

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

MELISA DA SILVA CERQUEIRA

**RECONHECIMENTO AUTOMÁTICO DE PLACAS DE SINAIS DE TRÂNSITO  
BRASILEIRO**

TRABALHO DE CONCLUSÃO DE CURSO

Campos dos Goytacazes, RJ

07 de dezembro de 2022

# **RECONHECIMENTO AUTOMÁTICO DE PLACAS DE SINAIS DE TRÂNSITO BRASILEIRO**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Ciência da  
Computação da Universidade Estadual do  
Norte Fluminense Darcy Ribeiro como  
requisito para a obtenção do título de Bacharel  
em Ciência da Computação, sob orientação de  
Prof. Luis Antonio Rivera Escriba.

Universidade Estadual do Norte Fluminense Darcy Ribeiro – UENF

Centro de Ciência e Tecnologia – CCT

Laboratório de Ciências Matemáticas – LCMAT

Curso de Ciência da Computação

MELISA DA SILVA CERQUEIRA

**RECONHECIMENTO AUTOMÁTICO DE PLACAS DE SINAIS DE TRÂNSITO  
BRASILEIRO**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Ciência da Computação  
da Universidade Estadual do Norte Fluminense  
Darcy Ribeiro como requisito para a obtenção do  
título de Bacharel em Ciência da Computação,  
sob orientação de Prof. Luis Antonio Rivera  
Escriba.

Trabalho aprovado. Campos dos Goytacazes, RJ, 07 de dezembro de 2022:

---

Prof. Luis Antonio Rivera Escriba  
Orientador

---

Fermín Alfredo Tang Montané  
Membro da Banca

---

Ausberto Silverio Castro Vera  
Membro da Banca

Campos dos Goytacazes, RJ 07 de dezembro de 2022

## **AGRADECIMENTOS**

Gostaria de agradecer imensamente aos meus pais, Helma e Geraldo (in memoriam), meu esposo, Pedro Henrique e minha irmã, Fernanda, pelo suporte e incentivo durante todo o período do curso, tornando esta jornada mais fácil de ser cumprida. Vocês são minha base.

Agradeço também a todos os meus familiares e amigos que estiveram presentes pelo apoio nos momentos difíceis.

Agradeço ao meu orientador Luis Rivera pelos ensinamentos e paciência durante a realização deste trabalho e aos professores que fizeram parte da minha formação.

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

*“Descobrir consiste em olhar para o que todo mundo  
está vendo e pensar uma coisa diferente.”*  
(Roger Von Oech)

## RESUMO

A classificação de placas de sinalização é um problema que vem sendo mais abordado na atualidade, possuindo diversas aplicações e pode ser aplicado no trânsito brasileiro para auxiliar os motoristas no reconhecimento de placas de sinais de trânsito em tempo real. Atualmente existe um grande número de pesquisas abordando este assunto, trazendo muitas ramificações no que diz respeito a detecção de sinalização. O modelo proposto apresenta um sistema de reconhecimento automático de placas de sinais de trânsito baseado em janelas deslizantes. O sistema é dividido nas seguintes etapas: aquisição de imagens, pré-processamento com o uso do modelo de cores HSV e limiarização; ROI através de detectores de borda; extração de características através do uso do método Histograma de Gradientes Orientados (HOG) e classificação com o uso de Máquinas de Vetores de Suporte (SVM's). O desenvolvimento dos processos de treinamento e teste foram implementados através de bibliotecas *OpenCV* e *Dlib* e foram utilizadas imagens estáticas contendo placas de sinalização de trânsito obtidas através de um aparelho celular e o banco de imagens GSTDB. O algoritmo proposto alcançou taxas de acerto de aproximadamente 91% para o reconhecimento das placas de sinalização na etapa de testes. Para o reconhecimento, apesar das características diferentes apresentadas pelas placas de sinalização, as taxas de acertos alcançaram cerca de 89% para o reconhecimento das placas.

**Palavras-chave:** HOG. SVM. HSV. Placas de Trânsito.

## **ABSTRACT**

The classification of road signs is a problem that has been more faithful nowadays, having several applications and can be applied in Brazilian traffic to help drivers in the recognition of road signs in real time. Currently there is a large amount of research addressing this subject, bringing many ramifications with regard to signaling detection. The standard model features an automatic traffic sign recognition system based on sliding windows. The system is divided into the following steps: image acquisition, pre-processing using the HSV color model and thresholding; ROI through edge detectors; Generation of features using the Oriented Gradient Histogram (HOG) method and classification using Support Vector Machines (SVM's). The development of training and testing processes were implemented through OpenCV and Dlib libraries and static images containing traffic signs passing through a cell phone and the GSTDB image bank were used. The standard algorithm achieved success rates of approximately 91% for the recognition of signs in the test stage. For the recognition, despite the different characteristics accepted by the signs, the success rates reached about 89% for the recognition of the signs.

**Keywords:** HOG. SVM. HSV. Traffic Signs

## LISTA DE ILUSTRAÇÕES

Figura II.1 - Passos de reconhecimento .....	17
Figura II.2- Forma e Cores Padrão das Placas de Regulamentação. ....	18
Figura II.3 - Exceções das Placas de Sinalização de regulamentação. ....	18
Figura II.4 - Placa padrão de Sinais de Advertência. ....	19
Figura II.5- Exceções das Placas de Advertência. ....	19
Figura II.6 - Modelo RGB .....	20
Figura II.7 - Modelo HSL. ....	21
Figura II.8 - Modelo HSV .....	21
Figura III.1 - Etapas do Processo de Reconhecimento.....	27
Figura III.2 - Representação do Algoritmo HOG .....	30
Figura III.3 - Etapas do método HOG.....	30
Figura III.4 - Divisão por células. ....	32
Figura III.5 - Histograma das Células .....	33
Figura III.6 - Deslizamento da Janela por Blocos.....	34
Figura III.7 - Classes divididas pelo hiperplano .....	35
Figura IV.1 - Conjunto de imagens de treinamento .....	38
Figura IV.2 - Imagem com mais de uma placa de sinalização.....	38
Figura IV.3 - Representação da Imagem multi-channel e single-channel. ....	39
Figura IV.4 - Imagem com filtro GaussianBlur. ....	40



Figura IV.5 - Imagem convertida no espaço de cores HSV.....	41
Figura IV.6 - Imagem binária com máscara com os intervalos de cores definidos. ....	42
Figura IV.7 - Imagem com filtro de erosão.....	43
Figura IV.8 - Região de Interesse e escala de cinza. ....	44
Figura IV.9 - Descritor HOG. ....	45
Figura IV.10 - Modelo SVM.....	47
Figura IV.11 - Imagem classificada corretamente. ....	48
Figura IV.12 - Placa não reconhecida. ....	50

## **LISTA DE TABELAS**

Tabela IV.1 - Placas de Teste e Treinamento.....	37
Tabela IV.2 - Parâmetros C utilizados no treinamento do classificador .....	46
Tabela IV.3 - Resultado do Classificador .....	49

## **LISTA DE SIGLAS E ABREVIATURAS**

HOG	Histograma Orientado a Gradiente
ROI	Região de Interesse
HSV	Hue Saturation and Value (Matiz, Saturação e Valor)
RGB	Red, Green e Blue
SVM	Máquina de Vetores de Suporte
GTSDb	German Traffic Sign Detection Benchmark

# SUMÁRIO

CAPÍTULO I	INTRODUÇÃO .....	13
1.1	Problemática.....	14
1.2	Hipótese .....	14
1.3	Objetivos .....	14
1.4	Justificativas.....	15
1.5	Métodos.....	15
1.6	Organização .....	16
CAPÍTULO II	RECONHECIMENTO AUTOMÁTICO DE SINAIS DE TRANSITO ....	17
2.1	Placas com cores e formas padrões.....	17
2.2	Sinalização de Trânsito .....	18
2.3	Cores das Placas.....	20
2.3.1	Modelo RGB .....	20
2.3.2	Modelo HSL.....	20
2.3.3	Modelo HSV .....	21
2.4	Captura das Imagens .....	22
2.5	Segmentação .....	22
2.6	Detecção .....	23
2.7	Reconhecimento.....	25
CAPÍTULO III	SISTEMA DE RECONHECIMENTO PROPOSTO.....	27
3.1	Processo de Captura de Imagens.....	27
3.2	Pré-Processamento .....	28
3.3	Extração de Características .....	29
3.3.1	Calcular os Gradientes .....	31
3.3.2	Divisão por Células .....	31
3.3.3	Seleção e Acúmulo de Bins.....	32
3.3.4	Normalização .....	33
3.4	Detecção.....	34
3.5	Reconhecimento.....	34
CAPÍTULO IV	IMPLEMENTAÇÃO E RESULTADOS .....	37
4.1	Captura de Imagens.....	37
4.2	Pré-processamento .....	39
4.2.1	Filtro GaussianBlur .....	39

4.2.2	Seleção da Área de Interesse .....	40
4.2.3	Filtro de Erosão .....	42
4.3	Extração de Características .....	44
4.4	ESTRUTURA SVM .....	45
4.4.1	Treino .....	45
4.4.2	Teste e Classificação .....	47
4.5	Resultados .....	48
CAPÍTULO V CONCLUSÃO .....		51

## CAPÍTULO I INTRODUÇÃO

O reconhecimento automático de sinais de trânsito (TSR) tem um interesse considerável devido ao aumento de recursos computacionais para obter informações nas imagens. Têm crescido os estudos relacionados a técnicas para reconhecimento e identificação de objetos, formas e cores nas imagens. A tendência da linha de reconhecimento se deve a vários fatores como por exemplo, rapidez e dispositivos eficientes.

O desenvolvimento destes recursos para a utilização em computador está focado na segurança dos usuários, no qual devem emitir um alerta para a segurança. Muito dos motoristas estão mais preocupados com multas do que na própria segurança, sem embargo descuidam em muitos casos, por motivos diversos, focar a atenção nos sinais de trânsito. Os sinais de trânsito têm sido elaborados como formas e cores padronizadas para chamar a atenção aos usuários, sem pensar na tecnologia de reconhecimentos.

Tendo em vista que o reconhecimento de características importantes em imagens estudadas e capturadas pode ser uma tarefa difícil, devido ao local no qual a placa está instalada e os fatores, como por exemplo: chuva, sol, iluminação, poeira, assim como pedestres, árvores, entre outros objetos, interferem diretamente nas tonalidades e qualidade da imagem.

Além de que as imagens que não fornecerem uma vista nítida, pois podem ser obtidas em veículos que estejam em movimento ou até mesmo devido ao clima, assim podem interferir na obtenção das características necessária para fazer a detecção de objetos e sucessivamente o reconhecimento, trazendo resultados negativos. É importante que o algoritmo seja capaz de executar em tempo real e deve ter grande confiabilidade.

De qualquer forma, se faz necessário o alerta dos sinais de trânsito. No Brasil, pelo clima tropical, contaminação do ar dentro das cidades grandes, assim como no México, a manutenção dessas placas é complicadas e na etapa de detecção dificulta ainda mais. Em aplicativos baseado em geo-posicionamento (GPS), que utiliza um banco de dados de posições e informações relacionados aos sinais de trânsito, nem sempre são efetivos, pois as informações ficam desatualizadas devido às grandes mudanças que ocorrem devido à troca de sinalizações. As atualizações do GPS devem ser feitas para minimizar a busca das placas.

Em sistemas de reconhecimento automático de sinais de trânsito, uma câmera captura constantemente a imagem no percurso da viagem. Cada imagem capturada deve ser editada para que apenas a região de interesse que contém a placa, se houver, seja analisada algoritmicamente para o reconhecimento e a identificação de qual sinal de trânsito se trata.

O reconhecimento do sinal de trânsito é importante para sistemas de assistente de motorista. Um algoritmo bom é aquele que apresenta os melhores resultados, todo o processo de reconhecimento são feitas em três etapas: segmentação; detecção; reconhecimento (Shelke; Karde; 2012); (Ellahyani et al., 2016); (Aziz et al., 2018).

As placas de sinalização de trânsito têm tamanhos, formas e cores padrões, que facilita um pouco o trabalho na segmentação na qual pode-se utilizar as cores das regiões de interesse, na detecção a forma do sinal e sua representação, no reconhecimento pode ser utilizar diversos recursos, utilizando redes neurais por exemplo.

Os sistemas e técnicas de detecção e reconhecimento podem ser feitos de várias formas podendo verificar a que obteve o melhor resultado. Ainda não existe uma solução ótima para qualquer método de detecção e reconhecimento.

## **1.1 Problemática**

Persistência de diversos fatores humanos que inibe prestar a devida atenção na detecção e interpretação oportuna de sinais de trânsito.

## **1.2 Hipótese**

Um sistema automático de reconhecimento de placas de sinais de trânsito, baseado em visão computacional e Maching Learning, para alertar oportunamente os motoristas das regras de trânsito.

## **1.3 Objetivos**

Propor e implementar um sistema automático computacional baseado em visão computacional e máquina de aprendizagem (Máquina de Suporte de Vetores - SVM), para reconhecimento das placas de sinais de trânsito em tempo real para auxiliar o motorista.

**Objetivos Específicos:**

1. Detectar elementos de placas de trânsito.
2. Reconhecimento das informações que transmitem as placas.

**1.4 Justificativas**

Os acidentes de trânsito ocorrem com muita frequência por omissão dos sinais de trânsito nas regiões urbana e rural, em particular nas estradas com muito tráfico de veículos motorizados.

Para evitar esses acidentes, existem nas vias sinalizações como placas, avisos, radares entre outros, para que os condutores prestem a devida atenção e tomem as ações que indicam essas sinalizações.

Para muitos que trafegam com responsabilidade, podem ocorrer situações em que essas sinalizações não estão claras ou por razões diversas não conseguem reconhecer oportunamente.

Neste sentido um sistema de alerta da existência e significado das placas de sinais de trânsito pode ser de bastante ajuda para estes motoristas, assim evitar tragédias nas estradas por exemplo.

Com o aumento das tecnologias em automóveis faz-se necessário criar uma técnica para auxiliar os motoristas através de celulares, no qual devem identificar possíveis situações de risco, placas de trânsito e multas.

**1.5 Métodos**

A detecção é baseada no trabalho de Monografia de Stelet (2019) na qual utiliza a detecção de cores fazendo adequações, para fazer o reconhecimento dos sinais deve ser feito através do método de aprendizado para uma melhor eficiência ou resposta.

A imagem deve ser tratada, pois pode ser obtida de um carro em movimento e a qualidade pode não estar boa, o método deve ser confiável e eficaz, para obter os melhores resultados possíveis.



Se usa o método de HOG (Histograma de Gradiente Orientado do Inglês Histogram of Oriented Gradients) para caracterização do ROI. Para treinar adapta-se a SVM para reconhecimentos das informações das placas, com imagens de banco de imagens GSTDB de Houben et al. (2013). Os resultados são apresentados com imagens capturadas nas estradas do Brasil.

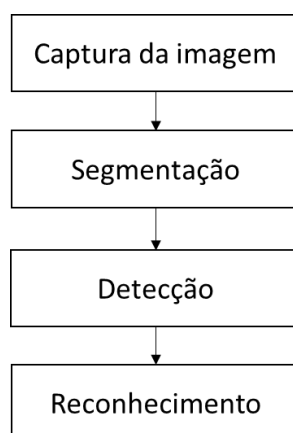
## **1.6 Organização**

Este trabalho está organizado da seguinte forma: Capítulo 1 é apresentado a introdução do que será abordado no trabalho, problema, hipótese, objetivos, justificativas e os métodos. Capítulo 2 apresenta a legislação de sinalização vigente, o sistema de segmentação e classificação de formas e os trabalhos relacionados; Capítulo 3 descreve o modelo proposto do sistema de reconhecimento das placas de sinalização, nele contém os detalhes sobre como foi estruturado e também sobre as tecnologias escolhidas para a implementação. Capítulo 4 contém o desenvolvimento e experimentação, dentre eles: captura de imagem, o pré-processamento, treinamento e teste do modelo proposto. Capítulo 5 apresenta a conclusão do trabalho e os trabalhos futuros.

## CAPÍTULO II RECONHECIMENTO AUTOMÁTICO DE SINAIS DE TRANSITO

Os sinais de trânsito no Brasil usam determinadas cores e formas geométricas como símbolos de alertas, algumas placas contendo mensagens diretas em texto, identificáveis ao olhar. Como as imagens são obtidas em tempo real, diversos fatores podem interferir na sua qualidade, com isso faz necessário que a imagem seja tratada. Existem quatro fases no sistema de detecção e reconhecimento: Captura da Imagem, Segmentação, Detecção e Reconhecimento. Na primeira fase no sistema de detecção e reconhecimento é a captura de imagem, onde deve ser tratada por conter objetos indesejáveis, no segundo é a segmentação, que podem ser feitas utilizando as cores, após esse passo é feito a detecção, que podem ser utilizados diversas técnicas e o reconhecimento, utilizando máquinas de aprendizagem. A Figura II.1 mostra a sequência típica de passos para ser feito o reconhecimento.

Figura II.1 - Passos de reconhecimento



### 2.1 Placas com cores e formas padrões

Um padrão é um modelo que deve ser seguido, no Brasil, como em outros países, existem modelos de placas de sinalização que utilizam formas e cores padrões. Existem diversas placas que são universais, ou seja, quando um motorista de outro país dirige e se depara com essas placas, ele consegue identificar, mas quando isso não ocorre, se faz necessário a presença de um alerta para a segurança do motorista. Para Shelke e Karde (2012), os sinais de trânsito são projetados para que pessoas possam reconhecer facilmente usando as cores e formas, em muitos países são usados os triângulos e círculos para transmitir avisos e proibições, no qual tem as bordas vermelhas para que possam ser vistas em longa distância.

Mas, não toda pessoa que dirige um carro consegue identificar os sinais, por diversos fatores. No trabalho elaborado por Hercik et al. (2013) é mostrado que os algoritmos propostos no trabalho são feitos principalmente para identificar sinais de trânsito na República Tcheca, mas podem ser facilmente estendidos para uso em todos os estados, pois eles utilizam formas padronizadas de sinais de trânsito.

## 2.2 Sinalização de Trânsito

No Brasil, o CONTRAN (Código Brasileiro de Trânsito) – define a forma padrão – disponibiliza as placas classificadas de acordo com as suas funções, em 3 tipos de sinalização vertical:

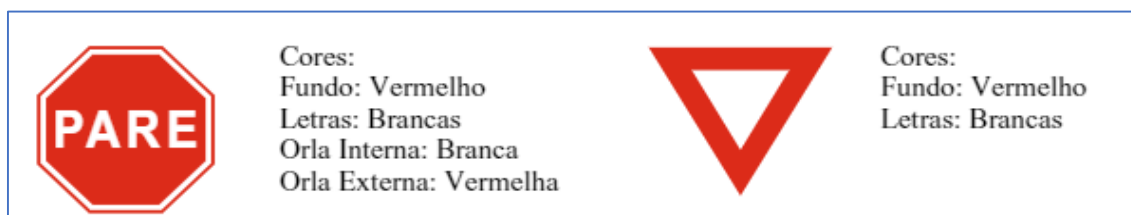
- **Sinalização de Regulamentação:** que orienta os motoristas das condições, proibições, obrigações ou restrições das vias. A forma padrão das placas é a circular como mostrado na Figura II.2. Existem apenas duas exceções da placa de regulamentação, a placa “PARE” representado por octógono e a placa “Dê a Preferência” representado por um triângulo invertido, como mostra a Figura II.2.

Figura II.2- Forma e Cores Padrão das Placas de Regulamentação.



Fonte: Docplayer

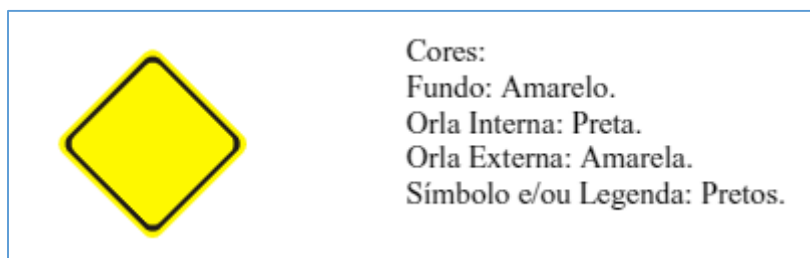
Figura II.3 - Exceções das Placas de Sinalização de regulamentação.



Fonte: Docplayer

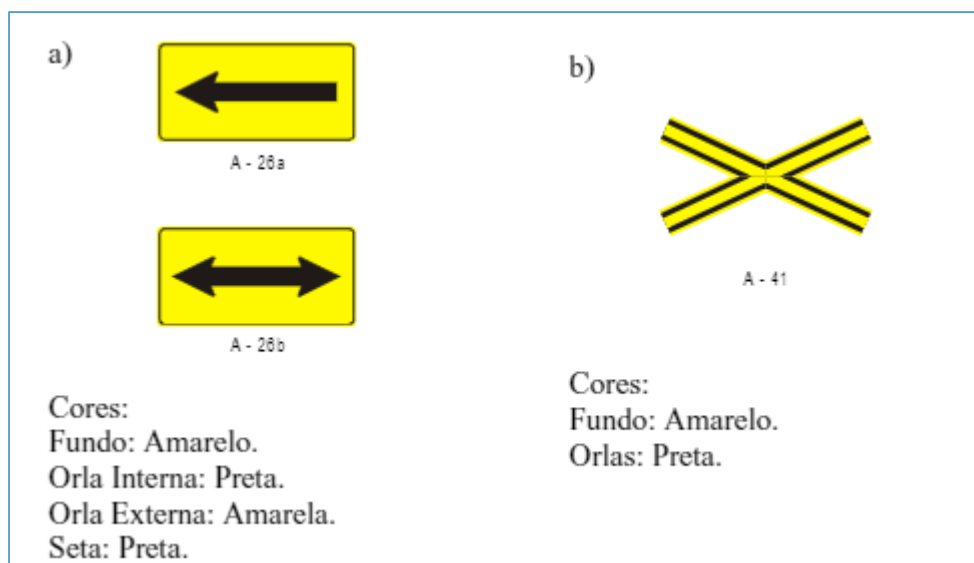
- Sinalização de Advertência: alertam os motoristas a situação das vias que podem ser perigosas. A forma padrão das Placas de Advertência é quadrada, podendo aparecer na diagonal como mostrado na Figura II.4. Existem três tipos de placas de advertência que tem como exceção sua forma, Figura II.5, "Sentido Único" - A-26a, "Sentido Duplo" - A-26b e "Cruz de Santo André" - A-41

Figura II.4 - Placa padrão de Sinais de Advertência.



Fonte: Docplayer

Figura II.5- Exceções das Placas de Advertência.



Fonte: Docplayer

- Sinalização de indicação: Identificam as vias e orientar os motoristas quanto sua localização e trajeto, mostrando a distância, serviços e percursos. As placas de identificação estão subdivididas em três grupos: Placas de Identificação e Localização de Destino, Placas de Orientação de Destino e Placas Educativas.

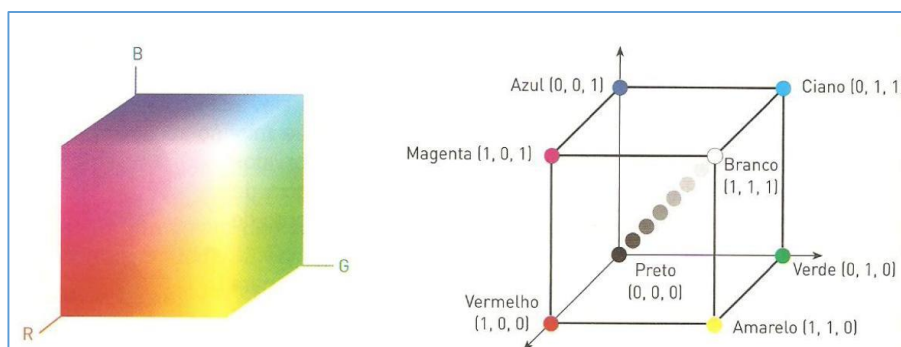
## 2.3 Cores das Placas

Na captura de placas de trânsito é gerada uma sequência de imagens no modelo luz RGB, de três canais conhecidos: vermelho, verde e azul. Em situações adversas de tempo e poluição, esse formato dificulta a identificação das cores das placas. Sendo, por esse motivo, necessário transformá-lo em outros modelos de luz, como HSL ou HSV. Esses modelos podem facilitar a identificação das cores afetadas pelos fatores externo do ambiente.

### 2.3.1 Modelo RGB

O RGB (vermelho, verde e azul) é um modelo que é baseado em um sistema de coordenadas cartesianas. Na Figura II.6 é apresentado um subespaço em um cubo de cores RGB. No modelo RGB a cor branca corresponde as três cores primárias (1, 1, 1) e a preta a falta dessas cores (0, 0, 0).

Figura II.6 - Modelo RGB



Fonte: Buulex

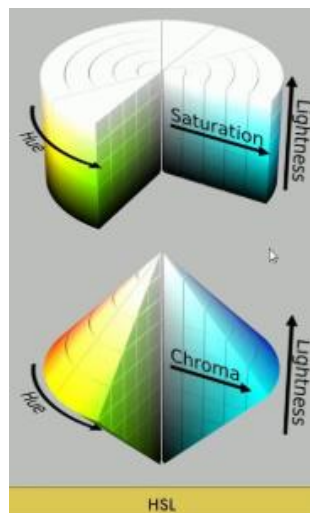
Os valores primários estão nos vértices vermelho, azul e verde, os secundários estão nos vértices ciano, magenta e amarelo, já o branco está no vértice mais longe da origem e o preto está na origem. Logo podem ser representado por um segmento de reta que unem os pontos dos vértices do preto até o branco.

### 2.3.2 Modelo HSL

O modelo HSL utiliza as características da tonalidade (Hue), Saturação (Saturation) e Brilho (Lightness). O Hue é a essência da tonalidade, que está no centro, a Saturation varia entre a cor real e a pouco intensa, dependendo do raio, a cor pode estar mais próximo da

matiz, a Lightness apresenta a variação que tem entre o preto e o branco, quanto mais próximo do topo, mais branco ficará e abaixo mais preto.

Figura II.7 - Modelo HSL.

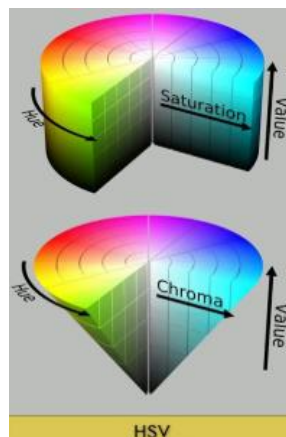


Fonte: Buulex

### 2.3.3 Modelo HSV

O modelo de cores HSV é composta pelas características de matiz, saturação e brilho, como pode ser visto na Figura II.8. O Hue (tonalidade) é composto por todas as cores, atinge os valores do 0 ao 360, na Saturation (saturação) se o valor for menor a tonalidade do cinza vai aparecendo e o Value (valor) os valores ficam de 0% a 100% e é definido o brilho da cor. Os valores do HSV estão relacionados com a luminância (componente que só contém as informações de brilho) da cor, que podem variar de 0 (preto) a 1 (branco).

Figura II.8 - Modelo HSV



Fonte: Buulex

## 2.4 Captura das Imagens

A captura de imagem para reconhecimento de sinal de trânsito deve ser realizada em tempo real por câmera de dispositivo móvel, para as suas respectivas operações de pré-processamento para facilitar a extração de características efetivas. Fatores como iluminação, vandalismo, clima, interferem diretamente, dificultam na fase de detecção, para fazer algum processamento nas imagens caso ela não esteja com uma qualidade boa. Grande parte deste procedimento utiliza as cores e texturas.

Segundo Silva et al. (2017), o pré-processamento transforma as imagens para escala de cinza e redimensiona. Também são utilizados os histogramas que ajudam na tonalidade e nos detalhes. No trabalho de Aziz, Mohamed e Youself (2018), como os sinais de trânsito variam em escalas muito diferentes, todas as imagens são redimensionadas. Informações de cores são descartadas e as imagens em escala de cinza resultante são normalizadas por contraste, as imagens são ajustadas aumentando o contraste, é feita a equalização do histograma para realçar o contraste, para que a saída seja uniforme.

## 2.5 Segmentação

Segundo Solomom e Breckon (2013) a Segmentação é processo de subdividir a imagem em regiões ou objetos que a compõem. A segmentação busca objetos ou regiões de interesses em uma imagem, ou seja, o que é importante na imagem. A segmentação de imagens é uma das tarefas mais difíceis e importantes no processamento de imagens.

O processo é feito em quatro etapas, através da captura da imagem, obtenção das características importantes, muitas das vezes é preciso fazer algum processamento na imagem para restaurar a imagem que pode não estar com uma qualidade boa. Com a segmentação torna-se possível obter informações importantes como cores e bordas.

Existem muitos trabalhos abordando a segmentação apresentados por (Shelke; Karde; 2012) e (Ellahyani; Ansari; Jaafari; 2016), cada um utiliza dados diferentes e diferentes métodos de reconhecimento que serão abordados em outra sessão.

O trabalho elaborado por Shelke e Karde (2012), faz o reconhecimento dos sinais de trânsito em 3 estágios: segmentação, Detecção e Reconhecimento. Na segmentação foi feita

utilizando o canal de cores RGB, onde foi extraído objetos do segundo plano (imagens de sinais de trânsito).

A proposta de Ellahyani et al. (2016) foi dividida em 3 fases: Segmentação, Detecção e Reconhecimento. O autor utilizou um banco de dados do GTSDb e STS. A segmentação foi realizada com base no canal de cor HSI. Na segunda fase detecta formas das placas que são formas triangulares, circulares, retangulares e octogonais. O artigo, propões utilizar metodos geométricos invariantes Hu com uma métrica simples para combinar as ROIs fornecidas pelo processo de segmentação com as formas, com isso dá melhores resultados num menor tempo de processamento em comparação com a máquina de algoritmos de aprendizagem.

## **2.6 Detecção**

A detecção refere-se a buscar os sinais de trânsito nas imagens correspondentes. Existem diversos métodos que podem fazer a detecção no reconhecimento das placas de sinalização, alguns deles foram comparados mostrando os que obtiveram os melhores resultados.

No trabalho apresentado por Silva et al. (2017), eles utilizaram o algoritmo SIFT. O algoritmo busca pontos de interesses na imagem e nas placas de sinalização. Depois de ter aplicado o algoritmo nas placas foi gerado um banco de dados com todos os descritores de pontos interesse, com isso minimiza o tempo de busca e aumenta o desempenho do algoritmo. Linhas retas são ligadas os pontos no template, logo após deve-se identificar os pontos de interesses que se ajustam com um modelo esperado comparando os ângulos de orientação, buscando uma região onde se concentra a maior parte dos pontos esperados e calculando a correlação entre as coordenadas espaciais dos pontos de interesse na imagem de entrada e no template, assim a maioria dos pontos chave do template estão relacionados com a imagem de entrada, ainda sim existem pontos que não estão relacionados com a imagem e que devem ser descartados verificando a região que tem mais pontos chaves na imagem de entrada e eliminando o restante.

Os autores Creusen, Wijnhoven, Herbschleb (2010) mostraram que o algoritmo HOG tem alto desempenho e possibilidades de implementação e treinamento rápido. Foi verificado que entre varios algoritmos o algoritmo HOG explora implicitamente recursos como padrões de gradiente e a forma dos sinais de trânsito. A escolha do espaço de cores influencia



significativamente o desempenho, e a escolha ideal depende do tipo de sinal de trânsito. O autor comparou um algoritmo específico de última geração e o algoritmo baseado em HOG.

A proposta dos autores Silva et al. (2017) foi demonstrar um sistema que seja capaz de obter uma grande porcentagem de acertos com custos computacionais mais baixos, com isso eles utilizaram um algoritmo de otimização que define máscaras para extração de características e logo em seguida a classificação de sinais de trânsito. As máscaras são definidas utilizando o ABC, que é um algoritmo que inspira o comportamento das colônias de abelha e apresenta boa capacidade de busca e amplitude, assim como boa capacidade de busca em profundidade.

A proposta de Aziz et al. (2018) foi criar sistema deve ser capaz de lidar com a visibilidade dos sinais de trânsito em uma ampla gama de condições climáticas e iluminar ambientes com variações de visibilidade, como neblina, chuva, nuvens e sol. Utilizaram os recursos extraídos histograma de gradientes orientados (HOG), o recurso Gabor e o recurso Compound local binary pattern (CLBP).

Alguns dos métodos mais utilizados nos artigos que foram estudados:

- SIFT: busca pontos de interesses na imagem e nos templates, são geradas informações para cada ponto de interesse no qual são inseridas em um vetor de 128 posições no qual descrevem cada ponto da imagem; a localização x, y; a escala s; a orientação. O algoritmo é dividido em quatro etapas: é feita a procura dos pontos de interesse invariantes à escala e rotação utilizando a função de Diferença de Gaussianas (DoG), para cada ponto extremo encontrado é estabelecido a localização e a escala, a orientação de cada ponto de interesse é feita através dos gradientes locais da imagem, com isso é gerado um histograma com as informações da imagem como à orientação, escala e a localização dos pontos de interesses, após o histograma gerado, deve-se calcular um descritor para a região da imagem local, essas medidas são mostradas para ver o nível de distorção e mudanças de iluminação. É feito um histograma para cada região com a magnitude dos pixels pertencentes a cada uma das regiões, assim o descritor é mostrado.
- HOG: (Aziz, Mohamed e Youssself, 2018) a proposta é dividir a imagem em blocos sobrepostos, esses blocos são divididos em células. Cada pixel é calculado com base na magnitude e ângulo do gradiente. Um histograma é formado com as orientações dos

gradientes de cada célula. A formação gerada pela concatenação dos histogramas de toda as células no bloco correspondente é gerado pelo vetor de característica de cada bloco. O recurso HOG é formado pela concatenação de todos os vetores de recursos do bloco.

- Gabor: (Aziz, Mohamed e Youssself, 2018) é uma técnica de extração de recursos com ótimos resultados usada no processamento de imagens. Na utilização do filtro Gabor é em uma imagem, os pixels que têm a mesma frequência e orientação locais darão a resposta mais forte. A equação do filtro de Gabor é representada como uma função gaussiana. É utilizado um banco de filtros Gabor para obter um vetor. Os filtros Gabor que tem multi-orientações e uma ou mais escalas são projetados, a extração de recursos do Gabor é realizada pela convolução desses filtros com a imagem de entrada.

## 2.7 Reconhecimento

Na etapa do reconhecimento pode-se utilizar diversos tipos de classificadores, bem como o SVM e Random Forest, ambos os classificadores anteriores devem ser treinados e testados. Para fazer o treinamento utiliza-se imagens contendo placas de sinais de trânsito e imagens que não tenha placas de trânsito.

No trabalho apresentado por Ellahyani et al. (2016) foram utilizados diferentes classificadores, o Random Forest e SVM foram usados para usar recursos diferentes. Assim que o classificador é selecionado, diferentes recursos considerados como Histograma de Gradiente Orientado (HOG), Padrão Binário Local (LBP) e Auto-Similaridade Local (LSS) são levados em consideração. Todos os recursos foram experimentados de forma independente com o Random Forest. São fornecidas visões gerais do classificador Random Forest, SVM classificador, e os recursos usados para testar esses classificadores. O método seguido para decidir qual recurso a ser usado é detalhado e o sistema de reconhecimento é deduzido dos resultados de comparação.

Na proposta de Creusen et al. (2010) foi utilizado o classificador de SVM, ele é treinado por um processo iterativo. Na primeira iteração, todas as imagens que foram consideradas positivas são processadas e caso o resultado dê negativo, um conjunto de regiões de fundo selecionadas aleatoriamente é usado. Em cada iteração adicional, o detector atual é

aplicado a uma nova imagem sem sinais de tráfego e as falsas detecções resultantes são adicionadas ao conjunto de treinamento para a próxima iteração. Após cada iteração, o classificador é treinado novamente e todas as amostras de treinamento negativo são descartadas. A consequência dessa técnica é que o conjunto de características negativas permanece pequeno.

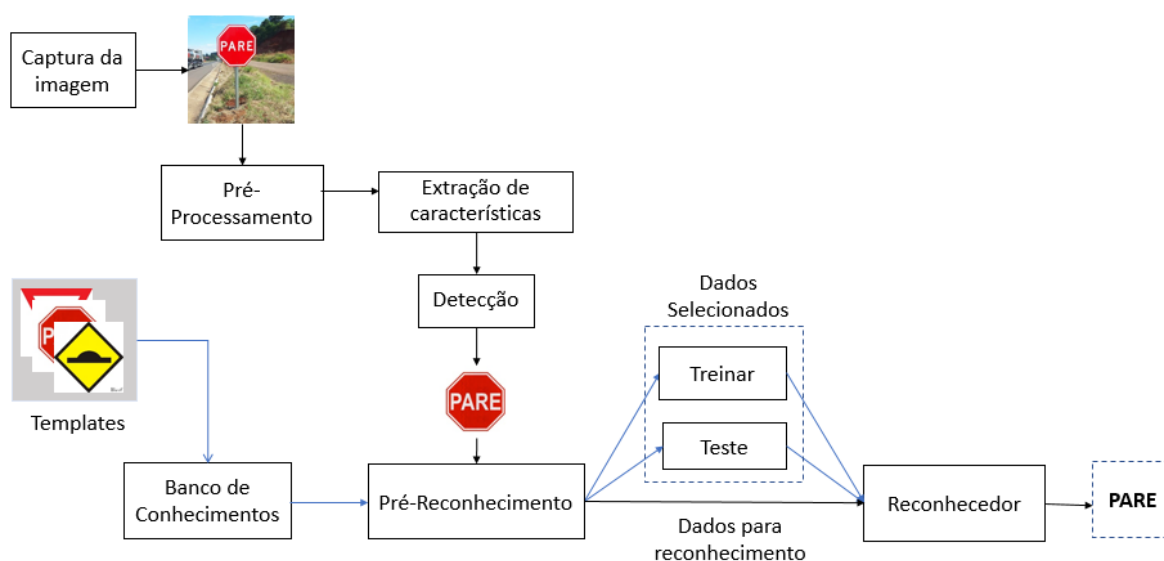
No trabalho de Silva et al. (2012), após serem encontradas as máscaras com o algoritmo ABC e realizada a convolução com as imagens, a classificação é feita utilizando o algoritmo Random Forest, no qual tem alto rendimento para implementações em tempo real e com objetivo de reconhecimento de sinais de trânsito, a precisão e o tempo de classificação foram comparados para as técnicas SVM, MLP e Random Forest, resultando esta última como sendo a mais rápida e precisa.

Na proposta de Aziz et al. (2018) foi comparado o desempenho do classificador ELM, com as técnicas SVM, k-Nearest Neighbors (k-NN). Foi comparado o tempo de otimização de parâmetros, o tempo de treinamento, o tempo de teste e a taxa de reconhecimento. Foi utilizado os parâmetros do kernel usando o método da busca de grade. Com os resultados obtidos foi revelado que o classificador ELM supera o restante dos dois algoritmos, ou seja, SVM e KNN. O desempenho foi melhor em termos de tempo de otimização de parâmetros e tempo de teste para os dois conjuntos de dados. É devido ao menor número de parâmetros de ajuste do algoritmo de aprendizado do ELM. Além disso, foi mostrado que o ELM é um classificador básico, em vez de usar classificadores SVM ou KNN. O estudo comparativo mostrou que a capacidade de generalização do classificador SVM é maior comparada com a do classificador KNN na classificação dos sinais de tráfego das bases de dados GTSRB e BTSC.

## CAPÍTULO III SISTEMA DE RECONHECIMENTO PROPOSTO

A proposta para o sistema de reconhecimento de placas de sinais de trânsito parte de dois tipos de entrada templates e imagens instantâneas, que são pré-processadas, extraídas seus atributos, detectadas a formas das placas diferenciando do resto da imagem instantânea, consideradas outros elementos diferenciadas registrados nos templates estáticos (conhecimentos reais de adversidades), no pré-reconhecimentos. Reconhecimento da placa é realizado por uma máquina previamente treinada para identificar qual tipo de alerta representa a placa. A Figura III.1 ilustra a organização em módulos o processo de reconhecimento treino. Portanto, o sistema de reconhecimento pode ser dividido nas etapas: captura da imagem, Pré-Processamento, Extração das Características, Detecção, pré-reconhecimento e Classificação.

Figura III.1 - Etapas do Processo de Reconhecimento



### 3.1 Processo de Captura de Imagens

As imagens são necessárias para três categorias de operações neste sistema, elas são: imagens para treino, imagens para teste, e imagens para reconhecimento. As imagens para treino e testes consistem em um conjunto de variedades de imagens em situações diversas, que serão úteis para gerar o conhecimento das máquinas classificadoras. As imagens para esse

propósito são difíceis de prever o como e quantidades serão usadas de cada situações. Por isso, trabalhos de pesquisas recomendam usar imagens de bancos de dados definidas por trabalhos de pesquisas selecionadas com critérios apropriados. Por essa razão, neste trabalho são utilizadas para treinos as imagens de banco de dados de pesquisas “GTSDb” de Houben et al. (2013).

Para capturar a imagem será utilizado um celular que possa obter uma imagem com uma boa resolução. As imagens serão obtidas por vídeo que será gravado em vários momentos, no qual possa ser obtido várias imagens em posições diferentes. É importante uma câmera de boa qualidade para que o resultado esperado seja o melhor possível, evitando que dê resultados errados para evitar incidentes. Após ter feito a captura da imagem é necessário que seja feito o processamento nas imagens para melhorar a qualidade da imagem.

### **3.2 Pré-Processamento**

Devido aos vários fatores, como iluminação do ambiente, velocidade que o carro trafega e objetos no qual atrapalhe a visualização e que podem interferir na qualidade da imagem e deixar menos legível quando for capturada, é importante que seja feita o pré-processamento na imagem adquirida.

Pode se utilizar métodos que possam melhorar a qualidade da imagem, deixando a imagem mais nítida, tirando ruídos e ajustando para que o resultado seja o melhor. A utilização desses métodos tem como objetivo de melhorar a imagem para aprimorar a visualização, podem ser melhoradas as características de linhas, cores, bordas e texturas. Existem diversos filtros que podem ser utilizados nas imagens, como por exemplo gradiente, laplaciano, entre outros.

O filtro de gradiente pode ser utilizado em uma imagem distorcida, após ter sido obtida pelo movimento do veículo. Isso ocorre em ambientes de segmentação de placas de sinalização, no qual são imagens são obtidas em um veículo em movimento, dificultando a identificação, devido a características como bordas e cores não estão em situações favoráveis. Com aplicação do filtro, é obtido uma imagem com melhor qualidade e detalhamento, facilitando o processo de identificação.

Para a identificação de cores em placas de sinalização a transformação da imagem para o modelo de cores RGB para o modelo HSV. A imagem obtida, pode sofrer variação na iluminação, dificultando a visualização. O espaço HSV é muito utilizado no realce de regiões baseado na cor, que facilite a detecção do objeto de interesse.

Em diversas aplicações é necessário selecionar o ROI, separando essas regiões das demais. Segundo Dougherty (2009), a limiarização é um dos métodos mais práticos para separar o ROI de acordo com os diferentes níveis de intensidade ou cor de cada pixel da imagem utilizada. Pode ser calculado o limiar para definir quais cores específicas devem ser destacadas nas imagens. As imagens podem ser utilizadas em escala de cinza ou colorida e a saída é uma imagem binária, contendo valores de 0 para o fundo da imagem e 1 para os objetos de interesses na imagem. Também pode ser calculado um limiar para definir quais cores específicas, deve ser destacada na imagem.

### **3.3 Extração de Características**

Uma imagem pré-processada deve ser feita a extração de características, para a obtenção de informações importantes no qual pode ser através de cores e da borda. Devem ser atribuídas, também, nesses atributos as informações relacionadas às situações do status das placas, como as informações relacionadas às inclemências climáticas, periódicas, tempo e contexto do ambiente.

Uma das etapas fundamentais para caracterização é que a imagem seja isolada unicamente o setor de ROI, que reduz drasticamente o tamanho de análise. Essa imagem convertida em um único canal de cinza, seguida por uma padronização de tamanho, para depois subdividir em áreas menores para facilitar o cálculo dos atributos, por histogramas de gradientes orientados. Trata-se de atribuir um vetor numérico que descreva em forma global à imagem.

Na Figura III.2 é representado o descritor Histograma de Gradientes Orientados (HOG) proposto por Dalal e Triggs (2005), é utilizado no processamento de imagens para o reconhecimento de padrões. O objetivo do descritor é extrair dados de uma imagem de acordo com a orientação das arestas, sendo estas arestas calculadas através do método de detecção de borda.

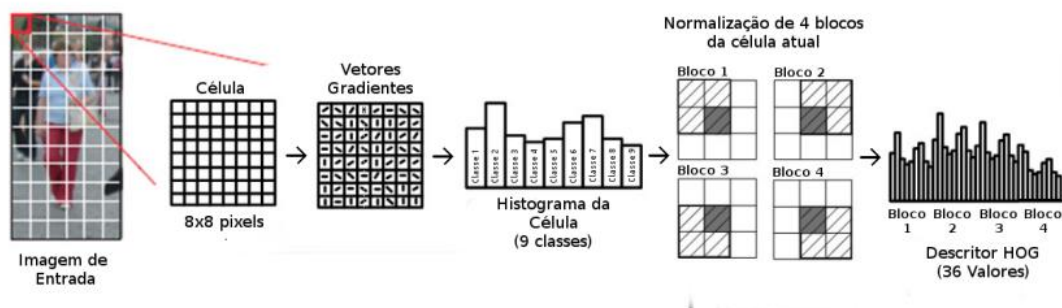
Figura III.2 - Representação do Algoritmo HOG



Os atributos da imagem obtidos de forma que, mesmo sob diferentes aspectos – como a luminosidade e qualidade, por exemplo– o descritor continua o mesmo ou quase iguais (McCormick, 2013), isso pode ser conseguido eficientemente por HOG. Para isso, se agrupam as partições da imagem em blocos sobrepostos das imagens. As partições são conhecidas como células, que agrupados formam blocos. O histograma de orientações dos gradientes é formado para cada célula e calculado na horizontal e vertical. Os pixels são calculados pela magnitude do gradiente e pelo ângulo (Aziz et al. 2018).

A Figura III.3 representa as etapas do algoritmo HOG. A imagem de entrada foi redimensionada para uma proporção fixa (largura: altura), sendo esta imagem particionada em blocos e os blocos dividido em 8:16 células e cada célula de tamanho 8x8 pixels, formando os vetores gradientes e os histogramas da célula (x orientação). Depois que o histograma foi gerado, é realizado a normalização dos blocos através do deslizamento da janela determinada pelo bloco para a imagem. Assim que todas as células dentro do bloco forem normalizadas e todos os blocos serem percorridos, os histogramas apresentam as informações do frame de entrada.

Figura III.3 - Etapas do método HOG.



Fonte: (Suleiman; SZE; 2014)

### 3.3.1 Calcular os Gradientes

Segundo Lira (2021, p.39, 40 e 41), para realizar o processamento corretamente após calcular seus gradientes, precisa de duas informações básicas: o módulo e a orientação dos gradientes do frame. É importante aplicar um filtro à imagem em escala de cinza para obter melhores resultados, destacando as áreas da imagem com maior cor e realçando as bordas. Dois filtros são usados: para o eixo cartesiano x, gerando um frame filtrado no eixo horizontal e para o eixo y, gerando um frame filtrado no eixo vertical.

O frame que resulta  $g_x$  possui o módulo do gradiente  $g_x$  para a relação de cada pixel com o eixo x, enquanto o frame que resulta em  $g_y$  é composto pelos gradientes  $g_y$  para a relação de cada pixel com o eixo y.

			+1
-1	0	+1	0
			-1

O módulo final de cada gradiente é calculado na imagem através do cálculo pixel a pixel a partir da raiz quadrada da soma dos quadrados dos gradientes através da expressão (1) abaixo, sendo  $g_x$  e  $g_y$  valores de gradiente do mesmo pixel em seus respectivos eixos após a filtragem. Enquanto a orientação dos gradientes é obtida a partir do arco tangente de  $g_y$  a  $g_x$  (2).

$$g = \sqrt{g_x^2 + g_y^2} \quad (1)$$

$$\theta = \arctan \frac{g_y}{g_x} \quad (2)$$

### 3.3.2 Divisão por Células

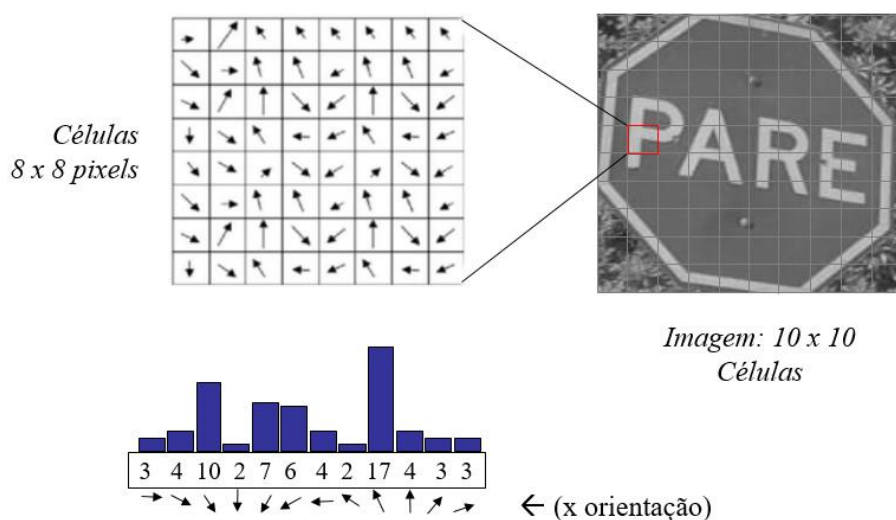
O algoritmo divide a imagem de entrada em subgrupos chamados células. Cada célula é representada por um vetor de informações derivadas da técnica de Histograma de Gradientes Orientados (HOG). Cada vetor representa uma distribuição de dados na forma de um histograma que corresponde a uma parte da imagem de entrada através da divisão por



células. Essa divisão permite a detecção de informações dentro de uma imagem, dando origem a descritores de informações baseados na localização da análise.

Na Figura III.4 a imagem é dividida em células (10x10) e cada célula tem tamanho de 8x8 pixels, gerando um histograma de orientações (bins) de gradiente para cada célula, de acordo com o intervalo  $\theta = 20$  graus: 0-19, 20-39, 40-59, ..., 160-179. O valor associado a um bin do histograma é denominado “voto” e provém das magnitudes dos gradientes dos pixels, onde as orientações contribuem para o valor do bin.

Figura III.4 - Divisão por células.



As dimensões de cada célula, são predeterminados como parâmetros do algoritmo em que a célula possui dimensões quadradas, ou seja, o comprimento e a altura da célula devem ser múltiplos do comprimento e da altura da imagem. À medida que o frame de entrada é varrido, os dados do gradiente que representam a imagem, são processados e acumulados em histogramas.

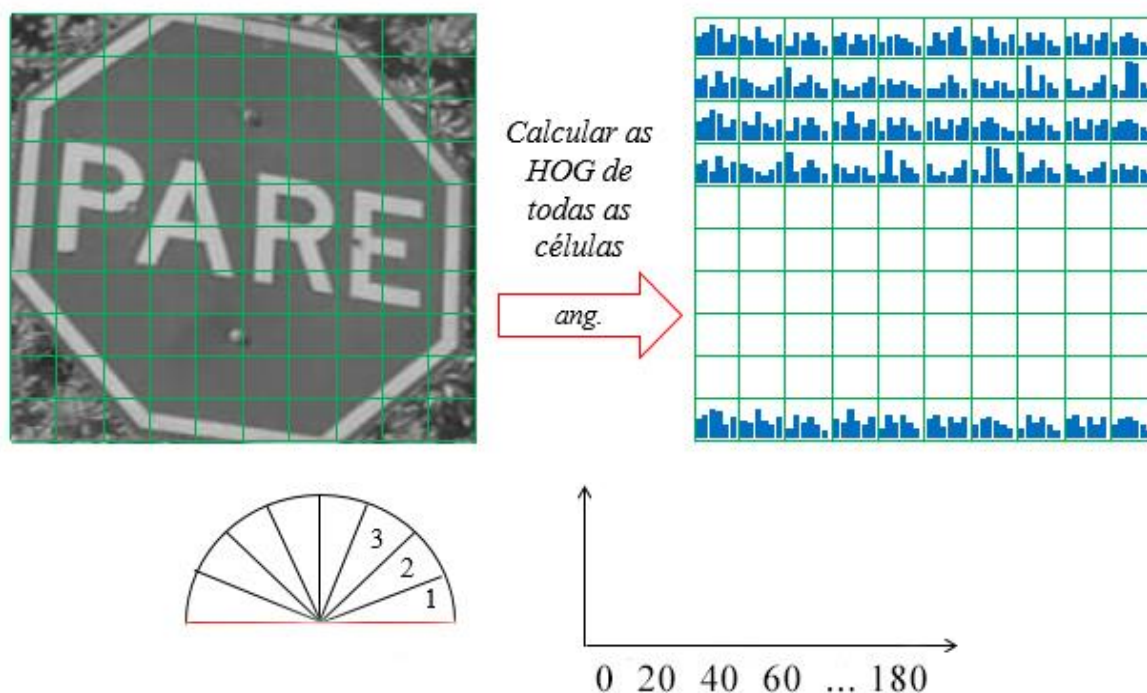
### 3.3.3 Seleção e Acúmulo de Bins

Primeiro, é definido uma constante que depende do número de bins do histograma a serem calculados, ou seja, do número de direções a serem representadas no histograma, particionando o vetor em intervalos de ângulos iguais.

Dado que a direção da orientação é importante para a separação dos gradientes de entrada em cada um dos bins correspondentes e a distância entre os bins é calculada usando o ângulo.

Os limites de cada bin são calculados quando a quantidade de cada bin para o histograma final de características for definido. Os valores continuam constantes para determinar a posição do vetor, de modo que cada um dos índices do vetor tenha um alinhamento contínuo com a posição. Os valores de direção são calculados para todas as células, fazendo com que as posições do histograma variem entre  $0^\circ$  a  $180^\circ$ , como mostra a Figura III.5

Figura III.5 - Histograma das Células

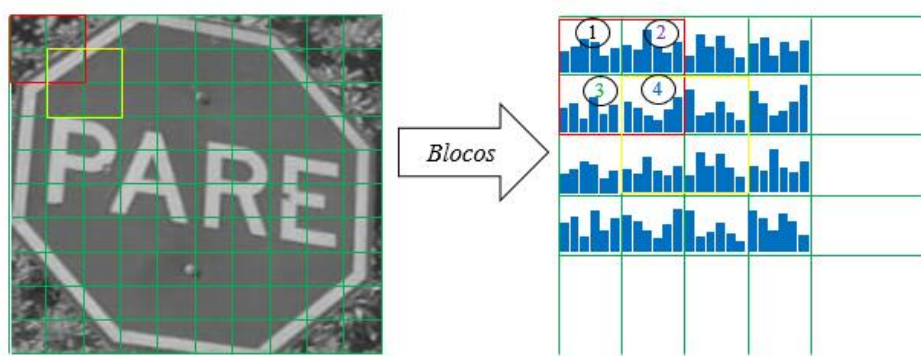


### 3.3.4 Normalização

A normalização é importante pois as placas de sinais de trânsito sofrem variações de iluminação, onde ocorre mudanças nos gradientes.

As células são agrupadas em blocos, com dimensões pré-determinadas de acordo com os parâmetros de tamanho das células. Esses blocos determinam uma nova subdivisão do frame que inclui grupos de células a partir das imagens e realiza uma normalização dos descritores dessas células através do deslizamento da janela determinada pelo bloco para a imagem, representado na Figura III.6.

Figura III.6 - Deslizamento da Janela por Blocos.



A imagem é percorrida até que todas as células dentro do bloco sejam normalizadas. Após todos os blocos serem percorridos, os histogramas normalizados apresentam com precisão as informações do frame de entrada.

### 3.4 Detecção

O processo de detecção da placa de trânsito pode se dar através da sua forma: quadrada, circular, octogonal e triangular. O processo de detecção das placas pode ser feito utilizando alguns métodos, um deles é aplicando o momento invariante de HU que utiliza a posição, escala e orientação descrevendo imagem estudada e são comparadas com os templates armazenados e processados, e o outro método utilizado depois de obter o resultado do método anterior é a transformada de Hough, que evita a detecção de placas “reconhecidas” de maneira errônea. Depois da utilização dos métodos, podem ser possíveis obter resultado positivos da detecção da forma da placa, através da imagem estudada.

O descritor HOG também pode ser utilizado na detecção, ele é semelhante a outros métodos, mas tem algumas vantagens, pois ele também é invariante para transformações geométricas e trabalha em células localizadas.

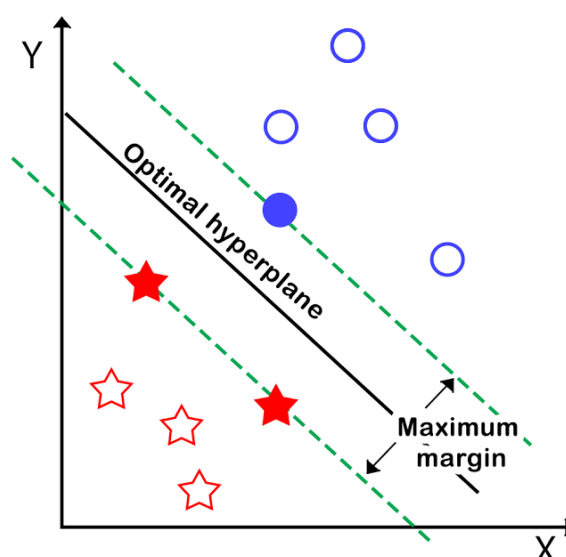
### 3.5 Reconhecimento

O reconhecimento de cada placa identifica o significado de cada placa em função da figura e que estão inseridas em pictogramas. Existem diversos métodos que podem ser utilizados como o Random Florest, SVM (Support Vector Machine), entre outros métodos. Esses métodos precisam ser treinados e testados e deve ser avaliado o que se adapta melhor ao modelo proposto de acordo com as características obtidas de cada imagem.

O SVM é um algoritmo supervisionado que analisa dados e classifica padrões, ou seja, para classificação e análise de regressão. Segundo Campbell e Ying (2011) as SVM se popularizaram nas últimas décadas por possuírem uma eficácia superior, se comparadas a outros algoritmos de aprendizagem.

De acordo com Cortes e Vapnik (1995), os SVMs buscam as informações na entrada do conjunto de dados no espaço dimensional, que são apresentados como pontos. É traçado um hiperplano que separa os pontos através de margens associadas (linhas paralelas) que determinam a qual lado ou classe cada ponto pertence, esse processo é chamado de separação ou classificação linear, como podemos observar na Figura III.7.

Figura III.7 - Classes divididas pelo hiperplano



Fonte: Medium.

O objetivo do SVM é detectar a melhor fronteira de separação entre classes possíveis, para um dado conjunto de dados que sejam linearmente separáveis. As fronteiras que conseguem separar completamente as classes são conhecidas de hiperplanos.

As de Máquinas de Vetores de Suporte são robustas diante de dados de grande dimensão, como fornecidos pelos descritores. Diante do olhar computacional, é o processo de classificação veloz, (Bragatto et al; 2016).

O método SVM que permite erros de classificação é chamado de SVM de Margem Suave (*Soft Margin SVM*). Esse método introduz a variável de folga ( $\zeta$ ):

$$\begin{aligned} \min \quad & ||w|| + C \sum \xi_i \\ \text{sujeito à } & y_i(W^T X_i + b) \geq 1 - \xi_i, \text{ para todo } 1 \leq i \leq n \end{aligned}$$

- Se  $\xi = 0$ , o ponto está na margem ou depois;
- Se  $0 < \xi \leq 1$ , o ponto está entre as margens e o lado correto;
- Se  $\xi \geq 1$ , o ponto está do lado errado.

O método também possui a constante de custo ( $C$ ), que representa o custo de cada erro de classificação e deve ajustado de acordo com o conjunto de dados. Se o valor de  $C$  for baixo, as restrições são ignoradas (margem maior), o valor de  $C$  alto, as restrições são mais difíceis de serem ignoradas (margem menor) e se  $C = \infty$ , nenhuma restrição é ignorada (margem rígida).

É importante ressaltar que ajustar corretamente o  $C$ , pois aumenta ou diminui o desempenho do classificador referente ao conjunto de dados de conhecimento. É adequado achar um valor para  $C$  que considere a base de exemplos, para maximizar os acertos do classificador.







## CAPÍTULO IV IMPLEMENTAÇÃO E RESULTADOS

A partir do que foi apresentado nos capítulos anteriores, é possível perceber a diversidade de métodos que podem ser aplicados em sistemas de reconhecimento de placas de trânsito. Neste capítulo serão apresentados a implementação dos métodos e a utilização da classificação de algumas classes de placas de sinais trânsito brasileiras. O algoritmo proposto é baseado nas bibliotecas *OpenCV* e *Dlib* e na linguagem de programação *Python*.

### 4.1 Captura de Imagens

As imagens foram divididas em três categorias: teste, treino e reconhecimento. Como não existe um banco de imagens com placas brasileiras, foram capturadas algumas imagens e juntamente, foi utilizado o banco de imagens “GTSDb” de Houben et al (2013). As imagens foram divididas em pastas e foram definidas algumas classes de imagens. As informações das classes são apresentadas na Tabela IV.1 com suas devidas informações:

Tabela IV.1 - Placas de Teste e Treinamento

Classes	Treino	Teste	Reconhecimento	Total
	14	11	10	35
	20	12	16	48
	16	12	6	34
	20	26	7	53
	13	6	4	23
	25	19	22	66
<b>Total</b>	<b>108</b>	<b>86</b>	<b>65</b>	<b>259</b>

Como as placas de sinais de trânsito possuem variação, para se treinar um classificador, as imagens foram redimensionadas. As informações do banco de dados não possuem uma razão de aspecto constante, pois foram feitas manualmente através da ferramenta *Imglab* (ferramenta gráfica para rotular imagens em caixas delimitadoras).



A Figura IV.1 mostra alguns modelos do conjunto de treinamento e modelos de variações de diferentes ambientes, no qual pode-se notar as diferenças em vários aspectos, um deles é a tonalidade das imagens.

Figura IV.1 - Conjunto de imagens de treinamento



Uma imagem com mais de uma placa de sinalização, faz com que o treinamento crie um classificador mais robusto, capaz de detectar placas em meio a vários ambientes e com diferentes posições. Como pode ser observado na Figura IV.2.

Figura IV.2 - Imagem com mais de uma placa de sinalização.



Fonte: Banco de Imagens GSTDB

Para armazenar as imagens foi utilizado o método *Imread* que carrega a imagem de um arquivo, sendo armazenado uma matriz do tipo *Mat*, um tipo próprio da biblioteca *OpenCv*, de acordo com Bradski (2000).

O tipo *Mat*, é representado por uma matriz n-dimensional que pode ser de dois tipos: *single-channel*, imagem em escala de cinza ou *multi-channel*, imagens coloridas representadas conforme a imagem. Na Figura IV.3, lado esquerdo podemos observar a representação do *multi-channel* e lado direito *single-channel*. Cada uma das matrizes representam um canal do sistema de cores escolhidos, normalmente sendo este o RGB, no entanto o padrão de exibição no *OpenCV* compõe as cores usando o espaço de cores BGR (trocando o vermelho e azul de posição).

Figura IV.3 - Representação da Imagem multi-channel e single-channel.

200	123	245	121	0		
11	200	123	245	121	0	
22	111	255	56	207	121	36
63	228	23	200	24	100	123
89	63	19	34	185	207	123
	89	0	78	51	245	66
		72	74	136	0	128

54	255	28	210	8
45	200	24	100	123
85	34	185	207	123
10	78	51	245	66
99	74	136	0	128

## 4.2 Pré-processamento

Para obter uma imagem completamente nítida é uma tarefa difícil, que depende de vários fatores, como a luminosidade, chuva e principalmente da velocidade em que se está trafegando. Sendo assim é importante utilizar de técnicas que possam melhorar as imagem afim de ter um resultado mais preciso.

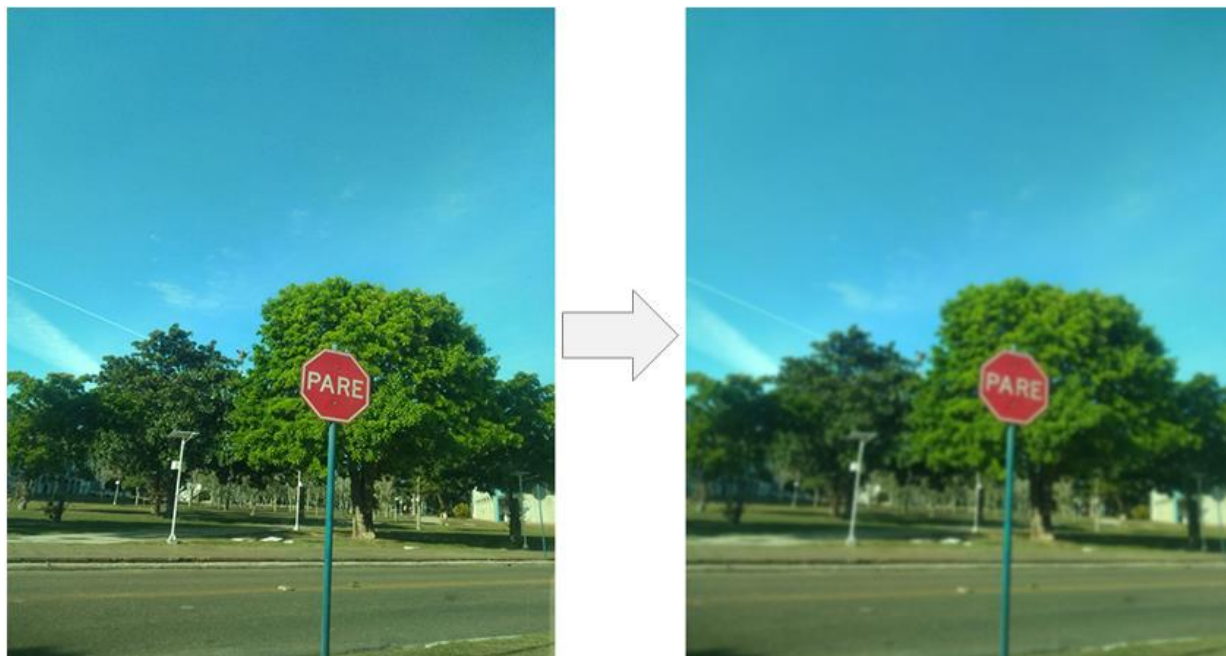
### 4.2.1 Filtro GaussianBlur

As imagens obtidas podem ter diversos ruídos, principalmente por causa do sensor da câmera e do ambiente no qual as placas estão posicionadas. Para melhorar esses ruídos, o Filtro GaussianBlur é uma técnica de suavização da imagem que ajuda a reduzir os ruídos. Nas imagens, quaisquer bordas nítidas nas imagens são suavizadas, isso faz com que haja a



redução do desfoque e melhora a qualidade da imagem, e para isso foi utilizada a função: *cv2.GaussianBlur()*, como podemos ver na Figura IV.4 após a aplicação do filtro.

Figura IV.4 - Imagem com filtro GaussianBlur.



#### 4.2.2 Seleção da Área de Interesse

As classes de imagens são adicionadas nas pastas do projeto e o passo inicial é fazer com que o programa passe por todas as imagens redefinindo o tamanho das mesmas para uma proporção fixa, com isso faz que o processamento seja mais rápido e eficiente.

Para a identificação das cores em placas de sinalização é importante fazer a transformação das imagens para o modelo de cores RGB para o modelo HSV. Para fazer a conversão, utiliza-se o método *cvtColor()* da biblioteca *OpenCV*, como apresenta a Figura IV.5, no qual as cores de interesse são realçadas e ficam predominantes.

Figura IV.5 - Imagem convertida no espaço de cores HSV.



De acordo com Bradski (2000) as equações usadas para a conversão para HSV são:

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

$$S = \begin{cases} \left( V - \frac{\min(R, G, B)}{V} \right) & \text{if } V \neq 0 \\ 0 & \end{cases}$$

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)} & \text{if } V = R \\ \frac{120 + 60(B - R)}{(V - \min(R, G, B))} & \text{if } V = G \\ \frac{240 + 60(R - G)}{V - \min(R, G, B)} & \text{if } V = B \end{cases}$$

Com a Região de Interesse e os intervalos de cores definidos para cada modelo de placa, as imagens passam pelos métodos `cv2.threshold()` e `cv2.bitwise_and()` que geram imagens binárias e criam uma máscara com os intervalos de cores correspondentes ao que é

compatível com o intervalo estabelecido e o preto ao que não é compatível, como podemos visualizar na Figura IV.6 e descrito no Código IV.1.

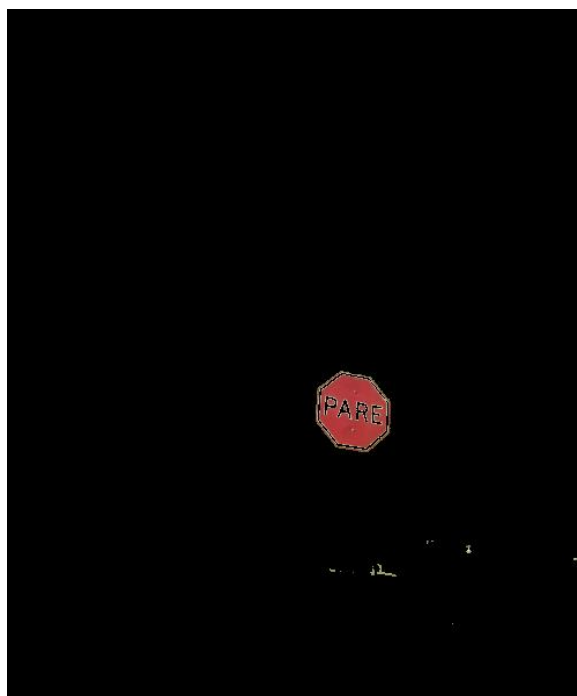
Código IV.1 – Definição do intervalo de cores

```

1  #converter espaço de cores RGB para HSV
2  hsv = cv2.cvtColor(imagem, cv2.COLOR_BGR2HSV)
3  #Intervalo de cores Amarelo
4  lower = np.array([0, 25, 50])
5  upper = np.array([30, 255, 255])
6  #Intervalo de cores Amarelo
7  lower2 = np.array([145, 25, 50])
8  upper2 = np.array([180, 255, 255])
9  #Intervalo de cores Vermelho
10 lower3 = np.array([0, 59, 49])
11 upper3 = np.array([0, 100, 100])
12
13 mask1 = cv2.inRange(hsv, lower, upper)
14
15 mask2 = cv2.inRange(hsv, lower2, upper2)
16
17 mask3 = cv2.inRange(hsv, lower3, upper3)
18
19 mask = mask1 + mask2 + mask3
20 target = cv2.bitwise_and(imagem, imagem, mask=mask)

```

Figura IV.6 - Imagem binária com máscara com os intervalos de cores definidos.



### 4.2.3 Filtro de Erosão

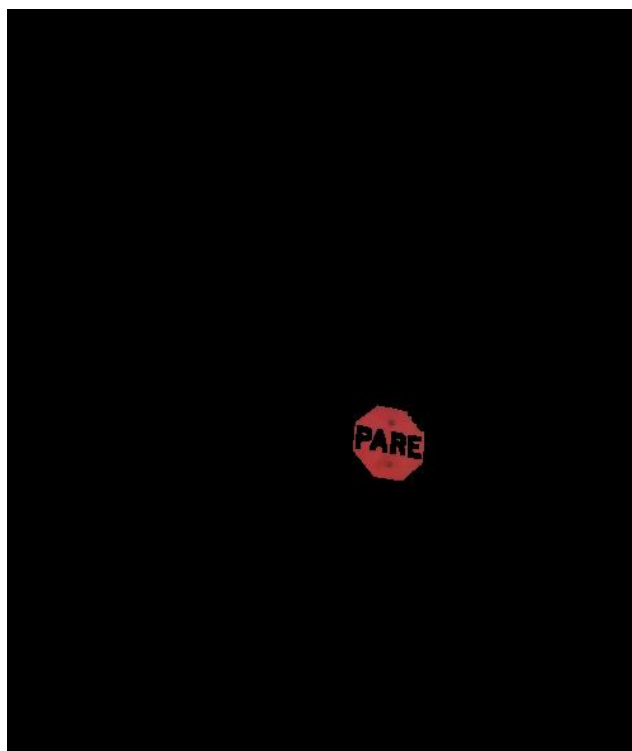
O ambiente em que as placas estão localizadas é bem diversificado e pode ocorrer imperfeições, ou seja alguns *pixels* podem estar dentro do intervalo escolhido e não fazerem

parte do ROI (região de interesse) que se quer obter. Essas "sujeiras" são removidas com filtros de erosão, que eliminam por área e proporção.

Quando a imagem é segmentada, podem constar pixels isolados que não fazem parte da região de interesse, com isso faz-se necessário que sejam eliminados esses ruídos para facilitar o processo. Na Figura IV.7 podemos observar a imagem após ter sido aplicado o filtro de erosão que consiste em usar um elemento estruturante para reduzir a imagem e eliminar as imperfeições. A intenção é remover elementos muito pequenos, por isso foi usado um elemento de 4 x 4, pelo método *cv2.erode()* e Bradski (2000) define a fórmula para o filtro de erosão como:

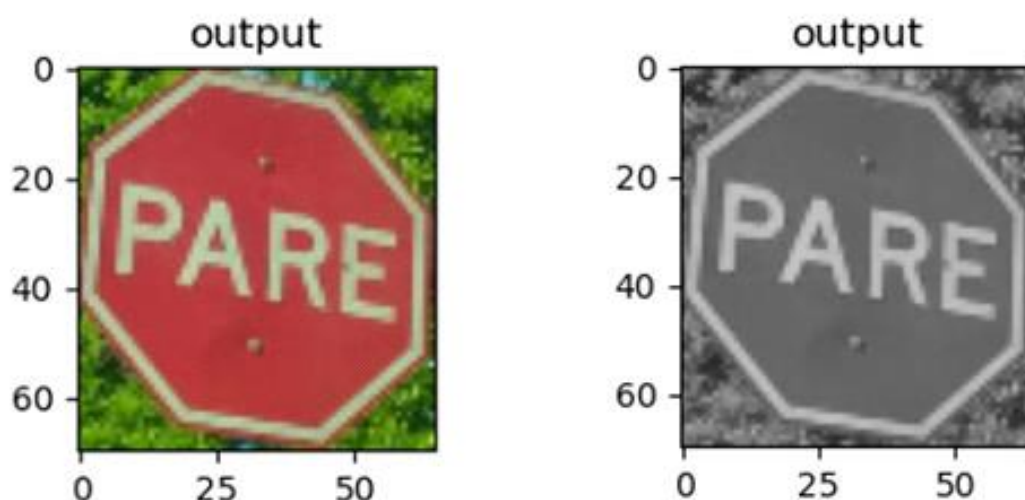
$$dst(x, y) = \min_{(x', y') : element(x', y') \neq 0} src(x + x', y + y')$$

Figura IV.7 - Imagem com filtro de erosão.



Na Figura IV.8 foi extraído o ROI utilizando o *cv2.findContours()* que detecta a mudança de cor da imagem e marca o contorno. Depois é feita a conversão de cor RGB para escala de Cinza, e para isso utiliza-se o a função *cv2.cvtColor()* da biblioteca *OpenCV*.

Figura IV.8 - Região de Interesse e escala de cinza.



### 4.3 Extração de Características

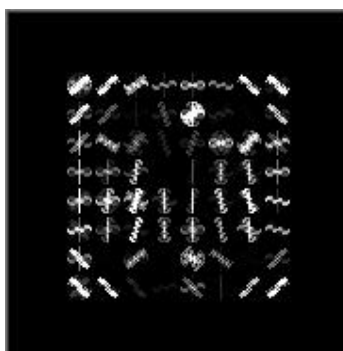
A etapa de extração de características da placa de sinalização é descrita através de um vetor numérico. No descritor HOG é feita a união de vetores e gradientes.

Nesta técnica é importante redimensionar a imagem e converter a imagem para escala de cinza. Logo após, as células são divididas em 8x8 pixels e os gradientes das imagens são calculados sobre esses pixels.

Para cada célula é extraído um histograma de 9 posições, essas podem variar de 0 a 180° e são calculadas através da análise dos valores de magnitude em relação aos valores de direção dos vetores gradientes.

Por último os blocos que incluem os valores de 4 histogramas são normalizados para evitar variações na iluminação. Toda célula do bloco é associada com as demais, gerando uma linha, chamada de histograma orientado a gradiente (HOG).

Figura IV.9 - Descritor HOG.



## 4.4 ESTRUTURA SVM

A classificação das imagens é feita utilizando a interface *Python* da biblioteca *Dlib*. A interface permite manipular a biblioteca a partir de classes e métodos *Python*. Foi instalado o Anaconda que nos permite criar pequenos ambientes virtuais e pode-se colocar a versão do *Python* da preferência e instalar outras bibliotecas através do prompt de comando. Para configurar o ambiente para a classificação com o uso de SVM em *Python*, roda-se o seguinte código a partir do terminal do Anaconda:

```
1 conda install -c conda-forge dlib
```

### 4.4.1 Treino

Para o treinamento das SVMs, são usadas classes criadas pela interface da biblioteca *Dlib*: *dlib.simple\_object\_detector\_training\_options()* e *dlib.train\_simple\_object\_detector()*.

Como as placas são simétricas, pode-se treinar um detector simétrico, isso facilita a obter o valor máximo dos dados treinados usando a classe *dlib.simple\_object\_detector\_training\_options()*.

A classe *dlib.train\_simple\_object\_detector()* é tipo uma máquina de vetores de suporte e, portanto possui um parâmetro *C* que deve se adequar ao treinamento, verificando qual detector funciona melhor em um conjunto de imagens de teste nas quais não foram treinadas. Esta classe treina um detector de objetos baseado em histograma orientado a gradientes de janela deslizante, que salva o detector final SVM, sendo a entrada um arquivo XML que lista as imagens no conjunto de dados de treinamento e também as posições das caixas. Abaixo podemos visualizar o Código IV.2 completo do treinamento das SVMs.

## Código IV.2 – Treinamento das SVMs

```







1  import dlib
2  opcoes = dlib.simple_object_detector_training_options()
3  opcoes.add_left_right_image_flips = True
4  opcoes.C = 1
5
6  dlib.train_simple_object_detector("treinamento_PLACAS.xml",
7  "detector_PLACAS.svm", opcoes)

```

O classificador é treinado e testado com diversos parâmetros de  $C$ , para melhorar a taxa de acerto do classificador, utilizando o mesmo conjunto de descritores. Os resultados da fase de teste para os valores do parâmetro  $C$  escolhidos são mostrados na Tabela IV.2 para todas as classes das placas de sinalização. Os testes foram feitos com 7 valores de  $C$  e todas as classes de imagens foram treinadas individualmente, o que apresentou melhor resultado foi com o índice 1, totalizando 91,67% de acerto. Para o valor de  $C = 0.1$ , nenhuma placa foi reconhecida.

Para as placas, “Proibido parar e estacionar”, “Pare”, “30 Km/h”, “Dê a Preferência” e “Curva a Esquerda”, o melhor resultado foi com o parâmetro 1 e apenas a placa “Quebra-molas” apresentou o melhor resultado com o parâmetro 5 e 7.

Tabela IV.2 - Parâmetros  $C$  utilizados no treinamento do classificador

<b>C</b>							<b>Total</b>
<b>0.1</b>	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	<b>0,00%</b>
<b>0.5</b>	50,00%	83,33%	58,33%	73,08%	66,67%	75,00%	<b>67,74%</b>
<b>1</b>	<b>90,90%</b>	<b>100,00%</b>	<b>81,82%</b>	<b>92,31%</b>	<b>100,00%</b>	85,00%	<b>91,67%</b>
<b>3</b>	81,82%	100,00%	81,82%	92,31%	100,00%	85,00%	<b>90,16%</b>
<b>5</b>	81,82%	100,00%	72,73%	92,31%	100,00%	<b>89,47%</b>	<b>89,38%</b>
<b>7</b>	81,82%	75,00%	50,54%	92,31%	100,00%	89,47%	<b>81,52%</b>
<b>10</b>	90,90%	75,00%	81,82%	88,46%	100,00%	85,00%	<b>79,29%</b>

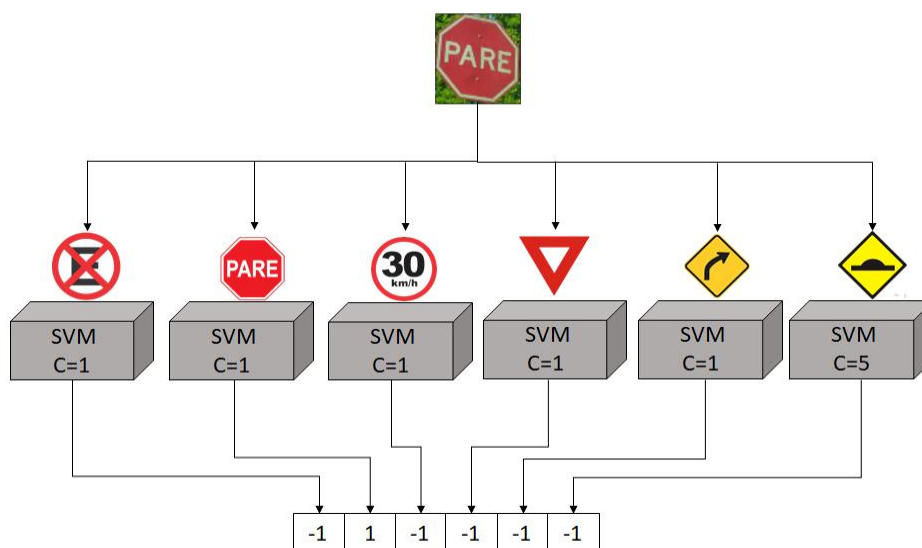
A implementação da fase de teste do classificador é a mesma que propriamente o processo de classificação, por isso esses dois processos são apresentados de forma única na próxima sessão.

#### 4.4.2 Teste e Classificação

Para o teste e classificação foram selecionadas 86 imagens contendo placas de sinalização e 20 imagens sem placas. A classificação tem como objetivo gerar a detecção final das placas de sinais de trânsito, através das caixas delimitadoras (*bounding boxes*) e com a descrição da imagem. Para isso, é feita a varredura de uma janela, onde o tamanho foi fixado. O vetor de características (HOG) é extraído e é feita uma classificação, usando o classificador que foi treinado.

Na Figura IV.10 são representadas as SVMs, onde foram alimentadas com informações importantes das imagens de entrada e foram classificadas através de uma lista de detectores, constituída por todas as SVMs que foram treinadas com suas respectivas classes.

Figura IV.10 - Modelo SVM.



As imagens são analisadas, e cada classe de imagem possui valor “1” caso seja uma ocorrência real de placa de sinalização e “-1” caso não seja. Na Figura IV.11 pode ser observada uma imagem que foi testada e classificada corretamente após todas as imagens serem analisadas pela lista formada pelas SVMs.



Figura IV.11 - Imagem classificada corretamente.



## 4.5 Resultados

A partir da implementação descrita anteriormente, foram realizados testes e reconhecimentos afim de apresentar os resultados da classificação das placas de sinais de trânsito. Também foi necessário capturar imagens de placas de sinais de trânsito brasileiras, pois no banco de dados alemão GTSDb de HOUBEN et al. (2013) não há placas amarelas e a incidência de sol é diferente.







Para isso foi utilizado a câmera do celular MotoOne da Motorola, para captura de imagens em resolução 500 x 595 e os testes/reconhecimentos foram realizados em um notebook Dell Core i7-6500, com 8 GB de memória e placa de vídeo AMD Radeon 2 GB.

Para cada uma das imagens foi verificado o Acerto (AC), caso a forma reconhecida de forma correta, também foi verificado os falsos positivos (FP), quando detecta uma placa onde não existe e o falso negativo (FN), quando não detecta uma placa que existe.

Na Tabela IV.3 é apresentado o resultado do reconhecimento das placas de sinais de trânsito. Observa-se através da tabela que o total de placas de sinalização classificadas

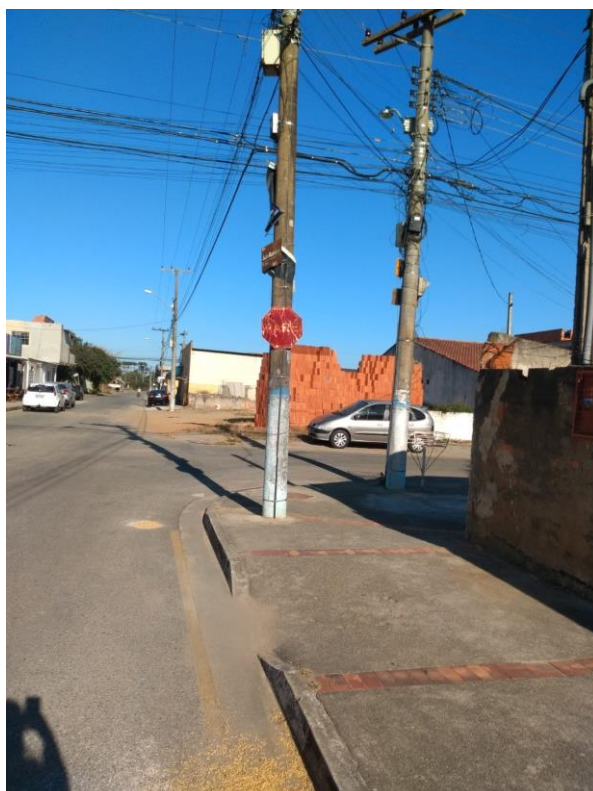
corretamente é superior a 89%. Nas SVMs foi utilizado o parâmetro  $C=5$  na classe “Quebra Molas” e  $C=1$  para as demais placas.

Tabela IV.3 - Resultado do Classificador

<b>Classes</b>	<b>Reconhecimento</b>	<b>AC</b>	<b>FP</b>	<b>FN</b>	<b>% TOTAL AC</b>
	10	08	02	02	80,00%
	16	13	01	03	81,25%
	06	06	02	-	100,00%
	08	07	02	-	87,50%
	04	04	01	-	100,00%
	22	20	-	02	90,91%
<b>Total</b>	<b>65</b>	<b>58</b>	<b>08</b>	<b>07</b>	<b>89,94%</b>

O algoritmo também apresentou falsos positivos e falsos negativos, sendo o valor mais significativo dos falsos positivos alcançando 10,06% do total de placas de sinalizações. Diversas razões que fazem com que algumas das placas não fossem classificadas, como por exemplo, deformações, iluminação e características extras (texto - km/h), que podem ser representadas como ruídos para as SVMs, desta forma não são identificadas corretamente ou nem são identificadas. Na Figura IV.12 é apresentada uma imagem que não foi reconhecida (FN).

Figura IV.12 - Placa não reconhecida.



O sistema proposto apresentou um bom resultado, mas para uma melhora significativa de assertividade das SVMs pode-se montar um banco de dados mais completo com as placas de trânsito do sistema brasileiro.

## CAPÍTULO V CONCLUSÃO

O trabalho apresenta estudos de técnicas de um sistema de reconhecimento de placas de sinais de trânsito de imagens obtidas de um banco de imagens alemão e de imagens capturadas no trânsito brasileiro.

O sistema proposto é formado por quatro etapas: pré-processamento, segmentação, descrição e classificação. Na fase de reconhecimento foi obtida eficácia superior a 89% e na de testes superior a 91%. Mesmo com resultados bons, é importante ressaltar que as imagens de treinamento, teste e reconhecimento, não são de um mesmo banco de dados, dificultando o reconhecimento, pois as placas brasileiras possuem mais informações, como por exemplo, km/h. Outra dificuldade é causada pela diferença climática, que faz com que a luminosidade que incidem sobre as placas alemãs seja menor em comparação com as brasileiras.

Ainda é necessário estudos de tecnologias que sejam mais eficazes para esse mesmo propósito, principalmente pelo aumento do número de veículos com piloto automático que requer tecnologias embarcadas para o auxílio na direção.

De acordo com os resultados obtidos e mencionados, considera-se que o objetivo do trabalho de reconhecer automaticamente placas de sinalização de trânsito foi alcançado, com as limitações citadas. Para trabalhos futuros sugere-se desenvolver um banco de imagens do sistema brasileiro de trânsito, utilizar ferramentas/câmeras que possuam alta resolução que se adaptam a diversas situações, comparar diferentes técnicas de reconhecimento com o trabalho apresentado e a migração do sistema para recursos mobile.

## REFERÊNCIAS BIBLIOGRÁFICAS

AZIZ, S.; MOHAMED, E. A.; YOUSSELF, F. Traffic Sign Recognition Based On Multi-feature Fusion and ELM. *Procedia Computer Science*, v.127, p 146-153, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050918301200>.

Blogspot Buulex. Modelos de cor RGB, CMYK, HSV e YUV. Disponível em: <http://buulex.blogspot.com/2016/11/modelos-de-cor-rgbcmyk-hsv-yuv.html>. Acesso em: 20 de maio de 2019.

BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, p.42, 2020.

BRAGATTO, T.A.; RUAS, G.; LAMAR, M. Uma comparação entre redes neurais artificiais e maquinas de vetores de suporte para reconhecimento de posturas manuais em tempo real., p. 1 – 6, 2016.

CREUSEN, I.M., WIJNHOFEN, R. G. J., HERBSCHLEB, E., & WITH, de P.H. N. Color exploitation in HOG-based traffic sign detection, *IEEE International Conference on Image Processing*. p. 2669-2672, 2010. Disponível em: <https://ieeexplore.ieee.org/document/5651637>

Docplayer. Forma padrão do sinal de regulamentação e a circulação nas seguintes cores. Disponível em: <https://docplayer.com.br/6405859-Anexo-ii-a-forma-padrao-do-sinal-de-regulamentacao-e-a-circular-nas-seguintes-cores.html>. Acesso em: 25 de junho de 2019.

ELLAHYANI, A.; ANSARI, M. E.; JAAFARI, I. E. Traffic sign detection and recognition based on random forests. *Applied Soft Computing*, v. 46, p. 805 – 815, 2016. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494616000028>.

GONZALES, R. C.; WOODS, R. E. *Processamento digital de imagens*. 3a edição. ed. [S.l.]: Pearson Prentice Hall, São Paulo, 2010.

HERCIK, R.; SROVNAL, V.; STANKUS, M.; SLABY, R.; VANUS, J. Autonomous Recognition System for Traffic Signs Detection from Camera Images. *The International Federation of Automatic Control*, p. 434-437, 2013. Disponível em: <https://kundoc.com/pdf-autonomous-recognition-system-for-traffic-signs-detection-from-camera-images-.html>

HOUBEN, S. et al. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: *International Joint Conference on Neural Networks* (submitted). [S.l.: s.n.], 2013.

LIRA, B. Implementação de módulo de Histograma de Gradientes Orientados para aplicações em streaming com prototipação em FPGA. *Graduação (Engenharia de Computação) – Universidade Federal de Pernambuco*, 2021.

MCCORMICK, C. HOG Person Detector Tutorial. Chris McCormick on WordPress.com. Disponível em: <http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/>. Acesso em: 5 set. 2022.

Medium. Máquinas de Vetores de Suporte – SVM. Disponível em: <https://medium.com/@cdabakoglu/what-is-support-vector-machine-svm-fd0e9e39514f>. Acesso em: 04 de setembro de 2022.

OLIVEIRA, F. Avaliação de Alternativas ao Mouse Utilizando Técnicas da Visão Computacional. Graduação (Engenharia Mecatrônica) - Universidade Federal de Santa Catarina, 2022.

ROSA, M. Reconhecimento de Placas de Sinalização de Trânsito via Processamento Digital de Imagem e Aprendizado de Máquina. Graduação (Engenharia de Telecomunicações) – Instituto Federal de Santa Catarina, São José, 2017.

SHELKE, C. J., KARDE, P. Traffic Sign Recognition. International Journal Of Computational Engineering Research. V. 2, p. 47-52, 2012. Disponível em: [https://pdfs.semanticscholar.org/d32e/e2c9f2e1bdb13305be85bd7aed8baa0d0c1d.pdf?\\_ga=2.60236845.1488299605.1561775922-2145903625.1561775922](https://pdfs.semanticscholar.org/d32e/e2c9f2e1bdb13305be85bd7aed8baa0d0c1d.pdf?_ga=2.60236845.1488299605.1561775922-2145903625.1561775922).

SILVA, F. A. da; ARTERO, A. O.; PAIVA, M. S. V. de; BARBOSA, R. L. Uma metodologia para detectar e reconhecer placas de sinalização de trânsito. Anais. São Paulo: Universidade Presbiteriana Mackenzie, 2012. Disponível em: [http://iris.sel.eesc.usp.br/wvc/Anais\\_WVC2012/pdf/97147.pdf](http://iris.sel.eesc.usp.br/wvc/Anais_WVC2012/pdf/97147.pdf).

SILVA, J. C., FARIAS, F. C., LIMA, V. C. F, SILVA, V. L. B, SEIJAS, L. M., BASTOS-FILHO, C. J. A. Classificação de Sinais de Trânsito Usando Otimização por Colmeias e Random Forest, ABRICOM - Associação Brasileira de Inteligência Computacional. P. 1-6, 2015. Disponível em: [http://abricom.org.br/eventos/cbic\\_2015/cbic2015\\_submission\\_166/](http://abricom.org.br/eventos/cbic_2015/cbic2015_submission_166/).

STELET, T. Segmentação e detecção de placas de sinalização nos modelos de cores HSI, HSV e HLS. Graduação (Ciência da Computação) – Universidade Estadual do Norte Fluminense Darcy Ribeiro, Campos dos Goytacazes, 2019.

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.