais perpendiculares ao hiperplano direcionados aos respectivos pontos, conforme Figura 2.17 (CAMPBELL; YING, 2011, p.3).

O comprimento do vetor projetado através das diferenças entre os vetores normais é dado por $\frac{2}{\|\mathbf{w}\|}$, onde é possível obter a maximização da margem entre o hiperplano e os vetores de suporte através da minimização de $\|\mathbf{w}\|$.

Em aplicações reais, raramente o conjunto de dados é linearmente separável. Isso se deve a diversos fatores como presença de ruídos, *outliers* (exemplos fora da curva) ou à própria natureza do problema. Para tentar contornar esta dificuldade, a SVM's lineares são estendidas para lidar com conjuntos de treinamento mais gerais. Para realizar essa

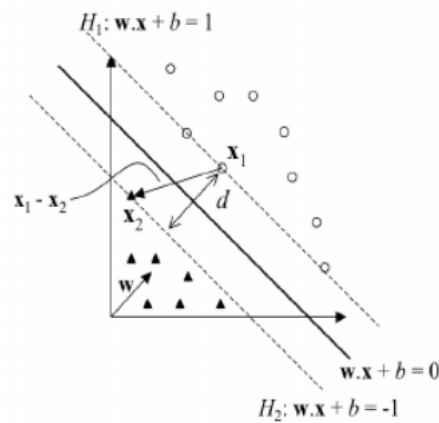


Figura 2.17: Cálculo da margem máxima.

Fonte: (LORENA; CARVALHO, 2007, p.54)

tarefa é necessário que alguns dados possam violar a restrição imposta anteriormente ($\mathbf{w} \cdot \mathbf{x} + b \geq 1$ e $\mathbf{w} \cdot \mathbf{x} + b \leq -1$) (CAMPBELL; YING, 2011, p.6).

Este procedimento suaviza as margens da fronteira de classificação, permitindo que alguns dados permaneçam dentro da margem máxima e também a ocorrência de alguns erros de classificação. O critério de quantas classificações podem ser realizadas de forma errônea é definido através do risco empírico. SVM's com estas características são denominadas SVM lineares com margens suaves (LORENA; CARVALHO, 2007, p.57).

Há casos de conjuntos de dados dos quais não são possíveis de serem separados através de um hiperplano. Exemplos disso são apresentados através da Figura 2.18. Para realizar a separação dos dados deste tipo de conjunto, a SVM mapeia este conjunto de treinamento de seu espaço original para um novo espaço de maior dimensão, denotado de espaço de características. Este mapeamento permite que o conjunto seja separado através de uma SVM Linear (SMOLA; SCHÖLKOPF, 1998, p.13).

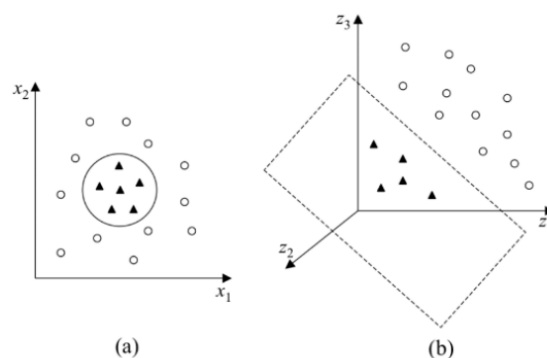


Figura 2.18: (a) Dados não separáveis linearmente em um plano bidimensional (b) Dados não separáveis linearmente em um plano tridimensional .

Fonte: (LORENA; CARVALHO, 2007, p.60)

3 Desenvolvimento

Nesta seção são apresentadas todas as etapas do algoritmo proposto, evidenciando as ferramentas utilizadas para a concepção do sistema e abordando todas as metodologias envolvidas em cada etapa. Com isso, na seção 3.1 é apresentada a arquitetura geral do algoritmo. Na seção 3.2 é apresentado o papel de um banco de imagens e qual foi escolhido para testes iniciais do algoritmo. A seção 3.3 descreve toda a implementação dos módulos do algoritmo proposto. A seção 3.4 retrata os experimentos realizados com placas de sinalização do sistema brasileiro de trânsito. Por fim, a seção 3.5 apresenta os resultados obtidos.

3.1 Arquitetura Geral

A [Figura 3.1](#) apresenta a arquitetura geral de um sistema de reconhecimento automático de placas de sinalização, onde é possível notar um fluxo bem definido através dos módulos do sistema. Para a parte superior do diagrama, os resultados são ainda imagens, enquanto que para a parte inferior os resultados são informações numéricas ou rótulos. O resultado final é a identificação da classe da qual a placa de sinalização pertence.

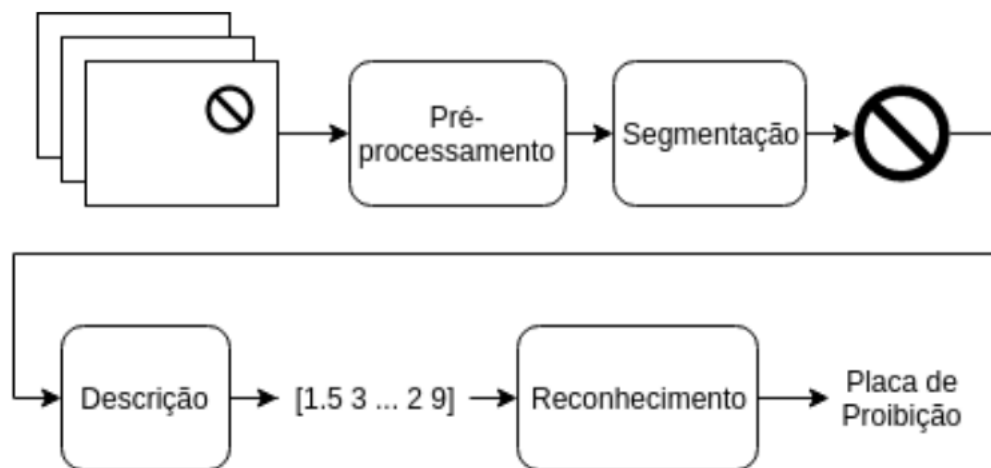


Figura 3.1: Diagrama da arquitetura geral

Fonte: Elaborado pelo Autor

Os dados de entrada são imagens completas contendo placas de sinalização em localizações arbitrárias na imagem e o ambiente do qual se encontra esta placa. Estas imagens são aplicadas ao módulo de pré-processamento e posteriormente ao módulo de segmentação. O resultado são regiões segmentadas, contendo apenas a estrutura básica da placa de sinalização em questão.

Na parte inferior, a região segmentada é então aplicada ao módulo de descrição, resultando em vetores numéricos que representam suas características. Por fim, esse vetor numérico serve como dado de entrada para o módulo de reconhecimento que apresenta em sua saída os rótulos correspondentes a cada placa de sinalização das imagens de entrada do sistema.

Nas próximas seções são apresentadas as técnicas utilizadas em cada módulo da arquitetura geral do sistema, bem como as imagens utilizadas como dados de entrada, extraídas de um banco de imagens específico para este cenário.

3.2 Banco de Imagens

Um banco de imagens fornece os dados necessários para os processos de treinamento e teste do sistema em geral, e por isso deve possuir determinadas características para englobar, se não todas, mas grande parte das imprevisíveis situações onde o algoritmo pode ser aplicado. Dessa forma, para poder idealizar os diversos algoritmos recentemente apresentados, diversas instituições em diversos países se propuseram a criar seus próprios bancos de imagens, uma vez que cada país possui um conjunto de placas de sinalização de trânsito diferente dos demais.

Neste contexto, foram criados bancos de imagens públicos como o LISA ([MOGELMOSE; TRIVEDI; MOESLUND, 2012](#)) constituído de placas de sinalização do sistema americano de trânsito; o BelgiumTS ([MATHIAS et al., 2013](#)) constituído de placas de sinalização do sistema belga de trânsito e o GTSDDB ([HOUBEN et al., 2013](#)) constituído de placas de sinalização do sistema alemão de trânsito, onde este último foi utilizado em competições na conferência internacional de redes neurais ([IJCNN](#)) de 2013.

As Figuras [3.2](#) e [3.3](#) apresentam, respectivamente as amostras de treinamento e teste extraídas do banco de imagens GTSDDB. Este banco possui imagens de 42 classes de placas de sinalizações de trânsito, totalizando em 600 imagens de treinamento e 300 imagens de teste. Como é possível observar através da [Figura 3.2](#), as imagens de teste já se apresentam segmentadas para o treinamento e foram obtidas em diversas situações de angulação e iluminação.

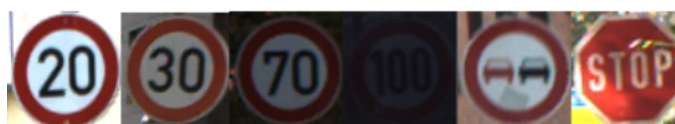


Figura 3.2: Amostras de treinamento GTSDDB

Adaptado de ([HOUBEN et al., 2013](#))

Como é possível observar nas próximas sessões, a imagem da [Figura 3.2](#) é utilizada para demonstrar as etapas de implementação, uma vez que esta representa situações crí-

ticas onde o algoritmo pode se aplicado, como ambientes complexos com diversos objetos semelhantes à placa de sinalização. Além disso, esta imagem também é utilizada para extrair os resultados finais deste trabalho.



Figura 3.3: Amostra de teste GTSDDB

Adaptado de (HOUBEN et al., 2013)

Em decorrência às características como a diversidade de imagens, a forte semelhança entre algumas classes de placas de sinalização com classes do sistema brasileiro de trânsito e o fato de as imagens serem coloridas, o banco de imagens GTSDDB foi utilizado para treinamentos e testes iniciais do algoritmo proposto. Entre as 42 classes, foram escolhidas 11 classes contendo apenas placas com borda vermelha e formato circular devido às propriedades de cor e formato utilizadas no algoritmo.

3.3 Implementação

O desenvolvimento do algoritmo foi realizado utilizando a plataforma MATLAB[®] na versão 2015a, conforme é indicado na configuração do sistema apresentada pelo [Apêndice A](#). Esta plataforma foi utilizada devido às ferramentas inclusas no *toolbox* de estatística de aprendizado de máquina, do qual possui módulos focados em classificação. Algumas funções, como o treinamento e teste das SVM, estão contidas nestes módulos e são utilizadas para implementar a etapa de classificação do trabalho proposto.

Com o uso das ferramentas deste *toolbox*, somente foram implementadas as etapas de pré-processamento, segmentação e descrição, conforme a literatura de cada técnica empregada. Para o treinamento e testes das etapas mencionadas, foram utilizadas em primeiro lugar, imagens do banco GTSDDB, que são coloridas e possuem resolução de 1360 x 800 *pixels*. Sendo assim, nas próximas subseções são apresentadas as implementações realizadas em cada módulo da arquitetura geral.

3.3.1 Pré-processamento

A etapa de pré-processamento pode ser dividida em submódulos conforme o diagrama apresentado pela [Figura 3.4](#). Ao final desta etapa são apresentadas imagens com alto nível de realce, evidenciando a região onde se encontra a placa de sinalização. O nível de realce alcançado permite a adequada execução das posteriores etapas de segmentação e descrição.

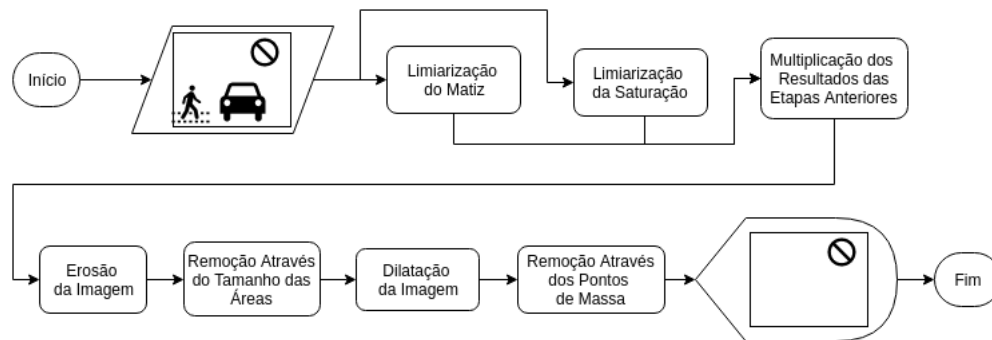


Figura 3.4: Diagrama do fluxo da etapa de pré-processamento.

Fonte: Elaborado pelo Autor.

Primeiramente, decidimos por investigar o modelo de cores HSV, pelo principal motivo de que este é capaz de diferenciar, entre os seus elementos a informação de intensidade da informação de cor. Um dos principais meios para realizar o realce de um objeto contido na imagem no modelo HSV é aplicar o processo individual de limiarização sobre os elementos deste modelo.

O elemento matiz define uma cor através de valores numéricos de angulação que variam entre 0° à 360° , como mencionado na seção 2.3.1. As placas analisadas possuem em suas extremidades a predominância de cores quentes, mais especificamente cores próximas ao vermelho e ao converter esta faixa de cores para os valores de matiz, obtemos um segmento definido aproximadamente pelos valores entre 316° à 46° . Contudo, através dos testes realizados e apresentados pela seção 3.5, os limiares que apresentam melhor performance para o cenário em questão são definidos pelos valores de 338° e 21° .

Em resumo, o processo de limiarização é aplicado sobre cada *pixel* da imagem, onde seu valor de matiz é comparado com os limiares anteriormente mencionados. Os *pixels* que condizem com a regra imposta pelos processos de limiarização possuem seus valores de matiz convertidos para 1, caso contrário são convertidos para 0, resultando na imagem binária apresentada pela [Figura 3.5b](#).

A saturação corresponde à quantidade de cor que um determinado *pixel* possui, como também comentado na seção 2.3.1. Semelhante ao processo de limiarização do elemento de matiz, a limiarização do elemento de saturação pode ser realizada aplicando-se limiares aos valores de saturação de cada *pixel* da imagem.

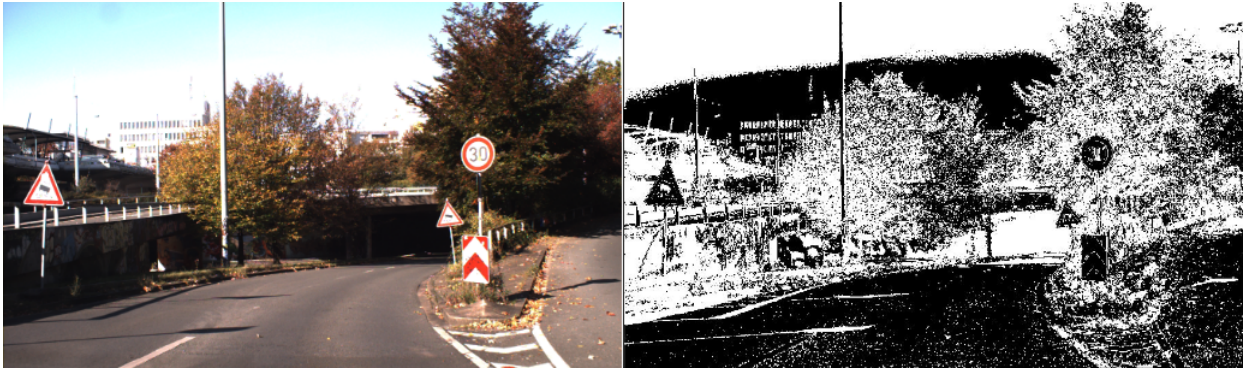


Figura 3.5: Processo de limiarização da matiz (a) Imagem original; (b) Imagem binária;

Fonte: Elaborado pelo Autor.

Entretanto, os valores de saturação correspondentes às cores mais intensas encontradas principalmente nas placas de sinalização analisadas aparecem com menor frequência na imagem e por isso, ao obter o histograma de saturação, os valores de saturação se concentram mais no início deste, como é apresentado pela [Figura 3.6](#). Para aumentar a precisão do processo de limiarização e conseqüentemente o realce da placa, foi realizada a equalização do histograma, onde o resultado é apresentado pela [Figura 3.7](#), dos quais os valores de saturação estão normalizados entre 0 e 1.

Após o processo de equalização do histograma de saturação, é efetuada a limiarização dos valores de saturação, onde é utilizado um limiar numérico de 255, do qual foi escolhido com base nos resultados apresentados pela seção 3.5. Com isso, são abstraídos 70% dos *pixels* com valores abaixo do deste limiar de toda a imagem original apresentada pela figura [Figura 3.8a](#), resultando na imagem binária apresentada pela [Figura 3.8b](#).

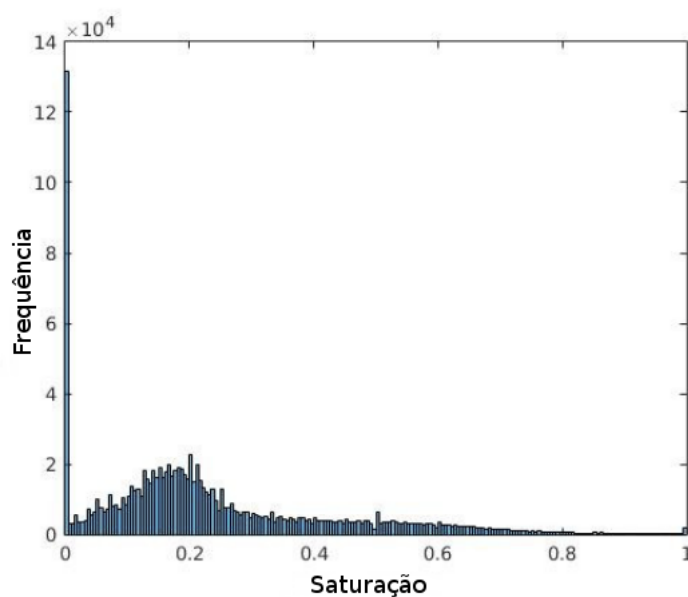


Figura 3.6: Histograma da saturação.

Fonte: Elaborado pelo Autor

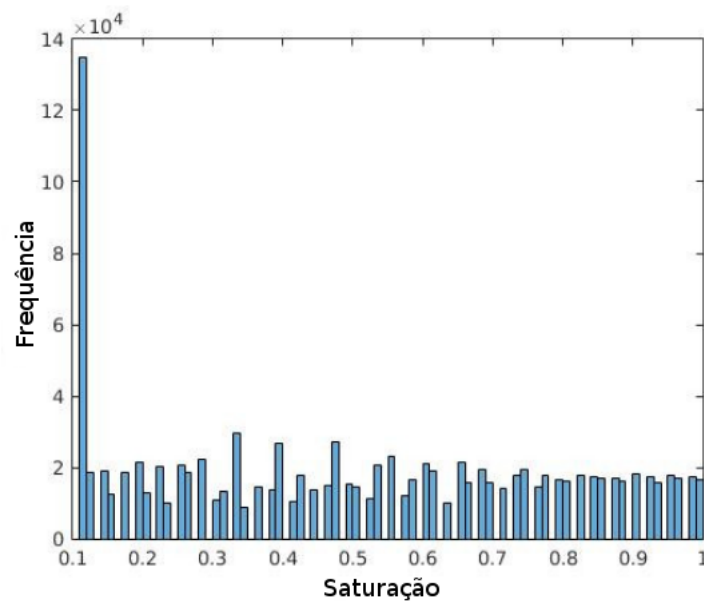


Figura 3.7: Histograma da saturação equalizado.

Fonte: Elaborado pelo Autor

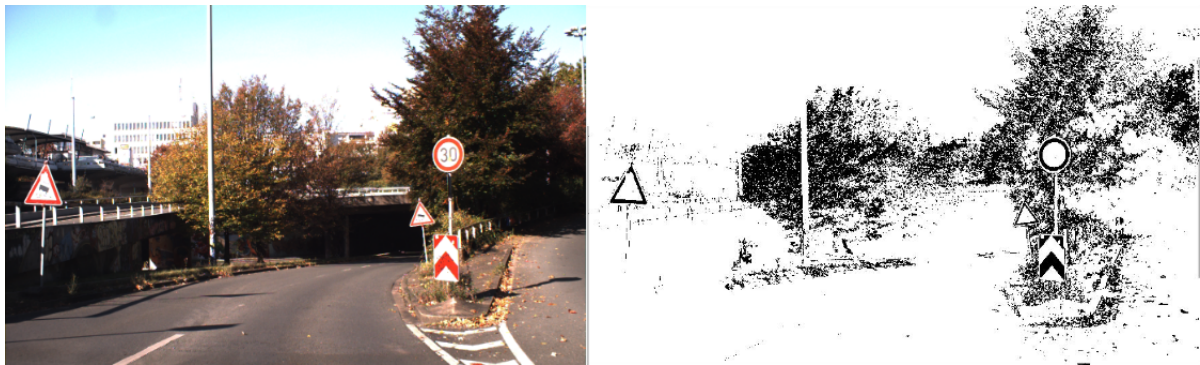


Figura 3.8: Processo de limiarização da saturação (a) Imagem original; (b) Imagem binária;

Fonte: Elaborado pelo Autor.

Cada *pixel* da imagem resultante da limiarização do matiz é multiplicado pelo *pixel* de mesma posição da imagem resultante da limiarização da saturação removendo totalmente os *pixels* que não sejam da cor vermelha e que não tenham intensidade de cor suficientes. O resultado deste processo é apresentado pela [Figura 3.9](#), onde é possível observar a borda das placas de sinalização já bem evidenciadas.

Além dos *pixels* pertencentes às placas de sinalização de trânsito, outros *pixels* não são removidos da imagem, pois também possuem matiz e saturação correspondentes aos limiares anteriormente aplicados, sendo denominados de falsos positivos da etapa de pré-processamento. A existência destes *pixels* faz com que os resultados da etapa de segmentação sejam afetados, e por isso devem ser removidos da imagem. Para reduzir os falsos positivos, investigamos a utilização de processos de morfologia e remoção de áreas.

A primeira operação morfológica aplicada é a operação de enchimento. Esta ope-



Figura 3.9: Multiplicação das limiarizações da matiz e saturação.

Fonte: Elaborado pelo Autor.

ração permite que *pixels* pertencentes ao fundo da imagem e que estejam envoltos por *pixels* de valor 1 tenha seus valores convertidos para este último, ou em outras palavras, este processo realiza o enchimento de buracos na imagem. O resultado obtido é o realce de toda a área interna definida pelas bordas vermelhas das placas de sinalização, conforme apresentado pela [Figura 3.10](#).



Figura 3.10: Processo de enchimento da imagem.

Fonte: Elaborado pelo Autor.

Em seguida, a operação de erosão da imagem, também morfológica, permite que ambas as informações, pertencentes ou não a placa de sinalização, sejam reduzidas em tamanho. Com isso, o efeito obtido é a remoção de pequenas áreas em proporção ao tamanho do elemento estruturante circular utilizado. A perda dos *pixels* pertencentes à placa não demonstra-se significativa, uma vez que em geral, a área das placas de sinalização é uma das maiores contidas na imagem da [Figura 3.10](#). O resultado deste processo é apresentado pela [Figura 3.11](#).

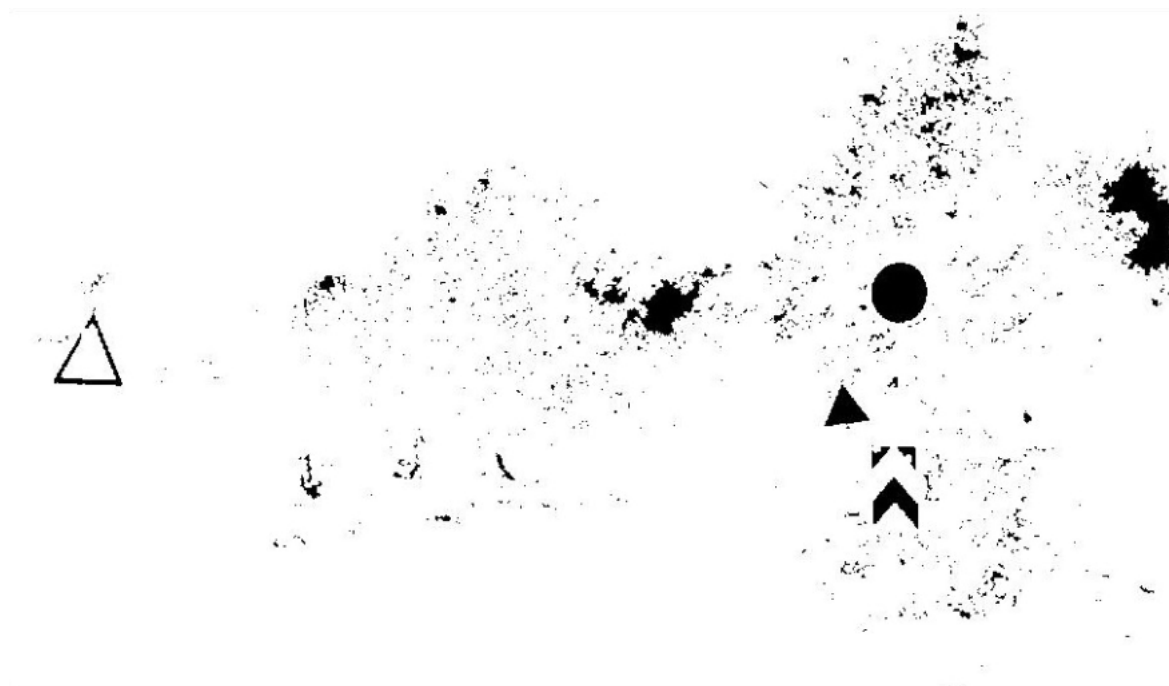


Figura 3.11: Processo de erosão da imagem.

Fonte: Elaborado pelo Autor.

É perceptível na [Figura 3.11](#) a diferença entre a área pertencente às placas de sinalização e a área dos falsos positivos. Para reduzir ainda mais a quantidade de falsos positivos é aplicado sobre a imagem a remoção de áreas conforme o tamanho mínimo e máximo delimitado pelas placas contidas no banco de imagens GTSDB.

As placas contidas nas imagens deste banco possuem valores de raio que variam de 10 *pixels* até 50 *pixels* e, conforme o cálculo de área de uma circunferência, são removidas áreas maiores e menores não correspondentes às áreas calculadas, como é apresentado pela [Figura 3.12](#).

Para realizar a segmentação da placa de sinalização, é necessário obter a borda circular definida pela sua respectiva área. Para obter o formato circular da placa de sinalização é realizada a detecção de bordas sobre a imagem da [Figura 3.12](#). Mas antes, para diminuir o serrilhamento e a descontinuidades dos outros formatos contidos na imagem, é aplicado a última operação morfológica: a dilatação da imagem.

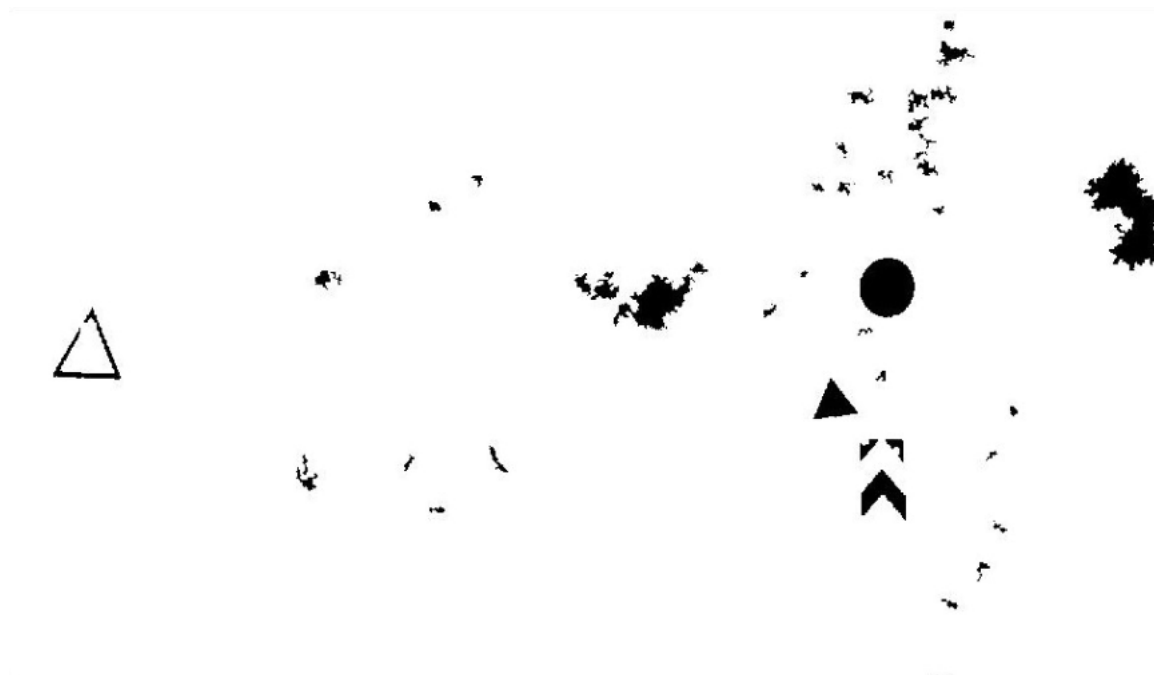


Figura 3.12: Remoção das áreas.

Fonte: Elaborado pelo Autor.

O serrilhamento e a descontinuidade características das demais bordas interferem no processo de segmentação, uma vez que estes *pixels* irão representar a soma de falsos positivos no sistema de votação da transformada Hough generalizada, detalhada nos próximos tópicos. A dilatação reduz as descontinuidades e serrilhamentos implodindo as áreas com *pixels* de valor 1 da imagem, conforme o tamanho do elemento estruturante. O resultado deste processo é apresentado pela [Figura 3.13](#).

A detecção de borda é aplicada sobre a imagem resultante com o objetivo de transparecer o formato circular ou elíptico característico da placa de sinalização em questão. Para a detecção de borda é utilizado o método de Canny que apresenta bons resultados com bordas que não sejam horizontais e verticais, como mencionado na seção 2.3.4. O resultado deste processo de detecção de borda é apresentado pela [Figura 3.14](#).

A última etapa do pré-processamento é a remoção dos *pixels* de borda através do cálculo do centro de massa de cada objeto contido na imagem. Para isso, são anulados todos os vetores internos destes objetos e o ponto de anulação obtido é utilizado como referência para se obter máscaras circulares de raio proporcional ao tamanho dos objetos analisados. Caso o raio destas máscaras for maior ou menor que os raios delimitados pelas placas do banco GTSDb, os *pixels* das bordas dos objetos em questão são abstraídos para o fundo da imagem. Com isso, o resultados deste processo é apresentado pela [Figura 3.15](#).

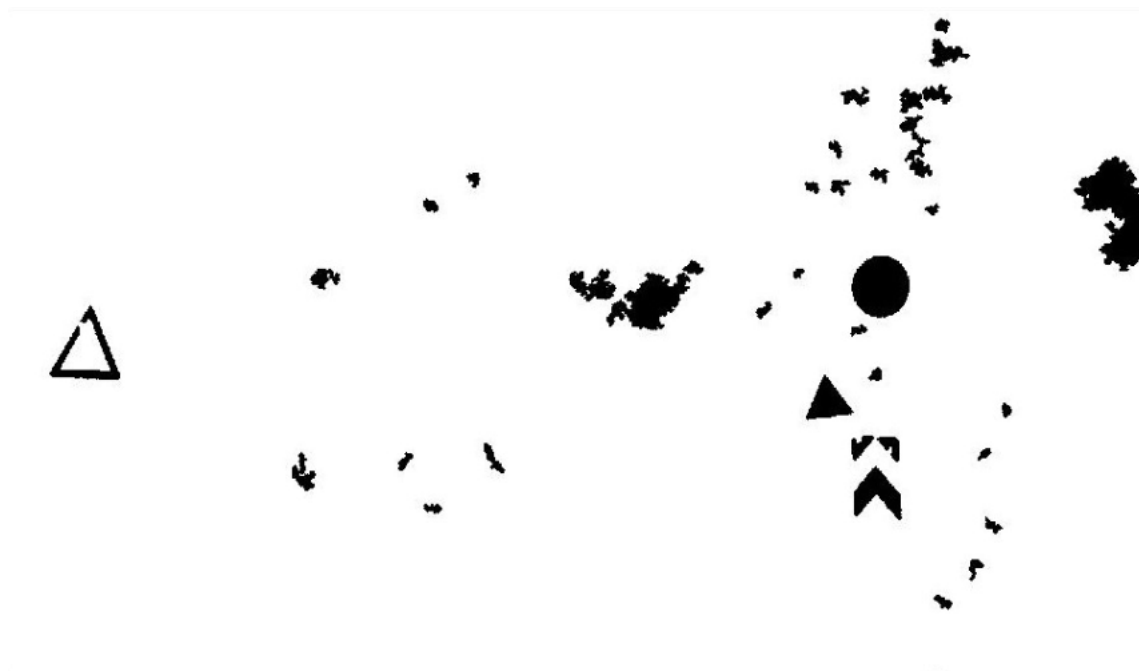


Figura 3.13: Processo de dilatação da imagem.

Fonte: Elaborado pelo Autor.



Figura 3.14: Aplicação do método de Canny para detecção de bordas.

Fonte: Elaborado pelo Autor.

3.3.2 Segmentação

Para a etapa de segmentação da região de interesse é realizada a varredura global da imagem utilizando a transformada Hough generalizada. Foram utilizados templates circulares com raios variando entre 10 *pixels* e 50 *pixels* com passo de 5 *pixels*, também

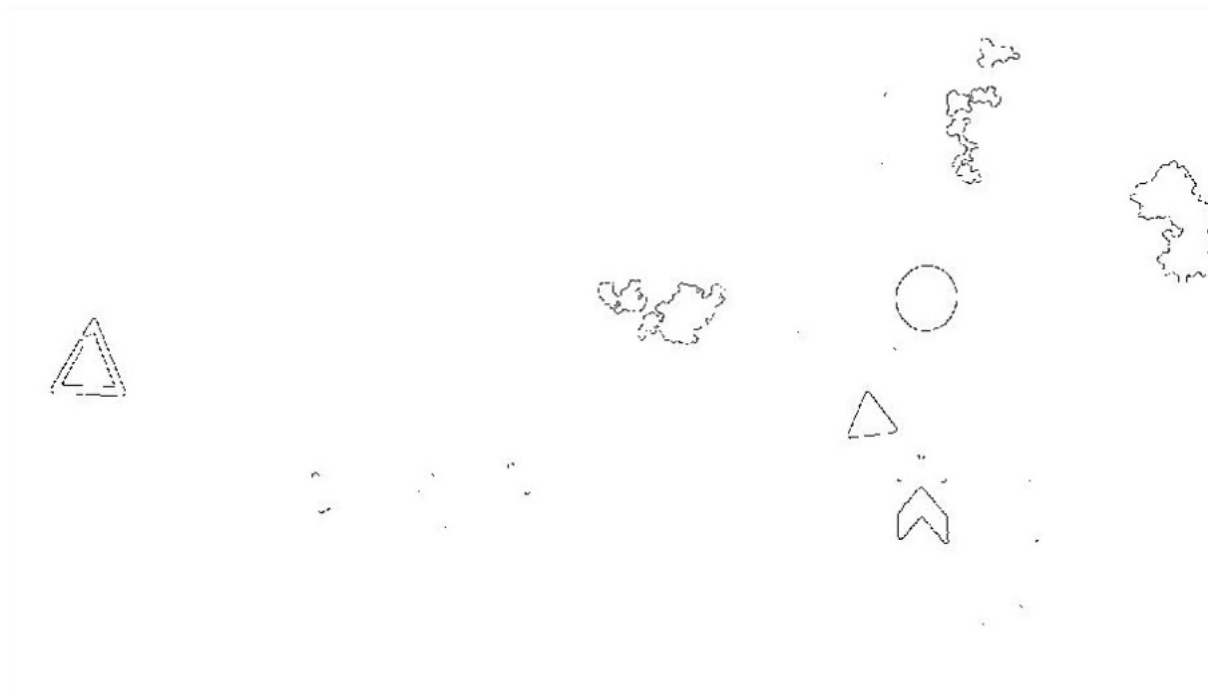


Figura 3.15: Remoção das bordas através do ponto de massa.

Fonte: Elaborado pelo Autor.

em conformidade aos tamanhos das placas de sinalização encontradas nas imagens do banco GTSDDB.

Com isso, a diferença entre os tamanhos dos templates proporciona um aumento significativo na probabilidade de detectar placas sinalização. Em um único movimento de aproximação, realizado com a câmera acoplada ao painel do veículo, imagens são obtidas de uma mesma placa, onde o tamanho desta é proporcional à distância entre a câmera e o ponto onde se encontra a placa de sinalização.

Ao realizar a convolução da imagem com um dos templates especificados, a localização da placa circular é obtida através de um limiar de votação. Os *pixels* centrais das placas possuem maior votação na imagem inteira caracterizados por valores de maior intensidade e por isso, obedecem a regra imposta pelo limiar indicando a suposta localização da placa de sinalização na imagem. Contudo, para cada template foi aplicado um limiar de votação diferente, uma vez que o perímetro de votação é diferente para cada um destes. A valores escolhidos para cada templates são apresentados pelo Teste F da [Tabela 3.3](#)

A placa de sinalização é então segmentada através do tamanho do template onde os valores dos *pixels* ultrapassam o limiar escolhido. É formada uma máscara quadrada com lado igual ao diâmetro do template e o produto resultante da multiplicação da imagem e a máscara é apresentada pela [Figura 3.16](#).



Figura 3.16: Segmentação da placa de sinalização.

Fonte: Elaborado pelo Autor.

3.3.3 Descrição

A etapa de descrição permite que a placa de sinalização segmentada seja descrita através de um vetor numérico. Para isso, é utilizada a técnica HOG, onde é realizada a junção de vetores gradientes e histogramas. Para realizar esta técnica, é necessário redimensionar (interpolação bicúbica) a imagem para um tamanho de 64x128. Em seguida, os gradientes das imagens são computados sobre todos os *pixels*, onde estes são divididos em células de tamanho 8x8 *pixels*.

De cada célula, é extraído um histograma de 9 posições, onde os valores de cada posição são computados através do ponderamento dos valores de magnitude em relação aos valores de direção dos vetores gradientes. Os valores de direção são considerados *unsigned*, fazendo com que as posições do histograma variem entre 0° à 180°.

Por fim, blocos contendo os valores de 4 histogramas (4 células, 16x16) são normalizados para diminuir a influência de iluminação. Cada subvetor (célula) do bloco é então concatenado sequencialmente com os demais, formando um vetor linha, do qual é denominado de descritor HOG. O resultado gráfico de cada subvetor, com sua célula correspondentes, é apresentado pela [Figura 3.18](#).

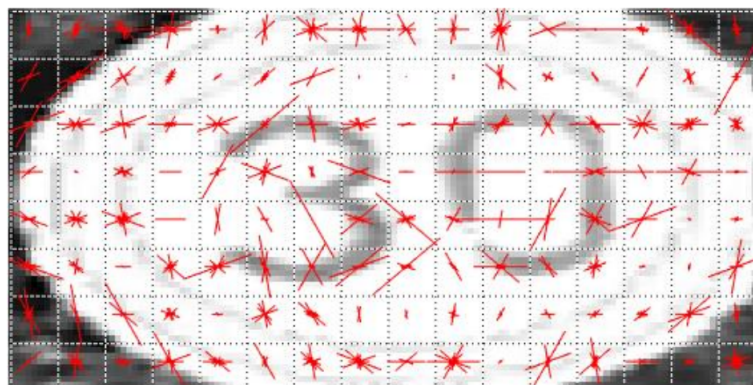


Figura 3.17: Descritor HOG sobre a imagem analisada.

Fonte: Elaborado pelo autor.

3.3.4 Classificação

A etapa de classificação envolve o treinamento e teste de Máquina de Vetores de Suporte (SVM). São realizados os treinamentos e testes de 11 SVM's para classificar as 11 classes de placas analisadas inicialmente. As SVM's obtidas são definidas pelos parâmetros apresentados pela [Tabela 3.1](#). Para o processo de treinamento foram utilizadas imagens positivas (contendo placas de sinalização) e imagens negativas (contendo placas de sinalização de classes não analisadas ou sem placa de sinalização), em igual proporção, conforme é apresentado pela [Tabela A.2](#) do Apêndice A.

Os resultados dos testes das SVM's são indicados através do rótulo 'P' para dados de entrada reconhecidos como pertencentes à classe em questão ou do rótulo 'N' para dados de entrada reconhecidos como não pertencentes à classe em questão. Cada SVM apresentou seus resultados independentemente das demais SVM's uma vez que estas foram aplicadas em paralelo aos dados de entrada.

| | |
|------------------|--|
| Dados de Entrada | 1x3780 |
| Nome das Classes | {'N' 'P'} |
| Função Kernel | $G(x_j, x_k) = x_j \cdot x_k$ (Linear) |

Tabela 3.1: Parâmetros de definição das SVM's

Fonte: Elaborado pelo Autor.

3.4 Cenário de Testes

Para validar o comportamento do sistema com placas de sinalização pertencentes ao sistema brasileiro de trânsito, foram coletadas imagens de placas que possuem grande semelhança com as placas de sinalização anteriormente aplicadas ao sistema através do banco de imagens GTSDDB. Para isso foi utilizada a câmera IP do modelo Intelbras VIP 4120 configurada para obter imagens coloridas e com resolução HD.

A câmera foi acoplada ao painel de um veículo e conectada ao sistema através de uma conexão *Ethernet* ponto a ponto, conforme é apresentado pelas [Figuras 3.19 e 3.20](#). A câmera Intelbras VIP 4120 possui entre suas características, a capacidade de obter fotos do ambiente à cada 500 ms através de um comando HTTP como é apresentado pela [Figura 3.21](#). O comando HTTP foi integrado diretamente à plataforma MATLAB[®] que armazena as fotos na memória do sistema.

Para o experimento em questão, foram obtidas imagens de quatro classes de placas de sinalização do sistema brasileiro de trânsito, das quais foram escolhidos em decorrência das semelhanças entre estas e as placas do banco GTSDDB. As classes escolhidas, bem como a quantidade de imagens obtidas de cada classe são apresentadas na seção 3.5.

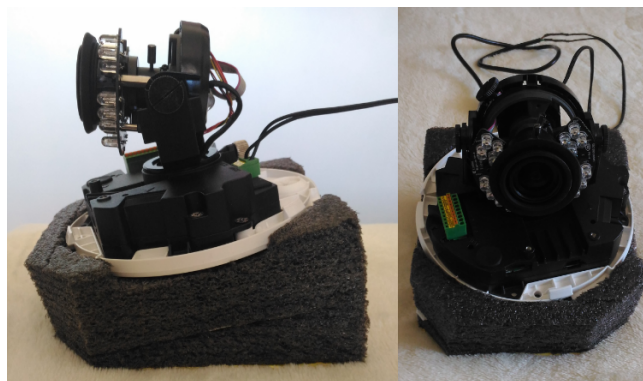


Figura 3.19: Suporte desenvolvido para acoplar a câmera ao painel do veículo.

Fonte: Elaborado pelo Autor.



Figura 3.20: Câmera acoplada ao painel frontal do veículo.

Fonte: Elaborado pelo Autor.



Figura 3.21: Amostra das imagens obtidas pela câmera Intelbras VIP 4120.

Fonte: Elaborado pelo Autor.

3.5 Resultados

Os parâmetros utilizados para os módulos do algoritmo foram obtidos através de testes realizados em conformidade com a literatura de cada técnica empregada em conjunto com as características das placas de sinalização de trânsito utilizadas. Os parâmetros definidos pelos testes e que afetam diretamente o reconhecimento das placas de sinalização são apresentados pela [Tabela 3.2](#). Ao longo desta seção, estes parâmetros são descritos e relacionados principalmente às etapas de pré-processamento e segmentação. Ao fim desta sessão, são analisadas as taxas de acertos obtidas na etapa de classificação em decorrência do uso destes parâmetros.

| | |
|---|---------------------|
| Limiars do Matiz | 338°-21° |
| Limiar de saturação | 225 |
| Raio de Elemento Estruturante de Erosão | 1 <i>pixel</i> |
| Raio do Elemento Estruturante de Dilatação | 2 <i>pixels</i> |
| Raios Limitantes para Remoção de Áreas e Bordas | 10-50 <i>pixels</i> |

Tabela 3.2: Parâmetros de definição do algoritmo

Fonte: Elaborado pelo Autor.

3.5.1 Pré-processamento

A primeira parte da análise do algoritmo proposto resume-se em alterar os parâmetros das etapas do módulo pré-processamento. Para este módulo, são alterados os limiars do matiz, saturação e os raios dos elementos estruturantes dos processos morfológicos de erosão e dilatação da imagem. Ao alterar estes valores é avaliado a quantidade de *pixels* pertencentes à placa de sinalização e a quantidade de falsos positivos caracterizados por *pixels* sem relação alguma com a placa de sinalização. Esta relação é analisada entre a imagem original da [Figura 3.3](#) e dos resultados das etapas precedentes ao processo em questão.

Inicialmente, são alterados os limiars mínimo e máximo da limiarização do matiz de forma a estreitá-los e, conseqüentemente, obter apenas cores vermelhas características da área da borda das placas de sinalização analisadas. Os resultados obtidos através da análise sobre a imagem original da [Figura 3.3](#) para o estreitamento destes limiars do matiz são apresentados pelo gráfico da [Figura 3.22](#). Nesta figura, os valores são normalizados entre 0 e 1, dos quais variam ao passo 0,01 em direção ao valor central.

É possível observar através do gráfico que a perda dos *pixels* pertencentes à placa de sinalização, caracterizada por uma quantidade de 1461 *pixels*, acontece exatamente após o estreitamento da faixa além dos valores 0.94 e 0.06. A diminuição desta área afeta negativamente a detecção da placa, uma vez que altera as características que serão utilizadas para o treinamento posterior das SVM's.

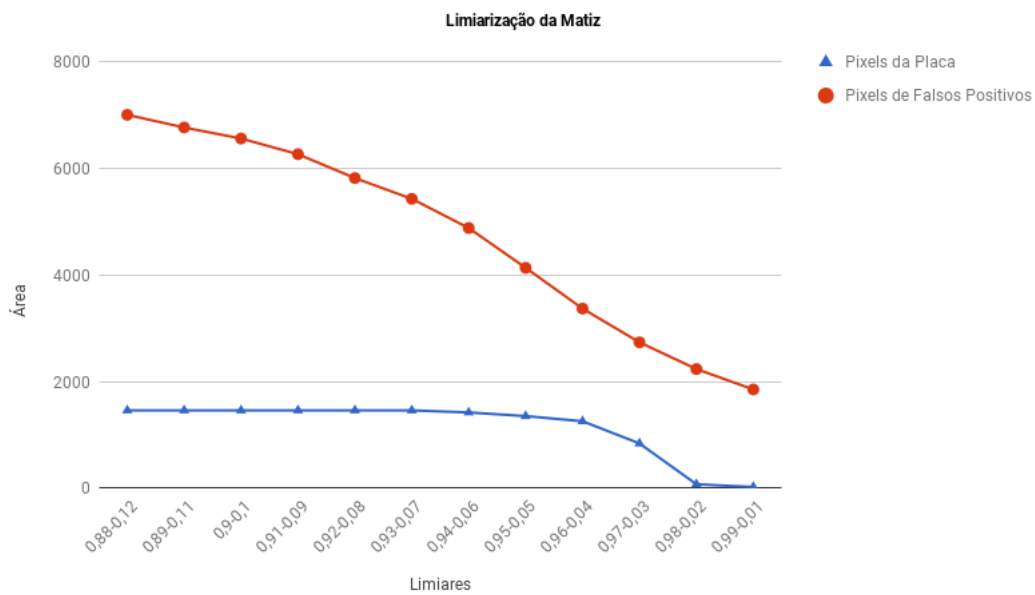


Figura 3.22: Resultados da limiarização da matiz.

Fonte: Elaborado pelo Autor.

Através desta análise, os valores mencionados anteriores são escolhidos para os limiares mínimo e máximo da subetapa de limiarização do matiz, evitando a perda significativa dos *pixels* pertencentes à placa de sinalização e reduzindo 22,5% os *pixels* dos falsos positivos em relação à imagem original.

De mesmo modo, os resultados obtidos através limiarização da saturação são apresentados pelo gráfico da [Figura 3.23](#). O limiar de saturação é alterado de forma crescente, entre os valores de 180 e 255 com passo de 5, que corresponde as cores de maior intensidade.

É possível observar através do gráfico que a perda dos *pixels* pertencentes à placa de sinalização nesta subetapa acontece a partir do limiar de valor 225. Através desta análise, este valor é escolhido como limiar fixo desta etapa, reduzindo 42,1% dos *pixels* dos falsos positivos em relação à imagem original.

Após o produto das imagens resultantes dos processos de limiarização do matiz e saturação, a imagem binária obtida já apresenta um nível de realce relativamente alto, convertendo para o fundo da imagem 65% dos *pixels* da imagem original e restando apenas 1.34% dos *pixels* pertencentes à placa analisada e 33% dos *pixels* pertencentes aos falsos positivos.

Para o processo morfológico de erosão, é considerada a área total interna da placa de sinalização. A variação do raio do elemento estruturante circular para o processo de erosão acontece entre os valores de 0 e 10 *pixels* ao passo de 1. Este processo não implica em alterações significativas para a área pertencente à placa de sinalização, porém os *pixels*

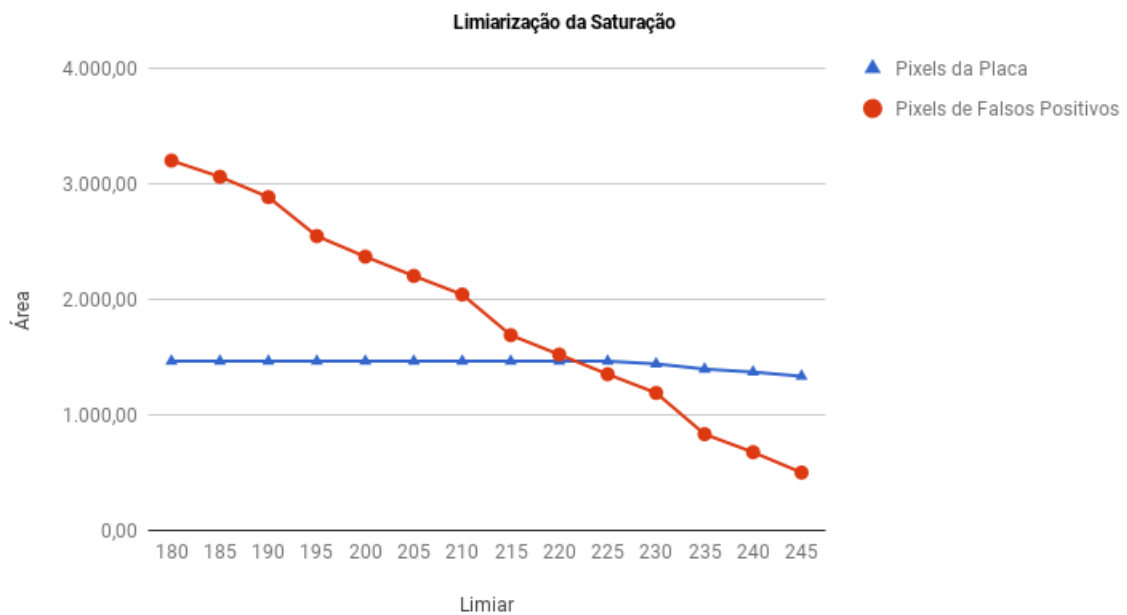


Figura 3.23: Resultados da limiarização da saturação.

Fonte: Elaborado pelo Autor.

pertencentes aos falsos positivos são reduzidos consideravelmente de forma exponencial com o aumento do raio.

Contudo, considerando que a placa de menor raio (10 *pixels*) pertencente ao banco de imagens GTSDDB possui uma borda de aproximadamente 2 *pixels*, suportando uma variação máxima de área de 113 *pixels*, o processo de erosão com elementos estruturantes muito grandes se torna inviável, uma vez que afeta a taxa de placas segmentadas com sucesso.

Com isso, o raio de 1 *pixel*, é escolhido para o elemento estruturante do processo morfológico de erosão, uma vez que este proporciona uma redução de apenas 5% da área total pertencente à placa de sinalização em questão e uma redução de 66% de falsos positivos em relação as características apresentadas pela imagem da Figura 3.9. Em relação a imagem original da Figura 3.3, a remoção dos *pixels* não pertencentes à placa de sinalização foi de 73,5%, restando apenas 3,3% de *pixels* pertencentes à placa de sinalização e apenas 23,2% dos *pixels* pertencentes aos falsos positivos.

Semelhantemente, para o processo de dilatação é alterado o raio do elemento estruturante, de forma também crescente, entre os valores de 1 e 10 ao passo de 1. Entretanto, o processo de dilatação interfere diretamente nas bordas dos objetos da imagem obtidas através do método de detecção de Canny, e por isso é analisado a quantidade de *pixels* pertencentes às bordas da placa de sinalização na imagem da Figura 3.15.

Os *pixels* pertencentes à borda da placa de sinalização não sofreram mudanças

significativas. Entretanto, a quantidade de *pixels* pertencentes às demais bordas sofrem mudanças não lineares e por isso, não foi obtido um valor concreto através da análise realizada. Entretanto, através de testes empíricos, o valor 2 para o raio do elemento estruturante apresenta boa performance para esta subetapa, uma vez que é constatado que elementos estruturantes com raio de grande valor para o processo morfológico de dilatação podem fazer com que as bordas das placas de menor tamanho se agrupem com outras bordas, interferindo no processo de remoção através das áreas e centros de massa.

3.5.2 Segmentação

Para a etapa de segmentação, é analisado a quantidade de placas de sinalização corretamente segmentadas das imagens em comparação com a quantidade total de placas de sinalização contidas no banco de imagens GTSDDB. Foram ajustados os limiares de cada template utilizado pela Transformada Hough Generalizada, de maneira a maximizar a quantidade de placas de sinalização segmentadas.

Decidimos por investigar o efeito da alteração dos limiares correspondentes à cada template, partindo de um valor fixo para todos os limiares até uma variação linear máxima, resultando nos testes apresentados pela Tabela 3.3. Os valores máximos e mínimos para os limiares foram escolhidos através do cálculo do perímetro dos templates de menor e maior tamanho, respectivamente.

| Diâmetro dos Templates (px) | Limiares | | | | | |
|-----------------------------|----------|---------|---------|---------|---------|---------|
| | Teste A | Teste B | Teste C | Teste D | Teste E | Teste F |
| 20 | 80 | 60 | 50 | 40 | 40 | 45 |
| 25 | 80 | 60 | 50 | 40 | 45 | 45 |
| 30 | 80 | 60 | 50 | 50 | 50 | 45 |
| 35 | 80 | 60 | 50 | 50 | 55 | 55 |
| 40 | 80 | 60 | 70 | 60 | 60 | 60 |
| 45 | 80 | 60 | 70 | 60 | 65 | 65 |
| 50 | 80 | 60 | 70 | 70 | 70 | 70 |
| 55 | 80 | 60 | 70 | 70 | 75 | 75 |
| 60 | 80 | 100 | 90 | 80 | 80 | 80 |
| 65 | 80 | 100 | 90 | 80 | 85 | 85 |
| 70 | 80 | 100 | 90 | 90 | 90 | 85 |
| 75 | 80 | 100 | 90 | 90 | 95 | 90 |
| 80 | 80 | 100 | 110 | 100 | 100 | 100 |
| 85 | 80 | 100 | 110 | 100 | 105 | 100 |
| 90 | 80 | 100 | 110 | 110 | 110 | 105 |
| 95 | 80 | 100 | 110 | 110 | 115 | 105 |
| 100 | 80 | 100 | 110 | 110 | 115 | 105 |

Tabela 3.3: Inclinação dos limiares da transformada Hough generalizada.

Fonte: Elaborado pelo Autor.

Através da análise dos limiares em comparação com os resultados obtidos, é constatado que os testes que apresentam melhor performance são os de maior linearidade. O motivo pelo qual isto acontece está relacionado ao fato de que os templates com tamanhos diferentes possuem limiares de votações proporcionais ao perímetro circular de cada tem-

plate. O template com menor tamanho possui a votação máxima de 63 *pixels*, enquanto que o template de maior tamanho possui a votação máxima de 314 *pixels*. Entretanto, são utilizados limiares com 30% a menos aos valores calculado, devido às imperfeições que as bordas das placas de sinalização podem apresentar.

A implicação disto é apresentada pela [Tabela 3.4](#) que demonstra a quantidade de placas de sinalização corretamente segmentadas para cada teste. É possível observar o aumento significativo dos valores entre o teste A com limiares fixos e a teste F com limiares lineares.












| |  |  |  |  |  |  |  |  |  |  |  |
|--------------|---|---|---|---|---|--|---|---|---|---|---|
| Total | 10 | 4 | 16 | 34 | 9 | 20 | 30 | 4 | 9 | 17 | 9 |
| A (%) | 20 | 50 | 44 | 18 | 22 | 45 | 43 | 75 | 33 | 0 | 22 |
| B (%) | 50 | 50 | 75 | 50 | 33 | 60 | 60 | 25 | 89 | 35 | 56 |
| C (%) | 70 | 75 | 56 | 50 | 56 | 70 | 73 | 75 | 78 | 65 | 44 |
| D (%) | 70 | 75 | 88 | 65 | 67 | 70 | 73 | 75 | 89 | 71 | 78 |
| E (%) | 70 | 75 | 75 | 59 | 56 | 70 | 73 | 75 | 89 | 53 | 67 |
| F (%) | 90 | 75 | 94 | 76 | 67 | 80 | 73 | 75 | 100 | 71 | 89 |

Tabela 3.4: Resultados da etapa de segmentação.

Fonte: Elaborado pelo Autor.

A utilização do limiares no teste F apresenta maior quantidade de placas de sinalização segmentadas corretamente. As quantidades de placas que foram segmentadas por esta função em relação ao total de placas contidas no banco GTSDDB são superiores à 70% em todas as 11 classes, comprovando o aumento de performance.

3.5.3 Classificação

Para a etapa de classificação, é analisada a quantidade de placas de sinalização corretamente classificadas através das SVM's que são aplicadas em paralelo a cada classes, conforme é apresentado pela [Tabela 3.5](#). É possível observar na 4ª linha desta tabela, que do total de placas segmentadas através do teste F do processo anterior, quantidades superiores à 67% foram reconhecidas com sucesso, atingindo em muitos casos taxas de acerto de 100%.

Entretanto, devido aos limiares e os templates da Transformada Hough Generalizada, uma provável região contendo uma placa de sinalização foi segmentada repetidamente, fazendo com que o processo de classificação também seja executado diversas vezes para uma mesma imagem. Por isso, as taxas de acertos são divididas entre placas reconhecidas, do qual indicam o reconhecimento individual de cada placa e acertos totais,

| |  |  |  |  |  |  |  |  |  |  |  |
|-------------------------------|---|---|---|---|---|--|---|---|---|---|---|
| Total | 10 | 4 | 16 | 34 | 9 | 20 | 30 | 4 | 9 | 17 | 9 |
| Segmentadas | 9 | 3 | 15 | 26 | 6 | 16 | 22 | 3 | 9 | 12 | 8 |
| Placas Reconhecidas(%) | 67 | 67 | 100 | 92 | 83 | 100 | 100 | 100 | 100 | 100 | 88 |
| Acertos Totais (%) | 84 | 90 | 76 | 82 | 83 | 80 | 76 | 96 | 96 | 90 | 98 |
| Falsos Positivos (%) | 13 | 10 | 22 | 15 | 16 | 17 | 17 | 2 | 3 | 5 | 1 |
| Falsos Negativos (%) | 3 | 1 | 2 | 3 | 1 | 3 | 7 | 2 | 1 | 5 | 1 |

Tabela 3.5: Resultados da etapa de classificação

Fonte: Elaborado pelo Autor.

incluído classificações positivas sobre várias imagens de uma mesma placa. O número de classificações corretas nos acertos totais é indicado na 5ª linha da [Tabela 3.5](#).

Estes dois valores representam nas duas situações, as classificações positivas correspondentes à cada SVM que apresenta os rótulos das placas que realmente estão contidas nas regiões segmentadas e classificações negativas correspondentes a cada SVM que apresenta o rótulo negativo indicando que as regiões não apresentam nenhuma placa ou apresentam placas de outra classe.

Com um número grande de regiões segmentadas, o algoritmo também apresenta falsos positivos caracterizados por classificações positivas em regiões que não contêm a placa da classe em questão e falsos negativos caracterizados por classificações negativas em regiões que contêm a placa da classe em questão. A quantidade mais significativa neste caso é a de falsos positivos, onde até 22% das regiões que não contêm a placa analisada foram classificadas positivamente.

3.5.4 Placas de Sinalização Brasileiras

Por fim, o algoritmo completo foi aplicado às placas de sinalização do sistema de trânsito brasileiro, onde os resultados são apresentados pela [Tabela 3.6](#). A 2ª linha da tabela informa a quantidade total de placas capturadas pela câmera acoplada ao painel de um veículo em movimento para cada classe.

É possível observar através desta tabela que do total de placas de sinalização, quantidade superiores a 72% foram segmentadas corretamente. Do total de placas segmentadas, foram atingidas taxas de acerto de até 90% para o processo de classificação. Ambos os resultados se aproximam do que foi obtido da etapa anterior utilizando placas de sinalização do banco de imagens GTSDB.

Semelhantemente a etapa anterior, o processo de segmentação através da transformada Hough generalizada também gerou grande número de regiões segmentadas contendo





| Placas |  |  |  |  |
|-------------------------|---|---|--|---|
| Total | 22 | 15 | 6 | 11 |
| Placas Segmentadas (%) | 72,7 | 73,3 | 83,3 | 100,0 |
| Placas Reconhecidas (%) | 81,8 | 46,7 | 33,3 | 90,9 |
| Acertos Totais (%) | 72,7 | 78,5 | 93,1 | 86,5 |
| Falsos Positivos (%) | 3,0 | 2,4 | 1,6 | 0,2 |
| Falsos Negativos (%) | 24,1 | 19,1 | 5,3 | 13,3 |
| Total de Imagens | 1027 | | | |
| Com Placa | 906 | | | |
| Sem Placa | 121 | | | |

Tabela 3.6: Resultados do algoritmo aplicado a placas de sinalização brasileiras.

Fonte: Elaborado pelo Autor.

ou não placas de sinalização. Os acertos totais, apresentaram taxas de acerto superiores 72% do total de 1027 regiões segmentadas.

O algoritmo também apresentou falsos positivos e falsos negativos, sendo o valor mais significativo dado aos falsos negativos alcançando 24% do total de placas de sinalizações. Isto faz com que imagens que contêm a placa de sinalização analisada sejam classificadas negativamente pelas SVM's.

O motivo para o número de falsos negativos está na presença de características adicionais nestas placas, como textos informativos ("km/h") e a mudança da linguagem textual em certas placas. Para as SVM's estas informações se apresentam como ruídos, fazendo com que as placas não sejam identificadas corretamente.

Apesar destas diferenças, os resultados apresentados por esta etapa também se assemelham aos resultados apresentados no uso de imagens extraídas do banco GTSDDB. Além disso, as características das SVM's podem ser aprimoradas com o uso de um banco de imagens constituído de placas de sinalização do sistema brasileiro trânsito, do qual aumentaria consideravelmente as taxas de acerto para o cenário em questão.

Conclusão

Esta monografia apresentou o estudo e a implementação de um sistema de reconhecimento automático de placas de sinalização de trânsito em imagens estáticas obtidas através de um banco público de imagens do sistema alemão de trânsito. Além disso, para a segunda etapa do trabalho foi investigado também os resultados deste sistema sobre imagens de algumas classes de placas de sinalização do sistema brasileiro de trânsito, obtidas através de uma câmera acoplada ao painel dianteiro de um veículo em movimento. Para concepção da arquitetura do projeto foram analisadas técnicas e mecanismos de áreas como Processamento Digital de Imagem, Inteligência Artificial, Visão Computacional, Matemática, Estatística, etc.

O sistema apresentado é formado por quatro principais módulos: pré-processamento, segmentação, descrição e classificação. O primeiro módulo de pré-processamento foi definido através da junção de técnicas de limiarização no modelo de cores HSV, operações morfológicas e detectores de borda. Este módulo apresentou redução considerável dos pixels não pertencentes à placa de sinalização em questão, demonstrando alto nível de realce das características desta. O conjunto de técnicas empregadas para este módulo resultou na remoção de 73,5% destes *pixels* em relação à quantidade de *pixels* total da imagem original.

Para o módulo de segmentação, foi utilizada a transformada Hough generalizada para localizar o centro da placa de sinalização através da detecção de círculos na imagem com o uso de templates circulares. Em ambas as etapas utilizando placas de sinalização alemãs e brasileiras, este módulo apresentou taxas de acerto superiores à 70% para todas as classes analisadas em relação ao total de placas de sinalização, evidenciando o nível de precisão para o processo de segmentação.

No módulo de descrição foi utilizado o método HOG que capaz de representar com nitidez as principais características das placas de sinalização. O descritor apresentado como resultado deste módulo apresentou alta capacidade de invariância em relação às alterações de iluminação e rotação características de imagens nas duas etapas. Além disso, este módulo proporcionou uma redução significativa dos dados para a posterior etapa de classificação.

Para o último módulo, responsável pela classificação e rotulação dos dados, foi utilizado um conjunto de SVM's que foram treinadas e testadas em paralelo para cada classe de placa de sinalização analisada. Para a primeira etapa do sistema, este módulo alcançou taxas de acerto superiores à 70% do total de placas. Para a segunda etapa, este módulo alcançou taxas de acerto de até 90% das placas.

Através destes resultados, foi possível perceber a capacidade das SVM's para a realização de tarefas desta natureza. Deve-se levar em conta que, para a segunda etapa, as placas de sinalizações apresentam características adicionais como textos informativos e mudança de linguagem que não são abordadas no treinamento das SVM's, sendo assim consideradas como ruídos.

Para as duas etapas, o algoritmo apresentou também falsos positivos e falsos negativos, em relação aos acertos totais das regiões segmentadas. Para a primeira etapa, a quantidade mais expressiva foi de falsos positivos, atingindo até 22% do total. De maneira oposta, para a segunda etapa o valor mais expressivo foi de falsos negativos, atingindo até 24% do total.

Esta diferença acontece devido aos dois principais fatores que afetam diretamente as taxas de acerto do sistema. O primeiro fator é relacionado à grande quantidade de regiões segmentadas repetidamente através da transformada Hough generalizada, implicando no grande número de execuções do módulo de classificação para uma mesma imagem.

O segundo fator é relacionado com objetos nas imagens que possuem características semelhantes às placas de sinalização, como por exemplos, veículos de cor vermelha que possuem formato próximo à um círculo e árvores com alta incidência de sol. A existência destes objetos na imagem implica na segmentação de regiões sem placas de sinalização de trânsito.

Diante dos resultados experimentais obtidos nos módulos mencionados e das limitações apresentadas pelo algoritmo proposto, pode-se considerar que o objetivo do trabalho de reconhecer automaticamente placas de sinalização de trânsito foi alcançado. Entretanto, é importante salientar que aumento significativo nas taxas de acerto do sistema pode ser alcançado com o aprimoramento das técnicas envolvidas em cada módulo, bem como o desenvolvimento de dados de treinamento adequados para o cenário em questão.

Com isso, sugere-se uma série de ações apresentadas abaixo que são voltadas para trabalhos futuros e possuem como objetivo aumentar a eficácia do sistema, atingindo taxas de acerto relativamente altas e possibilitando a utilização do sistema em aplicações reais.

Trabalhos futuros

- Desenvolver um banco de imagens público com somente placas de sinalização do sistema brasileiro de trânsito;
- Aprofundar-se na análise e na escolha do *kernel* para as SVM's adequado para o

cenário de classificação de placas de sinalização de trânsito;

- Realizar análises sobre influências de fatores como iluminação, resolução da imagem, exposição, velocidade do obturador, aquisição da imagem, etc, sobre os resultados apresentados;
- Desenvolver uma abordagem utilizando técnicas de processamento de imagens e aprendizado de máquina junto com sistemas de GPS e de Banco de Dados também apropriados para o cenário do reconhecimento de placas de sinalização de trânsito;
- Investigar o uso de *hardware* dedicados para o algoritmo proposto e a sua adaptabilidade em veículos;
- Investigar e comparar o algoritmo proposto com outros trabalhos;
- Investir técnicas e ferramentas de aquisição de imagens que mais se adaptam ao cenário em questão;

Por fim, espera-se ter com esta monografia, contribuído para um melhor entendimento de sistemas deste tipo e suas principais falhas, para que estas sejam superadas com o intuito de tornar o reconhecimento automático de placas de sinalização de trânsito cada vez mais viáveis e de forma eficiente melhorar a mobilidade e segurança de seus usuários.

Referências

- ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2014. Citado 4 vezes nas páginas 37, 38, 39 e 40.
- AUDI. *The driver assistance systems and integrated safety*. 2017. Disponível em: <goo.gl/x98tMK>. Acesso em: 4 de Junho de 2017. Citado na página 16.
- BALALI, V.; GOLPARVAR-FARD, M. Evaluation of multiclass traffic sign detection and classification methods for us roadway asset inventory management. *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, v. 30, n. 2, p. 04015022, 2015. Citado na página 36.
- BALLARD, D. H. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, Elsevier, v. 13, n. 2, p. 111–122, 1981. Citado na página 32.
- BAY, H. et al. Speeded-up robust features (surf). *Computer vision and image understanding*, Elsevier, v. 110, n. 3, p. 346–359, 2008. Citado na página 34.
- BEGG, R.; PALANISWAMI, M. *Computational intelligence for movement sciences: neural networks and other emerging techniques*. [S.l.]: IGI Global, 2006. Citado 2 vezes nas páginas 41 e 42.
- BOSCH. *Driver Assistance Systems*. 2017. Disponível em: <goo.gl/FAcBfz>. Acesso em: 2 de Junho de 2017. Citado na página 16.
- CAMASTRA, F.; VINCIARELLI, A. Machine learning for audio, image and video analysis. *Advanced Information and Knowledge Processing*, Springer, p. 83–89, 2008. Citado 2 vezes nas páginas 37 e 38.
- CAMILO, C. O.; SILVA, J. C. da. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. *Universidade Federal de Goiás (UFG)*, p. 1–29, 2009. Citado na página 37.
- CAMPBELL, C.; YING, Y. Learning with support vector machines. *Synthesis lectures on artificial intelligence and machine learning*, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–95, 2011. Citado 3 vezes nas páginas 41, 42 e 43.
- CASTRO, C.; HORBERRY, T. *The human factors of transport signs*. [S.l.]: CRC press, 2004. Citado na página 15.
- CHAKI, N.; SHAIKH, S. H.; SAEED, K. Exploring image binarization techniques. Springer, 2014. Citado na página 26.
- CHANDRASEKHAR, V. et al. Transform coding of image feature descriptors. In: *VCIP*. [S.l.: s.n.], 2009. p. 1. Citado na página 33.
- CHEN, Z.; YANG, J.; KONG, B. A robust traffic sign recognition system for intelligent vehicles. *Sixth International Conference on Image and Graphics*, 2011. Citado na página 17.

- CONTRAN. *Sinalização vertical de regulamentação*. [S.l.], 2007. Disponível em: <<https://goo.gl/2eVgwA>>. Acesso em: 2 de Outubro de 2017. Citado na página 15.
- CORSO, J. J.; HAGER, G. D. Image description with features that summarize. *Computer Vision and Image Understanding*, Elsevier, v. 113, n. 4, p. 446–458, 2009. Citado na página 33.
- DALAL, N.; DALAL, B. Histograms of oriented gradients for human detection. In: IEEE. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. [S.l.], 2005. v. 1, p. 886–893. Citado 3 vezes nas páginas 34, 35 e 36.
- DOUGHERTY, G. *Digital image processing for medical applications*. [S.l.]: Cambridge University Press, 2009. Citado 5 vezes nas páginas 22, 23, 24, 26 e 36.
- FERNÁNDEZ, A. et al. Real-time pattern recognition using an optical generalized hough transform. *Applied optics*, Optical Society of America, v. 54, n. 36, p. 10586–10591, 2015. Citado na página 32.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado 5 vezes nas páginas 21, 22, 23, 27 e 28.
- GAO, H. et al. Traffic signs recognition based on pca-sift. In: IEEE. *Intelligent control and automation (WCICA), 2014 11th world congress on*. [S.l.], 2014. p. 5070–5076. Citado na página 17.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento digital de imagens. tradução: Cristina yamagami e leonardo piemonte*. [S.l.]: Pearson Prentice Hall, São Paulo, 2010. Citado 12 vezes nas páginas 17, 20, 21, 22, 23, 24, 27, 29, 30, 31, 33 e 34.
- GOYAL, M. Morphological image processing. *IJCST*, v. 2, n. 4, 2011. Citado 2 vezes nas páginas 27 e 28.
- GREENHALGH, J.; MIRMEHDI, M. Real-time detection and recognition of road traffic signs. *IEEE Transactions on Intelligent Transportation Systems*, v. 13, n. 4, p. 1498–1506, 2012. Citado na página 17.
- HASSABALLAH, M.; ABDELMGEID, A. A.; ALSHAZLY, H. A. Image features detection, description and matching. In: *Image Feature Detectors and Descriptors*. [S.l.]: Springer, 2016. p. 11–45. Citado na página 33.
- HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2007. Citado 4 vezes nas páginas 39, 40, 41 e 42.
- HORNBERG, A. *Handbook of machine vision*. [S.l.]: John Wiley & Sons, 2007. Citado 7 vezes nas páginas 20, 21, 22, 24, 29, 33 e 34.
- HOUBEN, S. et al. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: IEEE. *Neural Networks (IJCNN), The 2013 International Joint Conference*. [S.l.], 2013. p. 1–8. Citado 3 vezes nas páginas 19, 45 e 46.
- HYUNDAI. *Advanced Driver Assistance System: Going beyond convenience for the driver to rider safety*. 2017. Disponível em: <goo.gl/Lse8ed>. Acesso em: 6 de Junho de 2017. Citado na página 16.

- ILLINGWORTH, J.; KITTLER, J. A survey of the hough transform. *Computer vision, graphics, and image processing*, Elsevier, v. 44, n. 1, p. 87–116, 1988. Citado na página 31.
- JAHNE, B. *Digital image processing*. 6. ed. [S.l.: s.n.], 2005. Citado 4 vezes nas páginas 21, 22, 31 e 34.
- KRIG, S. *Computer vision metrics*. [S.l.]: Springer, 2014. Citado 2 vezes nas páginas 26 e 30.
- LIN, F. et al. A traffic sign recognition method based on deep visual feature. In: IEEE. *Progress in Electromagnetic Research Symposium (PIERS)*. [S.l.], 2016. p. 2247–2250. Citado na página 16.
- LORENA, A. C.; CARVALHO, A. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado 3 vezes nas páginas 39, 42 e 43.
- LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157. Citado na página 33.
- MĂRIUȚ, F. et al. Detection and recognition of traffic signs using gabor filters. *34th International Conference on Telecommunications and Signal Processing*, p. 554–558, 2011. Citado na página 17.
- MATHIAS, M. et al. Traffic sign recognition—how far are we from the solution? In: IEEE. *Neural Networks (IJCNN), The 2013 International Joint Conference on*. [S.l.], 2013. p. 1–8. Citado na página 45.
- MOGELMOSE, A.; TRIVEDI, M. M.; MOESLUND, T. B. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 13, n. 4, p. 1484–1497, 2012. Citado na página 45.
- MONARD, M.; BARANAUSKAS, J. Conceitos sobre aprendizado de máquina de sistemas inteligentes: Fundamentos e aplicações. *Conceitos sobre aprendizado de máquinas em Sistemas Inteligentes: Fundamentos e Aplicações*, Editora Manole, 2003. Citado 2 vezes nas páginas 37 e 38.
- MÜLLER, A. C.; GUIDO, S. *Introduction to machine learning with Python: a guide for data scientists*. [S.l.]: O’Reilly Media Sebastopol, CA, 2016. Citado na página 36.
- ROMDHANE, N. B.; MLIKI, H.; HAMMAMI, M. Real-time detection and recognition of road traffic signs. *15th International Conference on Computer and Information Science (ICIS)*, p. 1–6, 2016. Citado na página 18.
- SANTOS, E. M. *Teoria e aplicação de support vector machines à aprendizagem e reconhecimento de objetos baseado na aparência*. Tese (Doutorado) — Universidade Federal da Paraíba, 2002. Citado na página 42.
- SANTOS, T. S. Histograma de gradientes orientados na detecção de motocicletas. *IV Jornada de Informática do Maranhão*, 2012. Citado na página 35.

- Seguradora Líder . *Boletim Estatístico*. 2015. Disponível em: <<http://bit.ly/2hJJpDm>>. Acesso em: 2 de Outubro de 2017. Citado na página 16.
- SHAIK, K. B. et al. Comparative study of skin color detection and segmentation in hsv and ycbcr color space. *Procedia Computer Science*, Elsevier, v. 57, p. 41–48, 2015. Citado na página 25.
- SILVEIRA, J.; RITA, C. A imagem: interpretação e comunicação. *Linguagem em (Dis)curso*, v. 5, p. 113–128, 2010. Citado na página 20.
- SLOUGHTER, D. The calculus of functions of several variables. 2001. Citado na página 42.
- SMOLA, A. J.; SCHÖLKOPF, B. *Learning with kernels*. [S.l.]: GMD-Forschungszentrum Informationstechnik, 1998. Citado 2 vezes nas páginas 39 e 43.
- SOLOMON, C.; BRECKON, T. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. [S.l.]: John Wiley & Sons, 2011. Citado 9 vezes nas páginas 20, 21, 23, 24, 25, 26, 27, 28 e 30.
- SULEIMAN, A.; SZE, V. Energy-efficient hog-based object detection at 1080hd 60 fps with multi-scale support. In: IEEE. *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*. [S.l.], 2014. p. 1–6. Citado na página 35.
- TANIMOTO, S. L. *An interdisciplinary introduction to image processing: pixels, numbers, and programs*. [S.l.]: MIT Press, 2012. Citado 4 vezes nas páginas 22, 25, 29 e 31.
- TESLA. *Full Self-Driving Hardware on All Cars*. 2017. Disponível em: <<https://www.tesla.com/autopilot>>. Acesso em: 1 de Junho de 2017. Citado na página 16.
- TOMASI, C. *Histograms of Oriented Gradients*. [S.l.]: Diakses, 2015. Citado na página 35.
- VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013. Citado 2 vezes nas páginas 40 e 41.
- VERNON, D. *Machine vision-Automated visual inspection and robot vision*. [S.l.: s.n.], 1991. Citado 3 vezes nas páginas 20, 21 e 32.
- WHO. A road safety technical package. *World Health Organization*, 2017a. Disponível em: <goo.gl/fmaHPf>. Acesso em: 25 de Novembro de 2017. Citado na página 15.
- WHO. Managing speed. *World Health Organization*, 2017b. Disponível em: <goo.gl/1XeFj2>. Acesso em: 2 de Outubro de 2017. Citado na página 15.

Apêndices

APÊNDICE A – Configuração Utilizada para a Implementação do Sistema

| | |
|-------------------------|------------------------------------|
| Versão MATLAB® | R2015a |
| Processador | Intel(R) Core (TM) i7-4500U 1.8GHz |
| Memória RAM | 8 GB |
| Sistema Operacional | Ubuntu LTS 14.04 64 Bits |
| Modelo da Câmera | Intelbras VIP 4120 |
| Resolução | 1280x720 |
| Taxa de Frames | 30 fps |
| Velocidade do Obturador | 1/250s |
| Formato de Vídeo | NTSC |
| Interface | RJ45 (10/100Base-T) |
| Comando HTTP | 'http://ip/cgi-bin/snapshot.cgi?1' |

Tabela A.1: Configuração do sistema.

Fonte: Elaborado pelo Autor.












| Placas |  |  |  |  |  |  |  |  |  |  |  |
|----------|---|---|---|---|---|---|--|---|---|---|---|
| P | 47 | 37 | 37 | 31 | 21 | 58 | 48 | 25 | 32 | 63 | 22 |
| N | 46 | 42 | 40 | 35 | 27 | 58 | 46 | 28 | 31 | 59 | 25 |

Tabela A.2: Quantidade de imagens de treinamento

Fonte: Elaborado pelo Autor.