



Universidade Federal do Piauí  
Centro de Ciências da Natureza  
Programa de Pós-Graduação em Ciência da Computação

# **Descritores Locais e *Bag of Features* na Classificação de Placas de Trânsito**

**Hugo Santos Piauilino Neto**

**Número de Ordem PPGCC: M001**

**Teresina-PI, Agosto de 2018**



Hugo Santos Piauilino Neto

# **Descritores Locais e *Bag of Features* na Classificação de Placas de Trânsito**

**Dissertação de Mestrado** apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Prof. Dr. Kelson Rômulo Teixeira Aires

Teresina-PI

Agosto de 2018

---

Hugo Santos Piauilino Neto

Descritores Locais e *Bag of Features* na Classificação de Placas de Trânsito/  
Hugo Santos Piauilino Neto. – Teresina-PI, Agosto de 2018-  
62 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Kelson Rômulo Teixeira Aires

Dissertação (Mestrado) – Universidade Federal do Piauí – UFPI  
Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação, Agosto de 2018.

1. Placas de Trânsito. 2. *Bag of Features*. 3. Descritores Locais. I. Prof. Dr.  
Kelson Rômulo Teixeira Aires. II. Universidade Federal do Piauí. III. Descritores  
Locais e *Bag of Features* na Classificação de Placas de Trânsito

CDU 02:141:005.7

---



Universidade Federal do Piauí  
Centro de Ciências da Natureza  
Pós-Graduação em Ciência da Computação

Ata de defesa de dissertação de Mestrado em  
Ciência da Computação da Universidade Federal do  
Piauí, 29 de agosto de 2018.

Ao vigésimo nono dia do mês de agosto de dois mil e dezoito, às oito horas e trinta minutos no Centro de Ciências da Natureza da Universidade Federal do Piauí, teve início a 64ª defesa de dissertação de mestrado em Ciência da Computação, intitulada “**Descritores Locais e Bag of Features na Classificação de Placas de Trânsito**”, elaborada sob a orientação do **Prof. Kelson Rômulo Teixeira Aires**, do candidato **Hugo Santos Piaullino Neto**. A banca examinadora é composta pelos professores Kelson Rômulo Teixeira Aires, Laurindo de Sousa Britto Neto e Rodrigo de Melo Souza Verás, todos do Programa de Pós-Graduação em Ciência da Computação desta Universidade, e Prof. Iális Cavalcante de Paula Júnior pertencente a Universidade Federal do Ceará. A Banca Examinadora, sob presidência do Prof. Kelson Rômulo Teixeira Aires resolveu: **(X) Aprovar e conceder 60 dias para entrega da versão final do trabalho** ou **( ) Reprovar**. Para constar, eu, Justina Célia Soares Cavalcante, secretária do PPGCC, lavrei a presente ata.

Teresina, 29 de agosto de 2018.

Prof. Kelson Rômulo Teixeira Aires  
(Presidente da Banca Examinadora)

Prof. Laurindo de Sousa Britto Neto  
(Examinador Interno)

Prof. Rodrigo de Melo Souza Verás  
(Examinador Interno)

Prof. Iális Cavalcante de Paula Júnior



*Aos meus pais, Maria da Conceição Ferreira Piauilino e Péricles Acioli Lins, por todo o apoio e dedicação necessário para que esse sonho se realizasse.*



# Agradecimentos

Agradeço a Deus.

Agradeço aos meus pais, Maria da Conceição Ferreira Piauilino e Péricles Acioli Lins, por todo o investimento motivacional, intelectual e financeiro que me permitiram caminhar até aqui.

Aos meus irmãos, Rebeca Maria Piauilino Lins e Péricles Accioly Lins Junior, por sempre me fornecer o apoio necessário.

À minha companheira, Nayara Holanda de Moura, pelo incessante apoio, incentivo, carinho e amor em todos os momentos.

Agradeço ao meu orientador, Kelson Rômulo Teixeira Aires, por todos os conselhos, pela paciência e ajuda nesse período.

Aos meus amigos Matheus Elvas, Luís Guilherme (GT), Neto Araújo (Netola), Francisco Carlos (Juninho), Francisco Wender (Manos), José Neto (Zé Note), Eugênio Carvalho (GG), Maíla Lima (Maiala) e Joselito Junior (Jojó) pela companhia de todas as horas, nos momentos de alegria e tristeza.

À todo o corpo docente do Curso de Bacharelado em Ciência da Computação e do Programa de Pós-graduação em Ciência da Computação da Universidade Federal do Piauí.



*“Listen and learn.  
Listen to the highest and the lowest.  
Learn from the prince, and from the shepherd.  
And, remember, the greatest virtue is humility.”  
(Autor Desconhecido)*



# Resumo

A área de Sistemas de Transportes Inteligentes surgiu como um esforço para tornar o transporte mais seguro, confiável e eficaz ao empregar diversas metodologias computacionais em ambientes de transporte. Uma das ramificações mais notáveis nessa área é o desenvolvimento de Sistemas de Apoio ao Condutor, com o objetivo de promover segurança e aprimorar as habilidades do condutor humano. Esses sistemas podem atuar como copilotos, monitorando continuamente o ambiente, fornecendo ao condutor informações atualizadas das variáveis monitoradas, e destacando possíveis ameaças à segurança. Além disso, podem executar, parcialmente ou totalmente, algumas tarefas anteriormente praticadas apenas por seres humanos. Uma das principais ações executadas por condutores, durante o processo de deslocamento, é o reconhecimento da sinalização de trânsito presente na via. Por diversas vezes, condutores não respeitam a sinalização por desatenção ou por estarem em situações de tráfego intenso. Em momentos semelhantes, sistemas que reconheçam placas de trânsito podem fornecer informações cruciais que poderiam ser ignoradas pelo condutor humano. Este trabalho propõe um modelo de classificação de placas de trânsito com a utilização de descritores locais e *Bag of Features*. Para a composição do modelo proposto foram utilizados os detectores de pontos de interesse SIFT e FAST, os descritores de características SIFT e BRIEF, os algoritmos de agrupamento *K-Means* e *Mini Batch K-Means* e os classificadores SVM multiclasse com abordagem *One-vs-One* e *One-vs-All*. O modelo proposto foi treinado em bases de imagens de placas de trânsito da Alemanha, Bélgica e Brasil, em que apenas placas que não possuem informações essencialmente por texto foram consideradas. Para a realização dos testes, o modelo treinado foi executado e as métricas de avaliação de desempenho resultantes foram analisadas. As maiores acurácias médias obtidas foram de 84,27% na base de placas de trânsito da Alemanha, 96,67% na base de placas de trânsito da Bélgica e 94,43% na base de placas de trânsito do Brasil. Os índices *Kappa* obtidos pelos melhores resultados de cada base de imagens foram considerados “excelentes”.

**Palavras-chaves:** descritores locais. bag of features. placas de trânsito. classificação.



# Abstract

The Intelligent Transportation Systems area has emerged as an effort to make transportation more safer, reliable and effective by employing various computational methodologies in transportation environments. One of the most notable ramifications in this area is the development of Driver Support Systems (DSS's), with the aim of promoting safety and improving the skills of the human driver. DSS's can act as co-pilots, continually monitoring the environment, providing the driver with up-to-date information on monitored variables, and highlighting possible security threats. In addition, they can perform, partially or totally, some tasks previously practiced only by humans. One of the main actions carried out by drivers during the process of displacement is the recognition of traffic signs present on the road. On several occasions, drivers do not respect the signs for lack of attention or for being in situations of intense traffic. At similar times, DSS's that recognize traffic signs can provide crucial information that could be ignored by the human driver. This work proposes a classification model of traffic signs using local descriptors and Bag of Features. For the composition of the proposed model, the SIFT and FAST detectors, the SIFT and BRIEF descriptors, the K-Means and Mini Batch K-Means clustering algorithms and the multi-class SVM classifiers with One-vs-One and One-vs-All approaches was used. The proposed model was trained on traffic signs datasets from Germany, Belgium and Brazil, where only signs that do not have information essentially by text were considered. In order to perform the tests, the trained model was executed and the resulting performance evaluation metrics were analyzed. The highest average accuracy obtained were 84.27% on the Germany Traffic Sign Dataset, 96.67% on the Belgium Traffic Sign Dataset and 94.43% on the Brazil Traffic Sign Dataset. The Kappa index obtained by the highest results of each dataset were considered "excellent".

**Keywords:** local descriptors. bag of features. traffic signs. classification.



# Lista de ilustrações

Figura 1 – Fluxograma de um modelo de reconhecimento de placas de trânsito. A cena da via é processada pelo módulo de detecção de regiões de interesse. As regiões candidatas detectadas são enviadas para o módulo de classificação e são rotuladas como uma das classes conhecidas, ou como objeto desconhecido. Fonte: Elaborado pelo Autor (2017). . . . .	2
Figura 2 – Tipos de placas de regulamentação e advertência. Fonte: iSinaliza.com.	10
Figura 3 – Exemplos da diversidade de formas de placas de trânsito. . . . .	11
Figura 4 – Placas de trânsito modificadas por variações climáticas, temporais e físicas. . . . .	12
Figura 5 – Amostra de imagens de placas de trânsito presentes na BTSD. Fonte: Timofte, Zimmermann e Gool (2009). . . . .	13
Figura 6 – Amostras de algumas das classes da BTSD. Fonte: Timofte, Zimmermann e Gool (2009). . . . .	13
Figura 7 – Amostra de imagens de placas de trânsito presentes na GTSRB. Fonte: Stallkamp et al. (2012). . . . .	14
Figura 8 – Amostras de algumas das classes da GTSRB. Fonte: Stallkamp et al. (2012). . . . .	15
Figura 9 – Amostra de imagens de placas de trânsito presentes na BraTSD. Fonte: Elaborado pelo Autor. . . . .	15
Figura 10 – Amostras de algumas das classes da BraTSD. Fonte: Elaborado pelo Autor. . . . .	16
Figura 11 – Exemplos de pontos de interesses detectados pelo SIFT em imagens de placas de trânsito. Fonte: Elaborado pelo Autor. . . . .	18
Figura 12 – Exemplos de pontos de interesses detectados pelo FAST em imagens de placas de trânsito. Fonte: Elaborado pelo Autor. . . . .	19
Figura 13 – Representação do gradiente da imagem e da descrição do ponto de interesse para o descritor de características SIFT. Fonte: Elaborado pelo Autor. . . . .	20
Figura 14 – Diferentes abordagens para encontrar os vetores $X$ e $Y$ do descritor BRIEF. Fonte: Calonder et al. (2010). . . . .	22
Figura 15 – Etapas do modelo proposto. A parte esquerda representa o processo de detecção e extração de características das placas de trânsito. Na parte central, as características são submetidas ao algoritmo de agrupamento para a criação do dicionário de palavras visuais. Por fim, a parte direita representa o processo de quantização dos descritores, onde um histograma de características é gerado. Fonte: Elaborado pelo Autor. . . . .	23

Figura 16 – Algoritmo <i>K-Means</i> . Os exemplos de treinamento e os centroides são representados como pontos e cruces, respectivamente. (a) Conjunto de dados original. (b) Inicialização aleatória dos $k$ centroides. (c-f) Ilustração da execução de duas iterações do <i>K-Means</i> . Em cada iteração, atribuímos cada exemplo de treinamento ao centroide mais próximo (mostrado ao “pintar” cada ponto da mesma cor que o centroide do grupo ao qual foi atribuído). Então movemos cada centroide para a posição média dos pontos atribuídos a ele. Fonte: Michael I. Jordan. . . . .	25
Figura 17 – Comparação entre os resultados obtidos para o agrupamento de um conjunto de dados com $k = 3$ , utilizando os algoritmos <i>K-Means</i> e <i>Mini Batch K-Means</i> . O <i>K-Means</i> se mostrou mais lento do que o <i>Mini Batch K-Means</i> . Fonte: Elaborado pelo Autor. . . . .	26
Figura 18 – SVM multiclasse com abordagem <i>One-versus-One</i> (O-v-O). Fonte: Elaborado pelo Autor. . . . .	28
Figura 19 – SVM multiclasse com abordagem <i>One-versus-All</i> (O-v-A). Fonte: Elaborado pelo Autor. . . . .	29
Figura 20 – Etapas do modelo proposto. Fonte: Elaborado pelo Autor. . . . .	33
Figura 21 – Comparação do desempenho do modelo proposto na BraTSD. . . . .	41
Figura 22 – Comparação do desempenho geral do modelo proposto na BTSD. . . . .	43
Figura 23 – Comparação do desempenho geral do modelo proposto na GTSRB. . . . .	44

# Lista de tabelas

Tabela 1	–	Quadro-resumo dos trabalhos relacionados. . . . .	8
Tabela 2	–	Nível de precisão da classificação, de acordo com o índice <i>Kappa</i> . . . .	30
Tabela 3	–	Resultado da classificação com a utilização do descritor e detector SIFT, algoritmo de agrupamento <i>K-Means</i> e os classificadores SVM <i>One-versus-One</i> e SVM <i>One-versus-All</i> nas imagens da BraTSD. . . .	38
Tabela 4	–	Resultado da classificação com a utilização do descritor e detector SIFT, algoritmo de agrupamento <i>Mini Batch K-Means</i> e os classificadores SVM <i>One-versus-One</i> e SVM <i>One-versus-All</i> nas imagens da BraTSD. . . .	39
Tabela 5	–	Resultado da classificação com a utilização do detector FAST e descritor BRIEF, algoritmo de agrupamento <i>K-Means</i> e os classificadores SVM <i>One-versus-One</i> e SVM <i>One-versus-All</i> nas imagens da BraTSD. . . .	40
Tabela 6	–	Desempenho do modelo proposto na BTSD. . . . .	42
Tabela 7	–	Desempenho do modelo proposto na GTSRB. . . . .	44



# Lista de abreviaturas e siglas

A	Acurácia
AMD	Advanced Micro Devices
ANN	Artificial Neural Network
BBF	Best Bin First
BCNN	Branch Convolution Neural Network
BLOB	Binary Large Object
BoF	Bag of Features
BoW	Bag of Words
BraTSD	Brazilian Traffic Sign Dataset
BRIEF	Binary Robust Independent Elementary Features
BTSD	Belgium Traffic Sign Dataset
C-HMAX	Color Hierarchical Max-pooling
CNN	Convolution Neural Network
CTB	Coódigo de Trânsito Brasileiro
DC	Departamento de Computação
DCNN	Deep Convolution Neural Network
DoG	Difference of Gaussian
DSRC	Dedicated Short-Range Communications
DSS	Driver Support Systems
ELM	Extreme Learning Machine
FAST	Features from Accelerated Segment Test
FN	Falso Negativo
FP	Falso Positivo

FPGA	Field Programmable Gate Array
GB	Gigabyte
GHz	Gigahertz
GP	Genetic Programming
GTSRB	German Traffic Sign Recognition Benchmark
HFTC	Haptic Feedback for Traction Control
HMAX	Hierarchical Max-pooling
HOG	Histogram of Oriented Gradients
HOGv	Histogram of Oriented Gradient variant
ICN	Information-Centric Networking
INNC	Iterative Nearest Neighbors Classifier
INNLP	Iterative Nearest Neighbors based Linear Projection
ITS	Intelligent Transportation Systems
K	Kappa
LDA	Linear Discriminant Analysis
LSH	Locality Sensitive Hasing
NNC	Nearest Neighbor Classifier
O-v-A	One-versus-All
O-v-O	One-versus-One
PCA	Principal Component Analysis
PD	Pedestrian Detection
PHOG	Pyramid Histogram of Oriented Gradients
PNG	Portable Network Graphics
PPM	Portable Pixel Map
RAM	Random Access Memory
RD	Rails Detection

RF	Random Forest
RGB	Red-Green-Blue
SCNN	Simple Convolutional Neural Network
SIFT	Scale-Invariant Feature Transform
SP	Smart Parking
SRC	Sparse Representation based Classifier
SRLP	Sparse Representation based Linear Projection
SURF	Speeded-Up Robust Features
SVM	Support Vector Machines
TSD	Traffic Sign Detection
UFPI	Universidade Federal do Piauí
V2D	Vehicle-to-Device
V2G	Vehicle-to-Grid
V2I	Vehicle-to-Infrastructure
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad-Hoc Network
VID	Visual Illusions Detection
VipLab	Laboratório de Processamento Digital de Imagens e Visão Computacional
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WAVE	Wireless Access in Vehicular Environments



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Objetivo	3
1.2	Contribuições	3
1.3	Organização da Dissertação	4
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>5</b>
2.1	Considerações Finais	8
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>9</b>
<b>3.1</b>	<b>Sinalização de Trânsito</b>	<b>9</b>
3.1.1	Sinalização Vertical	9
3.1.2	Propriedades das Placas de Trânsito	11
<b>3.2</b>	<b>Bases de Imagens</b>	<b>12</b>
3.2.1	<i>Belgium Traffic Sign Dataset (BTSD)</i>	12
3.2.2	<i>German Traffic Sign Recognition Benchmark (GTSRB)</i>	14
3.2.3	<i>Brazilian Traffic Sign Dataset (BraTSD)</i>	14
<b>3.3</b>	<b>Métodos</b>	<b>15</b>
3.3.1	Detectores	16
3.3.2	Descritores	19
3.3.3	<i>Bag of Features (BoF)</i>	22
3.3.4	Dicionários Visuais	23
3.3.5	Quantização dos Descritores	26
3.3.6	Classificadores	27
<b>3.4</b>	<b>Métricas de Avaliação dos Resultados</b>	<b>29</b>
<b>3.5</b>	<b>Método de Avaliação dos Classificadores</b>	<b>30</b>
<b>3.6</b>	<b>Considerações Finais</b>	<b>32</b>
<b>4</b>	<b>MODELO PROPOSTO</b>	<b>33</b>
4.1	Extração de Características	33
4.2	<i>Bag of Features</i>	34
4.3	Classificação	36
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>37</b>
5.1	<i>Brazilian Traffic Sign Dataset (BraTSD)</i>	37
5.2	<i>Belgium Traffic Sign Dataset (BTSD)</i>	42
5.3	<i>German Traffic Sign Recognition Benchmark (GTSRB)</i>	43

5.4	Discussão e Considerações Finais . . . . .	44
6	CONCLUSÃO E TRABALHOS FUTUROS . . . . .	47
6.1	Conclusão . . . . .	47
6.2	Trabalhos Futuros . . . . .	47
	REFERÊNCIAS . . . . .	49

# 1 Introdução

Um crescimento massivo de pesquisas na área de *Intelligent Transportation Systems* (ITS) foi observado nos últimos anos. Tal iniciativa surgiu como um esforço para tornar o transporte mais seguro, confiável e eficaz ao empregar diversas metodologias computacionais, como *Genetic Programming* (GP) (LÓPEZ et al., 2018), *Information-Centric Networking* (ICN) (LOSCIALE et al., 2018), *Wireless Access in Vehicular Environments/Dedicated Short-Range Communications* (WAVE/DSRC) (CHEN; HO, 2018), *Vehicular Ad-Hoc Network* (VANET) (SHEN et al., 2018), entre outras.

Uma importante ramificação de ITS é a construção de dispositivos autônomos que possam atuar de modo inteligente, em certa medida, em ambientes de transporte. Um exemplo de grande interesse atualmente é o desenvolvimento de automóveis inteligentes que incorporem vários sistemas de segurança ativos e passivos, dispositivos de navegação ou sistemas *Vehicle-to-Everything* (V2X). Sistemas V2X estão em constante desenvolvimento e permitem que veículos se comuniquem com partes móveis ou fixas do ambiente de transporte ao redor (SALEM; ELHILLALI; NIAR, 2018; BONJORN; FOUKALAS; POP, 2018; CHEN et al., 2018; ABBAS; FAN, 2018). Essa tecnologia incorpora outros tipos mais específicos de comunicação, como *Vehicle-to-Infrastructure* (V2I), *Vehicle-to-Vehicle* (V2V), *Vehicle-to-Pedestrian* (V2P), *Vehicle-to-Device* (V2D) e *Vehicle-to-Grid* (V2G).

Um tópico aberto na área de ITS é o desenvolvimento de *Driver Support Systems* (DSS's) com o objetivo de promover segurança e aprimorar as habilidades do condutor humano (PETERMEIJER et al., 2015; TARAMOV; SHILOV, 2017; KASHEVNIK et al., 2017). Esses sistemas podem atuar como copilotos, monitorando continuamente o ambiente, fornecendo ao condutor informações atualizadas das variáveis monitoradas e destacando possíveis ameaças à segurança. Aplicações DSS's de diversos tipos podem ser encontradas na literatura, tais como: *Haptic Feedback for Traction Control* (HFTC) (CORNO, 2013), *Visual Illusions Detection* (VID) (KASHIHARA, 2014), *Rails Detection* (RD) (SELVER et al., 2016), *Pedestrian Detection* (PD) (TOPRAK et al., 2018; WANG; ZHANG, 2018), *Smart Parking* (SP) (COULIBALY et al., 2018), *Traffic Sign Detection* (TSD) (ZENG et al., 2017; PHU; OO, 2018), entre outras.

A percepção visual é, de longe, a fonte de dados mais importante para condutores humanos. Um dos dados visuais que desempenha um papel fundamental na segurança do trânsito é a sinalização. Segundo o Código de Trânsito Brasileiro (CTB), a sinalização é um conjunto de sinais de trânsito e dispositivos de segurança colocados na via pública com o objetivo de garantir sua utilização adequada, possibilitando melhor fluidez no trânsito e maior segurança dos veículos e pedestres que nela circulam (BRASIL, 1997).

Por muitas vezes, condutores não respeitam a sinalização por pura desatenção, ou por estarem em situações de tráfego intenso. Em momentos como esse, um DSS que reconheça sinalizações de trânsito pode funcionar como um copiloto e fornecer informações que normalmente seriam ignoradas pelo condutor (WANG, 2018; WEI; RUNGE; XIAOLEI, 2018; SüRüCü; DOĞAN, 2018; ROXAS et al., 2018).

A sinalização vertical, categoria que engloba sinais colocados nas laterais ou acima das vias por meio de placas que transmitem uma mensagem aos usuários, é um dos tipos de sinalização mais abordado na literatura (MöGELMOSE; TRIVEDI; MOESLUND, 2012; DESHPANDE; SUBASHINI, 2017). A Figura 1 apresenta um exemplo de modelo de reconhecimento de placas de trânsito. Inicialmente, as imagens são adquiridas por meio de uma câmera digital. Em cada imagem, as potenciais placas de trânsito são localizadas pelo módulo de detecção. Cada uma das regiões candidatas é então analisada pelo módulo de classificação e atribuída a uma classe de placa conhecida ou rejeitada. Note que, embora o termo reconhecimento refira-se ao processamento completo, detecção e classificação são entendidos como dois estágios subjacentes.

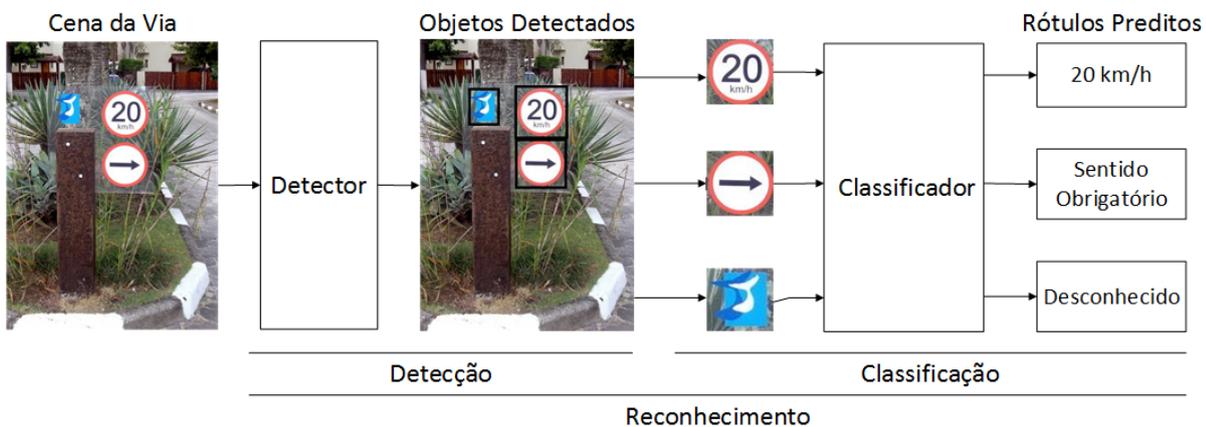


Figura 1 – Fluxograma de um modelo de reconhecimento de placas de trânsito. A cena da via é processada pelo módulo de detecção de regiões de interesse. As regiões candidatas detectadas são enviadas para o módulo de classificação e são rotuladas como uma das classes conhecidas, ou como objeto desconhecido. Fonte: Elaborado pelo Autor (2017).

Em um ambiente ideal, sem ruído, placas de trânsito podem ser classificadas por simples correspondência das imagens capturadas pelo sistema com os modelos derivados do padrão técnico definido. No entanto, uma série de fontes de ruído, como iluminação variável, obstruções, condições climáticas, vibrações ou interferências dos sensores, deterioram as informações das imagens enviadas para o sistema de reconhecimento. Os modelos derivados do padrão técnico definido não codificam esses fatores e, portanto, não são representativos das condições reais.

Uma abordagem mais aprimorada é desenvolver um reconhecimento com base em

condições reais. Portanto, para generalizar conceitos fracamente definidos e reconhecer observações ruidosas, um modelo de reconhecimento de placas de trânsito precisa ser treinado a partir de exemplos do mundo real.

Algumas pesquisas optam por combinar detecção e classificação (FAN; ZHANG, 2015; ZHU et al., 2016; SHEIKH; KOLE; MAITY, 2016; GUAN et al., 2018; ROXAS et al., 2018; WEI; RUNGE; XIAOLEI, 2018; WANG, 2018). Já outros estudos, que consideram uma grande quantidade de classes de placas de trânsito, preferem separar os dois estágios (ZENG et al., 2017; PENG et al., 2017; RATNASARI; IRAWAN; SETIANINGSIH, 2017; ARIFIN; IRAWAN; SETIANINGSIH, 2017; SÜRÜCÜ; DOĞAN, 2018; PHU; OO, 2018). Nesta pesquisa, apenas o estágio de classificação de placas de trânsito foi abordado.

## 1.1 Objetivo

No âmbito de *Driver Support Systems*, este trabalho está inserido no subproblema de classificação de sinalizações de tráfego, mais precisamente de placas de trânsito.

O objetivo geral desta pesquisa é desenvolver um modelo de classificação de placas de trânsito. Após a escolha do objetivo geral, os seguintes objetivos específicos foram propostos:

- elaboração de uma base de imagens de placas de trânsito do Brasil;
- desenvolvimento de um método de classificação direcionado para placas de trânsitos utilizando descritores locais e *Bag of Features*;
- coleta e análise de resultados do método proposto, aplicado em imagens de placas de trânsito da Alemanha, Bélgica e Brasil.

## 1.2 Contribuições

A principal contribuição deste trabalho é disponibilizar um modelo de classificação de placas de trânsito, que execute em *hardwares* com poder computacional reduzido. Outra importante contribuição está na elaboração de uma base de imagens de placas de trânsito do território brasileiro. Além disso, foi identificado que a utilização de algoritmos de tempo real pode prover excelente desempenho, mesmo na presença de recursos escassos. Por fim, a avaliação de diferentes combinações de algoritmos em diversas bases de imagens será apresentada. O método proposto, em resumo, é composto pelos seguintes algoritmos:

- detecção de pontos de interesse por meio do algoritmo FAST (ROSTEN; PORTER; DRUMMOND, 2010);

- descrição de características usando o algoritmo BRIEF (CALONDER et al., 2010);
- criação do dicionário visual utilizando o algoritmo *Mini Batch K-Means* (SCULLEY, 2010);
- classificação das placas de trânsito por meio do SVM multiclasse com abordagem *One-vs-All* (CORTES; VAPNIK, 1995).

### 1.3 Organização da Dissertação

Esta dissertação está organizada em 6 (seis) capítulos. Após a introdução (Capítulo 1), que apresentou o problema estudado, os objetivos e as contribuições, os próximos capítulos estão organizados da seguinte forma:

O Capítulo 2 demonstra uma análise dos trabalhos mais relevantes relacionados a classificação de placas de trânsito. No Capítulo 3 apresenta a fundamentação teórica para a realização da pesquisa, demonstrando as sinalizações de trânsitos com suas propriedades e categorias existentes. Por fim, apresenta e analisa as bases de imagens de placas de trânsito e descreve as métricas de avaliação de desempenho utilizadas.

O Capítulo 4 descreve o modelo proposto neste trabalho, com uma análise dos seus componentes e suas opções de configuração. No Capítulo 5 demonstra a descrição dos experimentos realizados juntamente com os resultados obtidos. Por fim, o Capítulo 6 apresenta as considerações finais relacionadas ao método proposto e aponta direcionamentos para a continuidade da pesquisa.

## 2 Estado da Arte

Nos últimos anos, a classificação de placas de trânsito atraiu a atenção de muitos pesquisadores em virtude da publicidade empregada pela mídia no desenvolvimento de veículos autônomos. A fim de fornecer uma concepção clara do problema proposto pelo presente trabalho, fez-se necessária a pesquisa, o estudo e a formulação de ideias sobre o tema. Para tanto, são apresentados a seguir alguns trabalhos relevantes da literatura que circundam a área de classificação de placas de trânsito. Eles servem como base teórica para a sustentação científica e definem o foco com o qual se formularão hipóteses para melhor entender a situação-problema. O estudo destes trabalhos irá, conceitualmente, permitir a comparação do modelo proposto com as abordagens do estado da arte para as bases de imagens *Belgium Traffic Sign Dataset* (BTSD) (TIMOFTE; ZIMMERMANN; GOOL, 2009) e *German Traffic Sign Recognition Benchmark* (GTSRB) (HOUBEN et al., 2013). As descrições detalhadas da BTSD e GTSRB serão realizadas nas Seções 3.2.1 e 3.2.2, respectivamente. A seguir, são apresentados os trabalhos relacionados:

Zaklouta e Stanculescu (2012) apresentaram uma comparação entre os classificadores *Random Forest* (RF) (BREIMAN, 2001), K-D Tree (BENTLEY, 1975) e Support Vector Machines (SVM) (CORTES; VAPNIK, 1995) no processamento das características de imagens de placas de trânsito da Alemanha extraídas pelo *Histogram of Oriented Gradients* (HOG) (DALAL; TRIGGS, 2005). A combinação dos algoritmos HOG + RF se demonstrou mais eficiente e obteve uma acurácia máxima de 97,2% em testes realizados nos particionamentos preestabelecidos da GTSRB.

Mathias et al. (2013) forneceram uma comparação ampla de diversos algoritmos de detecção e classificação, porém, conforme o direcionamento desta pesquisa, apenas os experimentos relacionados a classificação de placas de trânsito são analisados. Esse trabalho demonstrou que muitos dos métodos utilizados para classificar imagens de faces e dígitos podem ser aplicados na classificação de placas de trânsito devido ao estreito relacionamento entre essas categorias, em que apenas variações limitadas a pose e aparência são apresentadas. A operação de classificação de placas de trânsito desse trabalho foi dividida em três estágios: extração de características, redução de dimensionalidade e classificação.

No primeiro estágio foram utilizados os métodos de extração de características *Histogram of Oriented Gradients* (HOG) (FREEMAN; ROTH, 1994) e *Pyramid Histogram of Oriented Gradients* (PHOG) (BOSCH; ZISSERMAN; MUNOZ, 2007) com a configuração de parâmetros proposta em Maji e Malik (2009). Já no segundo estágio, os redutores de dimensionalidade *Linear Discriminant Analysis* (LDA) (FISHER, 1938; MARTINEZ;

KAK, 2001), *Sparse Representation based Linear Projection* (SRLP) (TIMOFTE; GOOL, 2011) e *Iterative Nearest Neighbors based Linear Projection* (INNLP) (TIMOFTE; GOOL, 2012) foram aplicados. No estágio final, os classificadores *Nearest Neighbor Classifier* (NNC) (ALTMAN, 1992), *Sparse Representation-based Classifier* (SRC) (WRIGHT et al., 2009), *Iterative Nearest Neighbors Classifier* (INNC) (TIMOFTE; GOOL, 2012) e *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995) foram executados.

A combinação dos algoritmos PHOG + INNLP + SRCA obtiveram uma acurácia máxima de 97,83% em testes realizados nos particionamentos preestabelecidos da BTSD. Já para os particionamentos preestabelecidos da GTSRB, os algoritmos (PHOG + HOG) + INNLP + INNC alcançaram uma acurácia máxima de 98,53%. Conforme observado em Mathias et al. (2013), apesar das altas taxas encontradas de acurácia, é necessário um alto poder de processamento e uma quantidade extensa de memória para o funcionamento da combinação. Além disso, o tempo de execução é razoavelmente alto, não possibilitando o seu uso em aplicações de tempo real.

Yang, Zhou e Deng (2013) expuseram um modelo cognitivo para estender o *Hierarchical Max-pooling* (HMAX) (RIESENHUBER; POGGIO, 1999) com um algoritmo de extração de cores inspirado no *stream* da visão colorida do cérebro humano, denominado de Color-HMAX ou C-HMAX. Após o processamento do modelo cognitivo, a saída do C-HMAX é enviada para o classificador *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995). Conforme observado em Yang, Zhou e Deng (2013), esse método possui limitações quanto ao brilho das instâncias utilizadas. Imagens “muito claras” ou “muito escuras” são descartadas, pois as características de cores não são óbvias para o algoritmo. O modelo proposto alcançou uma acurácia máxima de 98,41% nos testes realizados em um subconjunto dos particionamentos preestabelecidos da GTSRB, contendo apenas imagens “aceitáveis” pelo algoritmo.

Jurišić, Filković e Kalafatić (2015) propuseram o modelo OneCNN para a classificação de placas de trânsito. Esse modelo é baseado na *Simple Convolutional Neural Network* (SCNN) proposta no trabalho de Ciresan, Meier e Schmidhuber (2012). *Branch offs* (desvios) são incluídos na camada convolucional, criando assim uma arquitetura multi-escala, como proposto em Sermanet e LeCun (2011). Além disso, foi aplicada a estratégia *dropout*<sup>1</sup> (SRIVASTAVA et al., 2014). Foram treinados trinta modelos da OneCNN para cada base de imagens. Foi obtida uma acurácia média de 98,17% para a BTSD e 99,11% para a GTSRB. Conforme demonstrado em Jurišić, Filković e Kalafatić (2015), devido a alta complexidade do modelo, é indispensável a utilização de placas de vídeo de alto desempenho para a execução do treinamento e teste da abordagem em tempo hábil.

Qasaimeh, Sagahyroon e Shanableh (2015) apresentaram um modelo com-

<sup>1</sup> Estratégia que desliga randomicamente os neurônios artificiais para promover alguns dos benefícios dos modelos de múltiplas redes, comprimido para modelos de rede única.

posto pelos algoritmos *Scale-Invariant Feature Transform* (SIFT) (LOWE, 1999), *Bag of Features* (BoF) (CSURKA et al., 2004) e *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995) inserido em uma *Field Programmable Gate Array* (FPGA) (KUON; TESSIER; ROSE, 2008) para a classificação de placas de trânsito. Diversos aperfeiçoamentos foram aplicados aos algoritmos utilizados na tentativa de executar o modelo proposto na FPGA em tempo real e com desempenho aceitável. Foi obtida uma acurácia máxima de 78,00% em um subconjunto dos particionamentos preestabelecidos da BTSD, onde foram utilizadas 40 imagens de 5 classes distintas, perfazendo um total de 200 imagens.

Supriyanto, Luthfiarta e Zeniarja (2016) propuseram uma abordagem não supervisionada para a classificação de placas de trânsito utilizando os algoritmos *Speeded-Up Robust Features* (SURF) (BAY; TUYTELAARS; GOOL, 2006), *Bag of Features* (BoF) (CSURKA et al., 2004) e *K-Means* (MACQUEEN, 1967). Em testes realizados nos particionamentos preestabelecidos da GTSRB, foi obtida uma acurácia máxima de 15,21%. Segundo o autor, conforme a quantidade de *clusters* é incrementada, o tempo de processamento e quantidade de memória necessária para a execução cresce exponencialmente, impedindo a utilização de dicionários visuais extensos. Além disso, a qualidade das imagens presentes na GTSRB dificultaram a detecção de pontos de interesse e extração de características pelos algoritmos utilizados (SUPRIYANTO; LUTHFIARTA; ZENIARJA, 2016).

Huang et al. (2017) apresentaram uma abordagem utilizando o algoritmo *Histogram of Oriented Gradient variant* (HOGv) (HUANG et al., 2017) para extrair as características das imagens de placas de trânsito e, em seguida, uma *Extreme Learning Machine*<sup>2</sup> (ELM) (HUANG; ZHU; SIEW, 2006) é aplicada para realizar a classificação. Foram obtidas acurácias máximas de 97,62% e 99,56% em testes realizados nos particionamentos preestabelecidos da BTSD e GTSRB, respectivamente. Segundo o autor, apesar dos altos índices de acurácia encontrados, a abordagem apresentada é altamente dependente do ajuste de parâmetros do modelo e, conseqüentemente, do conjunto de treinamento utilizado (HUANG et al., 2017).

Hu et al. (2017) propuseram uma *framework*, inspirada em mecanismos biológicos (FUSTER, 2001) para acelerar o tempo de classificação de placas de trânsito denominado de *Branch Convolution Neural Network* (BCNN). Essa *framework* funciona como um mecanismo de *branch* (ramificação) e pode ser aplicado em modelos *Convolution Neural Networks* (CNN's) e *Deep Convolution Neural Networks* (DCNN's). Para a validação da *framework* foi utilizada a CNN proposta por Jin, Fu e Zhang (2014), retornando uma acurácia máxima de 99,12% em testes realizados nos particionamentos preestabelecidos da GTSRB.

---

<sup>2</sup> É um algoritmo de aprendizagem baseado em *Single-Hidden-Layer Feedforward Neural Network* (SLFN).

## 2.1 Considerações Finais

Foram apresentados neste capítulo alguns dos trabalhos relacionados com a presente pesquisa, que utilizaram as bases de imagens de placas de trânsito da Alemanha e Bélgica. A [Tabela 1](#) apresenta um quadro-resumo em que é possível observar que diversos tipos de abordagens podem ser utilizadas para a classificação de placas de trânsito.

Tabela 1 – Quadro-resumo dos trabalhos relacionados.

<i>Belgium Traffic Sign Database</i>		
Trabalhos	Metodologia	Acurácia (%)
<a href="#">Mathias et al. (2013)</a>	PHOG + INNLP + SRC	97,83%
<a href="#">Jurišić, Filković e Kalafatić (2015)</a>	OneCNN	98,17%
<a href="#">Qasaimeh, Sagahyroon e Shanableh (2015)</a>	SIFT + BoF + SVM	78,00%
<a href="#">Huang et al. (2017)</a>	HOG <sub>v</sub> + ELM	97,62%
<i>German Traffic Sign Recognition Benchmark</i>		
Trabalhos	Metodologia	Acurácia (%)
<a href="#">Zaklouta e Stanciulescu (2012)</a>	HOG + RF	97,2%
<a href="#">Mathias et al. (2013)</a>	(PHOG + HOG) + INNLP + INNC	98,53%
<a href="#">Yang, Zhou e Deng (2013)</a>	C-HMAX + SVM	98,41%
<a href="#">Jurišić, Filković e Kalafatić (2015)</a>	OneCNN	99,11%
<a href="#">Supriyanto, Luthfiarta e Zeniarja (2016)</a>	SURF + BoF + K-Means	15,21%
<a href="#">Huang et al. (2017)</a>	HOG <sub>v</sub> + ELM	99,56%
<a href="#">Hu et al. (2017)</a>	BCNN	99,12%

Conforme observado, grande parte dos trabalhos utilizam variações de *Artificial Neural Networks* (ANN's). É indiscutível o poder de classificação fornecido por ANN's. Porém, toda essa robustez vem com um custo atrelado, em que geralmente *hardwares* de alto desempenho e extensos tempos de treinamento são necessários para o correto funcionamento dos modelos.

Os trabalhos que utilizaram o modelo *Bag of Features* (BoF) obtiveram taxas de acurácia abaixo do esperado e provavelmente não conseguiram extrair todo o seu potencial. Entretanto, demonstraram que o modelo pode ser aperfeiçoado para fornecer um melhor desempenho e inspiraram o modelo de classificação de placas de trânsito proposto nesta pesquisa.

## 3 Fundamentação Teórica

Neste capítulo são apresentados conceitos introdutórios sobre sinalizações de trânsito e suas características. Além disso, são apresentadas as bases de imagens utilizadas para a validação do método proposto. Por fim, os algoritmos e as métricas de avaliação empregadas nesta pesquisa são detalhadas.

### 3.1 Sinalização de Trânsito

As sinalizações de trânsito podem ser classificadas em sinalização vertical, sinalização horizontal, dispositivos de sinalização auxiliar, sinalização semafórica, sinais sonoros e gestos. Nesta pesquisa, apenas a sinalização vertical será abordada.

Placas de trânsito, semáforos e outros dispositivos de tráfego são utilizados para regular, alertar, orientar ou informar os usuários da via. Eles auxiliam a alcançar um nível aceitável de qualidade do tráfego e aumentam a segurança, com o movimento ordenado e previsível de todo o fluxo, tanto de veículos, como de pedestres (FANG et al., 2003; FANG; CHEN; FUH, 2003).

As sinalizações são projetadas para serem facilmente reconhecidas pelos condutores, principalmente porque suas formas e cores devem ser facilmente distinguíveis dos seus arredores (HOOSE, 1991).

A aparência das sinalizações de trânsito pelo mundo, para a maior parte das nações, segue as orientações da Convenção sobre Trânsito Viário de 1968 (TRAFFIC, 1968). Também conhecida como Convenção de Viena, ela estabelece diretrizes para facilitar o trânsito de pessoas, em regime de cooperação mútua. Mesmo com a assinatura do tratado, do qual fazem parte 101 países, é possível que existam diferenças.

#### 3.1.1 Sinalização Vertical

A sinalização vertical é formada por placas que apresentam linguagem visual/simbólica sobre a(s) via(s) à frente e que podem ser interpretadas pelos condutores. Elas fornecem ao condutor informações que tentam tornar a condução mais segura e conveniente (TRANSPORT, 2016).

As placas podem ser fixadas ao lado da via ou suspensas sobre a pista e transmitem mensagens de perfil permanente. São classificadas em três tipos, de acordo com sua função:

- **Sinalização de Regulamentação:** as placas de regulamentação têm a finalidade de comunicar aos usuários as condições, proibições, restrições ou obrigações no uso

da via. Suas mensagens são imperativas e o desrespeito à elas, constitui infração;

- **Sinalização de Advertência:** as placas de advertência têm por finalidade alertar aos usuários da via sobre condições potencialmente perigosas, indicando sua natureza;
- **Sinalização de Indicação:** as placas de indicação têm por finalidade identificar as vias, os destinos e os locais de interesse; orientar condutores de veículos quanto aos percursos, destinos, distâncias e serviços auxiliares, podendo também educar o usuário. Suas mensagens são informativas ou educativas.

Nesta pesquisa, apenas as placas que representam sinalização de regulamentação e advertência serão utilizadas, pois as placas de sinalização de indicação apresentam informações essencialmente por texto. A [Figura 2](#) apresenta exemplos de placas de regulamentação e advertência.



Figura 2 – Tipos de placas de regulamentação e advertência. Fonte: iSinaliza.com <sup>1</sup>.

<sup>1</sup> Disponível em: <<https://isinaliza.com/>>.

### 3.1.2 Propriedades das Placas de Trânsito

Placas de trânsito são caracterizadas por uma série de características que as tornam reconhecíveis em relação ao ambiente:

- são projetadas, fabricadas e instaladas de acordo com regulamentos rigorosos (VITABILE; SORBELLO, 1998);
- são projetadas em formas 2-D fixas, como círculos, triângulos, retângulos, hexágonos, octógonos, entre outras (ver Figura 3) (PARODI; PICCIOLI, 1995; ITE; WOLSHON; PANDE, 2016);
- as cores são escolhidas para contrastar com o ambiente, o que as torna facilmente reconhecíveis pelos motoristas (JIANG; CHOI, 1998);
- as cores são reguladas pela categoria da sinalização (LALONDE; LI, 1995);
- a informação da placa possui uma cor predominante e o resto da placa possui outra cor, onde as tonalidades das cores devem corresponder a comprimentos de onda específicos no espectro visível (VITABILE; SORBELLO, 1998; VITABILE; GENTILE; SORBELLO, 2002);
- as sinalizações devem estar localizadas em lugares bem definidos em relação à via, de modo que o condutor poderá, mais ou menos, antecipar a localização desses sinais (LALONDE; LI, 1995);
- elas podem conter um pictograma<sup>2</sup>, uma série de caracteres ou ambos (VITABILE; GENTILE; SORBELLO, 2002);



Figura 3 – Exemplos da diversidade de formas de placas de trânsito.

A Figura 4 apresenta placas em situações do mundo real, incluindo placas distorcidas, danificadas, parcialmente ocluídas e agrupadas em um grupo de mais de uma placa.

<sup>2</sup> Um pictograma ou pictógrafa é um símbolo que representa um objeto ou conceito por meio de desenhos figurativos.



(a) Exemplos de placas de trânsito danificadas por fatores físicos.



(b) Exemplos de placas de trânsito ocultas por elementos do meio ambiente.



(c) Exemplos de placas de trânsito desgastadas pela ação do tempo.

Figura 4 – Placas de trânsito modificadas por variações climáticas, temporais e físicas.

## 3.2 Bases de Imagens

Nesta seção são apresentadas três bases de imagens: *Belgium Traffic Sign Dataset* (BTSD) (TIMOFTE; ZIMMERMANN; GOOL, 2009), *German Traffic Sign Recognition Benchmark* (GTSRB) (HOUBEN et al., 2013) e uma base de placas de trânsito gerada no decorrer da pesquisa. Elas foram utilizadas no treinamento e avaliação de desempenho do modelo proposto.

### 3.2.1 *Belgium Traffic Sign Dataset* (BTSD)

A base de imagens de Sinais de Trânsito da Bélgica (BTSD) foi construída em 2009 devido à ausência de repositórios de imagens de placas de trânsito na internet. Ela possui 4.577 imagens para treinamento e 2.520 para teste, totalizando 7.097 imagens. Esse particionamento foi idealizado pelos criadores da base para facilitar a realização de experimentos (TIMOFTE; ZIMMERMANN; GOOL, 2009).

Cada imagem possui dimensões diferentes que variam entre  $26 \times 26$  e  $674 \times 527$  pixels, não são necessariamente quadradas e estão representadas no espaço de cores RGB. A BTSD é composta por 62 classes de placas de trânsito que podem ser encontradas no sistema de tráfego do Reino da Bélgica (TIMOFTE; ZIMMERMANN; GOOL, 2014).

Os arquivos da BTSD também foram codificados no formato *Portable Pixel Map* (PPM) que possui 16 bits por *pixel* para o armazenamento da cor. A Figura 5 apresenta uma amostra de imagens presentes na BTSD.



Figura 5 – Amostra de imagens de placas de trânsito presentes na BTSD. Fonte: Timofte, Zimmermann e Gool (2009)<sup>3</sup>.

Na Figura 6 são exibidas amostras de algumas das classes da BTSD. Percebe-se que as placas possuem diferenças significativas em termos de contraste, brilho, fundo, foco e posicionamento. Além disso, algumas imagens possuem partes de outras placas em suas proximidades.



(a) Classe Pare (*Stop*).



(b) Classe Proibido Virar à Esquerda.



(c) Classe Sentido Obrigatório.



(d) Classe Velocidade Máxima 70 km.

Figura 6 – Amostras de algumas das classes da BTSD. Fonte: Timofte, Zimmermann e Gool (2009)<sup>3</sup>.

<sup>3</sup> Disponível em: <<https://btsd.ethz.ch/shareddata/index.html>>.

### 3.2.2 German Traffic Sign Recognition Benchmark (GTSRB)

A base de imagens de Sinais de Trânsito da Alemanha (GTSRB) foi concebida para a realização de uma competição com o objetivo de avaliar o desempenho de algoritmos de classificação multiclasse. Ela possui 39.209 imagens para treinamento e 12.630 para teste, totalizando 51.839 imagens. Esse particionamento foi realizado para certificar que todos os competidores utilizariam as mesmas imagens nas fases de treinamento e teste, tornando assim, a competição justa.

Cada imagem possui dimensões que variam entre  $25 \times 25$  e  $224 \times 224$  *pixels*, não necessariamente são quadradas e estão representadas no espaço de cores RGB. A GTSRB é composta por 43 classes de placas de trânsito que podem ser encontradas no sistema de tráfego da República Federal da Alemanha (STALLKAMP et al., 2012).

Os arquivos da GTSRB foram codificados no formato *Portable Pixel Map* (PPM) que possui 16 *bits* por *pixel* para o armazenamento da cor. A Figura 7 apresenta uma amostra de imagens presentes na GTSRB.



Figura 7 – Amostra de imagens de placas de trânsito presentes na GTSRB. Fonte: Stallkamp et al. (2012) <sup>4</sup>.

Na Figura 8 são exibidas amostras de imagens de algumas das classes da GTSRB. Percebe-se que as placas possuem diferenças significativas em termos de contraste, brilho, fundo, foco e posicionamento. Além disso, algumas imagens possuem partes de outras placas próximas.

### 3.2.3 Brazilian Traffic Sign Dataset (BraTSD)

Pesquisas realizadas na literatura expuseram a ausência de bases de imagens de placas de trânsito que contenham sinalizações pertencentes ao Sistema Nacional de Trânsito da República Federativa do Brasil. Por esse motivo, surgiu a necessidade da criação de uma base própria com o intuito de avaliar o desempenho do sistema proposto no contexto local da pesquisa.

A base de imagens de sinais de trânsito do Brasil, criada no Laboratório de Processamento Digital de Imagens e Visão Computacional (VipLab) do Departamento de Computação (DC) da Universidade Federal do Piauí (UFPI) e batizada com o acrônimo de BraTSD (*Brazilian Traffic Sign Dataset*), possui 2.122 imagens em sua totalidade e foi construída a partir de fotos capturadas nas vias de Teresina, capital do Piauí.

<sup>4</sup> Disponível em: <<http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>>.



Figura 8 – Amostras de algumas das classes da GTSRB. Fonte: [Stallkamp et al. \(2012\)](#) <sup>4</sup>.

Cada imagem possui dimensões que variam entre  $20 \times 19$  e  $383 \times 554$  *pixels*, não necessariamente são quadradas e estão representadas no espaço de cores RGB. A base é composta por 15 classes de placas de trânsito que podem ser encontradas no sistema de tráfego da República Federativa do Brasil. Os arquivos da base foram codificados no formato *Portable Network Graphics* (PNG) que possui 24 *bits* por *pixel* para o armazenamento da cor. A [Figura 9](#) apresenta uma amostra de imagens presentes na BraTSD.



Figura 9 – Amostra de imagens de placas de trânsito presentes na BraTSD. Fonte: Elaborado pelo Autor <sup>5</sup>.

Na [Figura 10](#) são exibidas amostras de imagens de algumas das classes da BraTSD. Percebe-se que as placas possuem diferenças significativas em termos de contraste, brilho, fundo, foco e posicionamento.

### 3.3 Métodos

O modelo proposto consiste em três etapas principais: extração de características (detecção e descrição dos pontos de interesse), criação do dicionário visual e classificação.

<sup>5</sup> Disponível em: <https://github.com/hugos94/traffic-signs-database>.



(a) Classe Proibido Estacionar.



(b) Classe Sentido Circular na Rotatória.



(c) Classe Siga em Frente.



(d) Classe Velocidade Máxima Permitida 60 km/h.

Figura 10 – Amostras de algumas das classes da BraTSD Fonte: Elaborado pelo Autor <sup>5</sup>.

Durante a primeira etapa, um conjunto de características que representam o conteúdo visual da imagem é detectado e descrito. Esse conjunto fornece entrada para a segunda etapa, em que as características extraídas são agrupadas com o objetivo de construir o dicionário visual. Um histograma é criado e quantizado para cada imagem a partir da comparação de suas características com as palavras do dicionário. Durante a etapa final, o conjunto de histogramas e suas classes são enviadas para o treinamento do classificador.

As Seções 3.3.1 até 3.3.6 apresentarão os algoritmos utilizados nesta pesquisa.

### 3.3.1 Detectores

Em visão computacional, o processo de detecção de pontos de interesse refere-se a métodos que visam computar abstrações de informações em imagens e tomar decisões locais em cada ponto encontrado, quer exista uma característica de determinado tipo nesse ponto ou não. As características resultantes serão subconjuntos do domínio da imagem, muitas vezes sob a forma de pontos isolados, curvas contínuas ou regiões conectadas.

Há um volume substancial de literatura que está concentrada em detectar pontos de interesse, incluindo *Corner Detectors* (HARRIS; STEPHENS, 1988), *Blob Detectors* (LINDBERG, 1998), *Dense Sampling* (TUYTELAARS, 2010) e *Random Sampling* (CHOI; HAN, 2014).

Detectores de cantos (*corner detectors*) surgiram de métodos que realizavam detecção de bordas (*edge detection*) e depois analisavam as bordas detectadas na procura

de mudanças rápidas de direção (*corners*). Esses detectores foram então aperfeiçoados de modo que a detecção de borda explícita não fosse mais necessária, por exemplo, procurando por níveis elevados de curvatura no gradiente da imagem. Observou-se então que os chamados cantos também estavam sendo detectados em partes da imagem que não eram cantos no sentido tradicional (por exemplo, um pequeno ponto brilhante em um fundo escuro pode ser detectado). Esses pontos são frequentemente conhecidos como pontos de interesse.

Detectores de *Binary Large Objects (blobs)* fornecem uma descrição complementar das estruturas presentes nas imagens em termos de região, em oposição aos detectores de cantos que fornecem apenas pontos. No entanto, detectores de *blobs* podem conter localizações preferidas (máximos locais ou centros de gravidade). Isso significa que muitos detectores de *blobs* também podem detectar pontos de interesse. Além disso, esses detectores geralmente detectam áreas em imagens que são muito suaves para serem detectadas por detectores de canto.

A amostragem densa (*dense sampling*) é baseada na extração de pontos de interesse por meio da aplicação de uma grade densa (*dense grid*) de pontos na imagem. A amostragem aleatória (*random sampling*) é baseada na seleção randômica de coordenadas de pontos que geralmente seguem uma distribuição uniforme.

Nas duas abordagens, qualquer quantidade de pontos de interesse pode ser alcançada ao ajustar o espaçamento entre eles.

Nesta pesquisa, os detectores SIFT (LOWE, 1999) e FAST (ROSTEN; DRUMMOND, 2005) foram utilizados. O detector SIFT foi selecionado devido a sua conhecida robustez e sua vasta utilização em problemas de classificação de placas de trânsito (KUS; GOKMEN; ETANER-UYAR, 2008; TAKAKI; FUJIYOSHI, 2009; LI et al., 2012; MALIK; SIDDIQI, 2014). O detector de cantos FAST, adequado para aplicações em tempo real, foi escolhido por sua comprovada eficiência computacional e velocidade de execução. Além disso, o FAST retorna um maior conjunto de pontos de interesse com maior entropia entre as suas detecções do que o SIFT. Não foram encontrados trabalhos na literatura que utilizassem o detector FAST em sistemas de classificação de placas de trânsito.

### 3.3.1.1 *Scale-Invariant Feature Transform (SIFT)*

O detector SIFT, proposto por Lowe (1999) para a detecção de pontos de interesse, transforma uma imagem em uma grande coleção de vetores de características. Esses vetores são invariantes para as transformações geométricas de translação, escala e rotação, parcialmente invariantes para mudanças de iluminação e robustos para distorções geométricas locais. Essas características compartilham propriedades semelhantes com neurônios do córtex visual primário que codificam formas básicas, cor e movimento para a detecção de objetos na visão de primatas (SERRE et al., 2007).

Pontos de interesse são definidos como máximos e mínimos do resultado da Diferença de Funções Gaussianas (DoG) aplicadas no espaço de escala para uma série de imagens suavizadas e reamostradas. Pontos candidatos que possuem baixo contraste ou que estejam ao longo de uma borda são descartados. Orientações dominantes são atribuídas aos pontos de interesse localizados. Esses passos garantem que os pontos de interesse encontrados sejam mais estáveis para correspondência e reconhecimento.

A Figura 11 exibe o resultado da aplicação do detector de pontos de interesse SIFT em imagens de placas de trânsito. Cada círculo colorido representa um ponto de interesse identificado e que será utilizado pelos descritores de características.



Figura 11 – Exemplos de pontos de interesses detectados pelo SIFT em imagens de placas de trânsito. Fonte: Elaborado pelo Autor.

### 3.3.1.2 Features from Accelerated Segment Test (FAST)

A literatura está repleta de detectores de características robustos. Entretanto, ao olhar pelo ponto de vista de aplicações em tempo real, eles não são rápidos o suficiente. Na tentativa de solucionar esse problema, o detector de pontos de interesse FAST foi proposto em Rosten e Drummond (2005), revisado em Rosten e Drummond (2006) e aperfeiçoado em Rosten, Porter e Drummond (2010).

Apesar de ser diversas vezes mais rápido que outros detectores de cantos existentes, ele não é robusto na presença de altos níveis de ruídos e é dependente de um limiar.

O algoritmo funciona da seguinte forma: um *pixel* é definido como um canto se (em um círculo ao redor do *pixel*)  $N$  ou mais *pixels* contíguos são todos significativamente mais claros ou escuros que o *pixel* central. O conjunto de pontos de interesse detectados são utilizados para treinar o algoritmo ID3 (QUINLAN, 1986). A árvore gerada pelo ID3 é utilizada no restante das imagens e é responsável por tornar a execução mais rápida Rosten, Porter e Drummond (2010).

A Figura 12 exibe o resultado da aplicação do detector de pontos de interesse FAST em imagens de placas de trânsito. Há uma diferença perceptível entre as quantidades de pontos de interesse encontrados em relação a Figura 11.



Figura 12 – Exemplos de pontos de interesses detectados pelo FAST em imagens de placas de trânsito. Fonte: Elaborado pelo Autor.

### 3.3.2 Descritores

Após a detecção dos pontos de interesse, um descritor de características é aplicado na área ao redor de cada ponto para produzir um vetor de características. Identificar o tamanho e o tipo de descritor apropriado para um problema de classificação é uma tarefa desafiadora que envolve uma série de experimentos.

A abordagem mais fácil é simplesmente usar os valores de intensidade de *pixels*, escalados para o tamanho da região, ou uma representação do seu espaço de imagem. As representações de *pixels* normalizados, no entanto, apresentaram pior performance do que a maioria das representações (FEI-FEI; PERONA, 2005; NOWAK; JURIE; TRIGGS, 2006) e foram amplamente abandonadas pela comunidade de pesquisa.

Nesta pesquisa, os algoritmos SIFT (LOWE, 2004) e BRIEF (CALONDER et al., 2010) foram utilizados na descrição de características. Essas escolhas foram realizadas devido a vasta utilização dos descritores SIFT (KUS; GOKMEN; ETANER-UYAR, 2008; TAKAKI; FUJIYOSHI, 2009; LI et al., 2012; MALIK; SIDDIQI, 2014; RATNASARI; IRAWAN; SETIANINGSIH, 2017) e BRIEF (ZAVADIL; TUMA; SANTOS, 2012; ERHAN et al., 2013; KIM et al., 2014; JANG; KIM, 2016) na literatura, com o objetivo de descrever características de imagens de placas de trânsito.

#### 3.3.2.1 *Scale-Invariant Feature Transform* (SIFT)

O método proposto por Lowe (1999), além de detectar pontos de interesse em imagens, pode descrever os pontos encontrados. Primeiro, uma vizinhança de  $16 \times 16$  *pixels* é identificada ao redor do ponto de interesse. Então 16 sub-blocos de tamanho  $4 \times 4$  são divididos. Para cada sub-bloco, é criado um histograma de orientação de 8 elementos. Esses histogramas são calculados a partir de valores de magnitude e orientação de amostras em uma região de  $16 \times 16$  *pixels* ao redor do ponto de interesse, de modo que cada histograma contenha amostras de uma sub-região  $4 \times 4$  da região da vizinhança original. As magnitudes são ponderadas por uma função gaussiana com desvio padrão ( $\sigma$ ) igual a metade da largura da janela do descritor. O descritor se torna então um vetor de

todos os valores desses histogramas. Uma vez que existem 16 histogramas cada um com 8 elementos, o vetor possui 128 elementos. Além disso, várias medidas são tomadas para alcançar a robustez contra mudanças de iluminação, rotação e etc. A Figura 13 apresenta uma representação do gradiente da imagem e da descrição do ponto de interesse para o descritor SIFT.

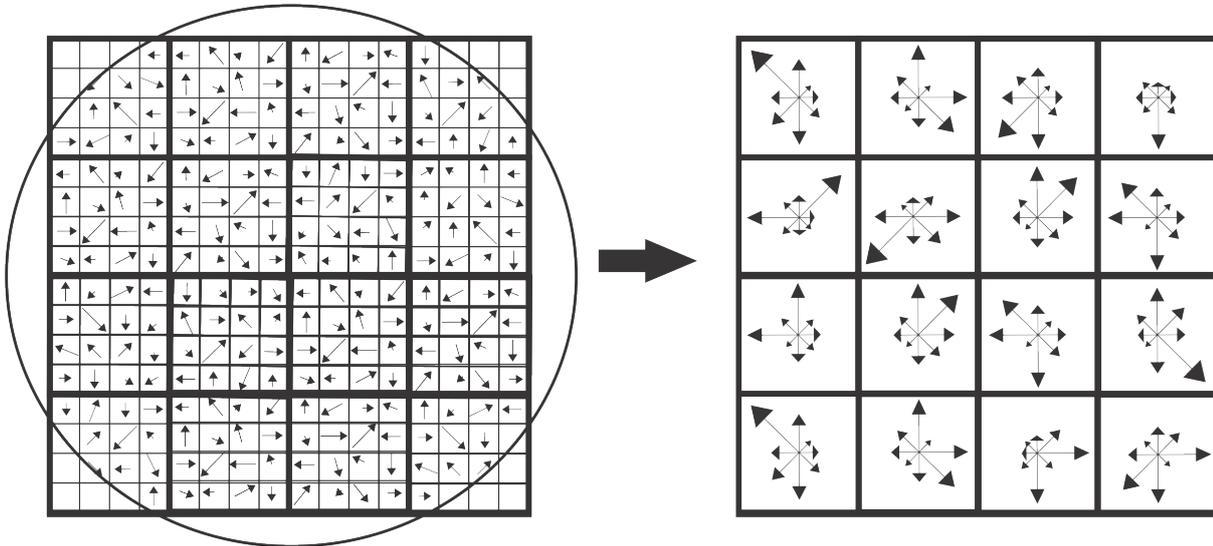


Figura 13 – Representação do gradiente da imagem e da descrição do ponto de interesse para o descritor de características SIFT. Fonte: Elaborado pelo Autor.

Embora a dimensão do descritor pareça alta, descritores com dimensões mais baixas do que 128 não funcionam tão bem (LOWE, 2004) e o custo computacional permanece baixo devido a utilização do método *Best Bin First* (BBF) utilizado para encontrar o vizinho mais próximo.

### 3.3.2.2 Binary Robust Independent Elementary Features (BRIEF)

O descritor SIFT implementado nesta pesquisa utiliza 128 dimensões. Como foram utilizados números em ponto flutuante, cada característica precisa de, no mínimo, 512 *bytes* para ser armazenada. Logo, a criação de vetores para milhares de características requer muita memória, o que não é viável para aplicações com recursos restritos. Quanto maior a memória, mais tempo leva para descrever características.

Além disso, as 128 dimensões dos descritores geralmente não são necessárias para a descrição das características. Essas dimensões podem ser reduzidas usando vários métodos, como *Principal Component Analysis* (PCA) (F.R.S., 1901), *Linear Discriminant Analysis* (LDA) (HOTELLING, 1933) e etc. Até mesmo outros métodos, como *hashing* usando *Locality Sensitive Hasing* (LSH) (INDYK; MOTWANI, 1998; GIONIS; INDYK; MOTWANI, 1999), são usados para converter os descritores como SIFT e SURF de números em pontos flutuantes para cadeias binárias (*binary strings*). Essas cadeias binárias são usadas para combinar características utilizando a distância de *Hamming* (HAMMING,

1950). Isso proporciona uma melhora na velocidade, pois após a distância de *Hamming* ser encontrada, basta aplicar a porta lógica XOR ( $\oplus$ ) e realizar a contagem dos bits resultantes. Esses dois processos são muito rápidos em CPU's modernas que possuem instruções SSE. Entretanto, utilizando essa abordagem, ainda é preciso encontrar os descritores primeiro, para só depois aplicar esse método, o que não resolve o problema do uso de muita memória.

O descritor BRIEF, proposto por Calonder et al. (2010) e aperfeiçoado em Calonder et al. (2012), fornece um atalho para localizar as cadeias binárias diretamente sem a necessidade de encontrar os descritores, transpondo esse problema.

Primeiro, um *patch* da imagem é suavizado e um conjunto de pares locais  $n_d$  ( $x$ ,  $y$ ) é selecionado. Cinco métodos principais são considerados pelo autor para encontrar os vetores  $X$  e  $Y$ . São eles:

- (a)  $X$  e  $Y$  são amostras uniformemente randômicas;
- (b)  $X$  e  $Y$  são amostras randômicas geradas utilizando a distribuição Gaussiana, onde locais mais próximos ao centro do *patch* são os preferidos;
- (c)  $X$  e  $Y$  são amostras randômicas geradas utilizando uma distribuição Gaussiana em que o primeiro  $X$  é gerado com um desvio padrão de  $0,04 \times S^2$  e então os  $Y_i$ 's são gerados usando uma distribuição Gaussiana – cada  $Y_i$  é gerado com média  $X_i$  e desvio padrão de  $0,01 \times S^2$ ;
- (d)  $X$  e  $Y$  são amostras randômicas a partir de locais discretos em uma grade polar;
- (e) Para cada  $i$ ,  $X_i$  é  $(0, 0)$  e  $Y_i$  toma todos os valores possíveis em uma grade polar.

A Figura 14 ilustra as cinco estratégias consideradas em Calonder et al. (2010) para encontrar os vetores  $X$  e  $Y$ .

Após uma das cinco estratégias ser aplicada, algumas comparações de intensidade de *pixel* são realizadas nos pares de locais encontrados. Por exemplo, considerando  $I$  como intensidade e que  $p$  e  $q$  são os primeiros pares locais, se  $I(p) < I(q)$  então o resultado é 1, senão é 0. Isso é aplicado para todos os pares locais  $n_d$ , gerando uma cadeia binária de  $n_d$  dimensões.

Esse  $n_d$  pode ser de dimensão 128, 256 ou 512, com 16, 32 e 64 *bytes*, respectivamente. Após a cadeia binária ser obtida, a distância de *Hamming* pode ser aplicada para combinar os descritores.

Em suma, o BRIEF é um descritor muito rápido, além de fornecer uma boa descrição de características, a menos que exista uma grande rotação no plano.

<sup>6</sup> Disponível em: <[https://www.cs.ubc.ca/~lowe/525/papers/calonder\\_eccv10.pdf](https://www.cs.ubc.ca/~lowe/525/papers/calonder_eccv10.pdf)>.

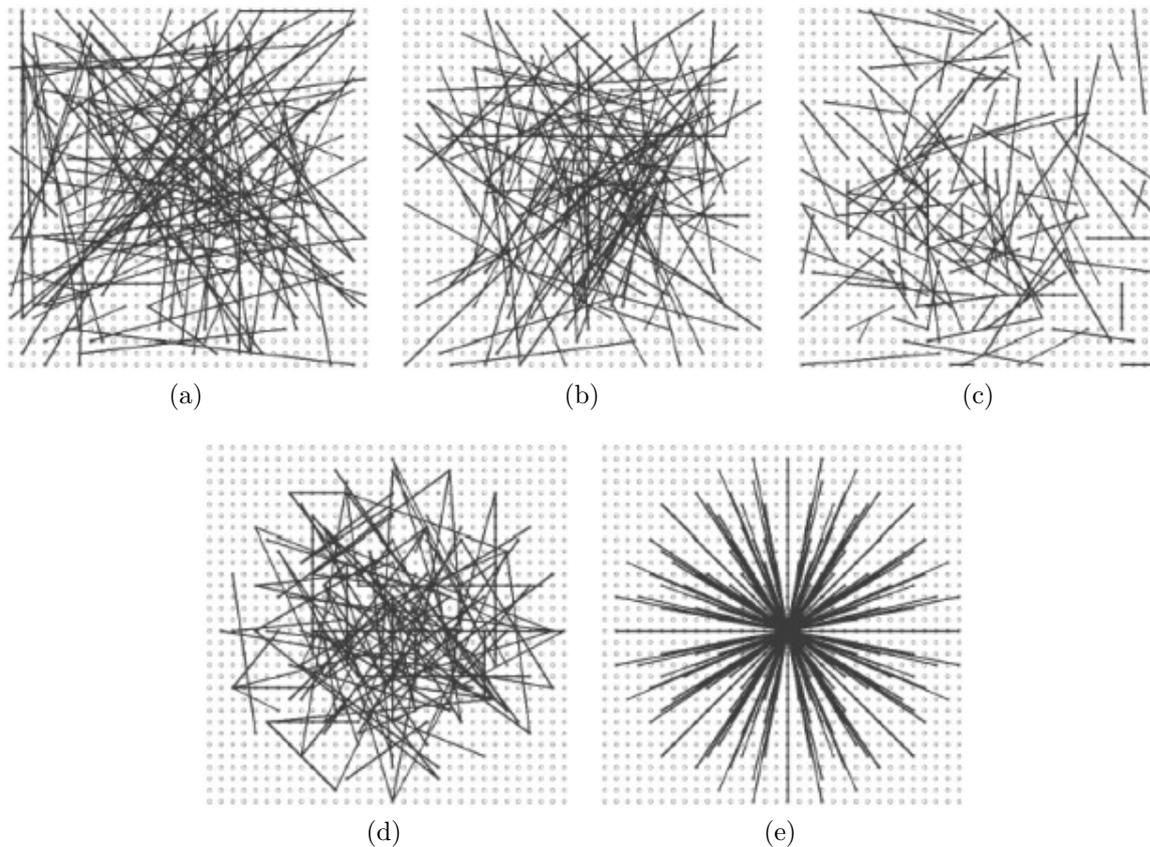


Figura 14 – Diferentes abordagens para encontrar os vetores  $X$  e  $Y$  do descritor BRIEF. Fonte: Calonder et al. (2010) <sup>6</sup>.

### 3.3.3 *Bag of Features* (BoF)

Para processar as descrições das características encontradas nas imagens de placas de trânsito, o modelo *Bag of Features* (BoF) foi escolhido devido à sua capacidade de lidar com alta diversidade visual e ausência de arranjos espaciais intraclasses típicos (CSURKA et al., 2004).

O modelo BoF foi introduzido na área de visão computacional como um descritor global de características para problemas de classificação complexos (FEI-FEI; PERONA, 2005). O BoF foi derivado do modelo *Bag of Words* (BoW) (SALTON; WONG; YANG, 1975; SALTON, 1992), que é uma representação simplificada utilizada no processamento de linguagem natural e recuperação de informações (JOACHIMS, 1998). Nesse modelo, um texto (como uma frase ou um documento) é representado como uma bolsa (*bag* ou *multiset*) de suas palavras, em que apenas a frequência com que elas aparecem é considerada. Da mesma forma, no contexto de análise de imagens, uma imagem é representada por um histograma de palavras visuais que são definidas por características que representam padrões comuns.

O conceito do BoF se adapta ao problema de classificação de placas de trânsito, uma vez que um determinado tipo de placa geralmente é percebida como um conjunto de

diferentes elementos visuais misturados com proporções específicas. O modelo BoF possui duas etapas básicas: criação do dicionário visual e quantização dos descritores.

A Figura 15 ilustra uma representação das etapas básicas do modelo BoF. A parte esquerda da Figura 15 demonstra as imagens de placas de trânsito com suas características detectadas e descritas, como apresentado nas Seções 3.3.1 e 3.3.2. Na parte central da Figura 15 os vetores de características gerados no processo anterior são submetidos ao algoritmo de agrupamento. O resultado final do agrupamento será o dicionário de palavras visuais. Esse processo será apresentado na Seção 3.3.4. Para finalizar, na parte direita da Figura 15 temos o processo de quantização dos descritores, em que os vetores de características são quantizados com o auxílio do dicionário visual criado no passo anterior. Esse processo gera um histograma de características para cada imagem e será descrito na Seção 3.3.5.

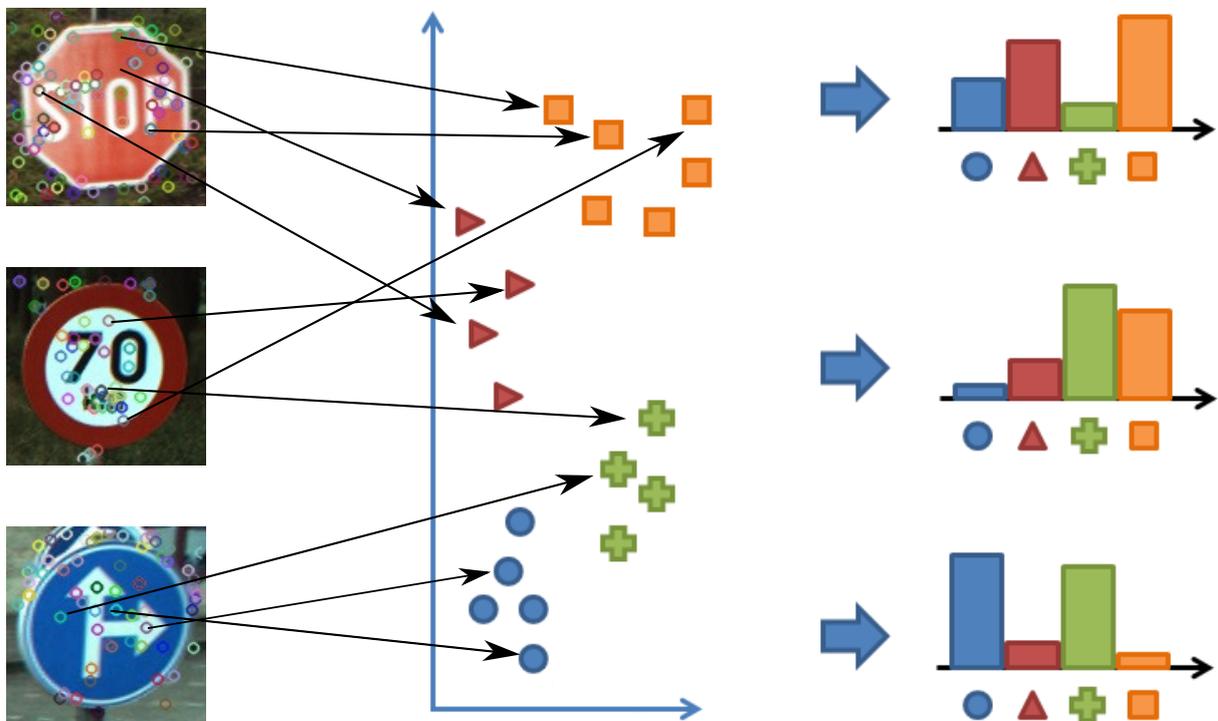


Figura 15 – Etapas do modelo proposto. A parte esquerda representa o processo de detecção e extração de características das placas de trânsito. Na parte central, as características são submetidas ao algoritmo de agrupamento para a criação do dicionário de palavras visuais. Por fim, a parte direita representa o processo de quantização dos descritores, onde um histograma de características é gerado. Fonte: Elaborado pelo Autor.

### 3.3.4 Dicionários Visuais

Após as características das imagens serem descritas, as descrições mais representativas precisam ser identificadas para a construção do dicionário visual no modelo BoF. Para esse fim, os vetores de características gerados são submetidos a um processo de

agrupamento dos vetores em um número predefinido de grupos (*clusters*). Ao final do procedimento, os centros dos grupos criados constituirão as palavras visuais do dicionário visual.

Existem muitos algoritmos de agrupamento na literatura e este trabalho não tenta enumerá-los. Dado qualquer método de agrupamento, existirão pontos que serão igualmente próximos de mais de um centroide. Esses pontos se situam próximos a um limite de *Voronoi* entre grupos (*clusters*) e criam ambiguidade ao atribuir características a termos. Em algoritmos de agrupamento, a escolha das posições iniciais dos centroides afeta o vocabulário resultante. Ao lidar com vocabulários relativamente pequenos, pode-se executar o agrupamento várias vezes e selecionar o vocabulário de melhor desempenho durante a etapa de validação. Isso se torna mais difícil para conjuntos de dados muito grandes.

Os algoritmos de agrupamento *K-Means* (MACQUEEN, 1967) e *Mini Batch K-Means* (SCULLEY, 2010) foram utilizados na criação do dicionário visual. Essas escolhas foram realizadas devido a vasta utilização dos algoritmos de agrupamento *K-Means* (SIVIC; ZISSERMAN, 2003; LAZEBNIK; SCHMID; PONCE, 2006; JIANG; NGO; YANG, 2007; VELVIZHY; PAVITHRA; KANNAN, 2014; YUAN; MENG, 2014; PASSALIS; TEFAS, 2016) e *Mini Batch K-Means* (VIDYA; PRIYANKKA; KUMAR, 2014; FEIZOLLAH et al., 2014; ALSEIARI; AUNG, 2015; FITRIYANI; MURFI, 2016; HSU; LIN, 2017; PENG; LEUNG; HUANG, 2018; HSU; LIN, 2018) na literatura para a etapa de construção do dicionário de palavras visuais do modelo *Bag of Features*.

#### 3.3.4.1 *K-Means*

Proposto em MacQueen (1967), o *K-Means* é um dos algoritmos de aprendizado não supervisionado mais simples que resolvem o bem conhecido problema de agrupamento. O agrupamento é uma tarefa que tem por objetivo segmentar uma base de dados em grupos de objetos, baseando-se na similaridade ou dissimilaridade entre os mesmos. O procedimento segue uma maneira simples e fácil de classificar um determinado conjunto de dados por meio de  $k$  grupos (*clusters*) fixados a priori.

A ideia principal é definir  $k$  centroides, um para cada grupo (*cluster*). Diferentes localizações de centroides geram resultados distintos. A escolha mais utilizada é colocá-los o máximo possível longe um do outro. O próximo passo é pegar cada ponto pertencente a um determinado conjunto de dados e associá-lo ao centroide mais próximo. Quando nenhum ponto está pendente, o primeiro passo é concluído e o agrupamento inicial é finalizado. Nesse ponto,  $k$  novos centroides são recalculados como baricentros dos grupos resultantes do passo anterior. Depois de ter esses  $k$  novos centroides, uma nova ligação deve ser feita entre os mesmos pontos do conjunto de dados e o novo centroide mais próximo. Um *loop* foi gerado. Como resultado desse *loop*, os  $k$  centroides mudam sua localização

passo a passo até que não sejam feitas mais alterações. Em outras palavras, os centroides não se movem mais e o agrupamento está terminado.

A Figura 16 apresenta a inicialização dos centroides e a execução de duas iterações do algoritmo *K-Means* para um problema de agrupamento com dois grupos.

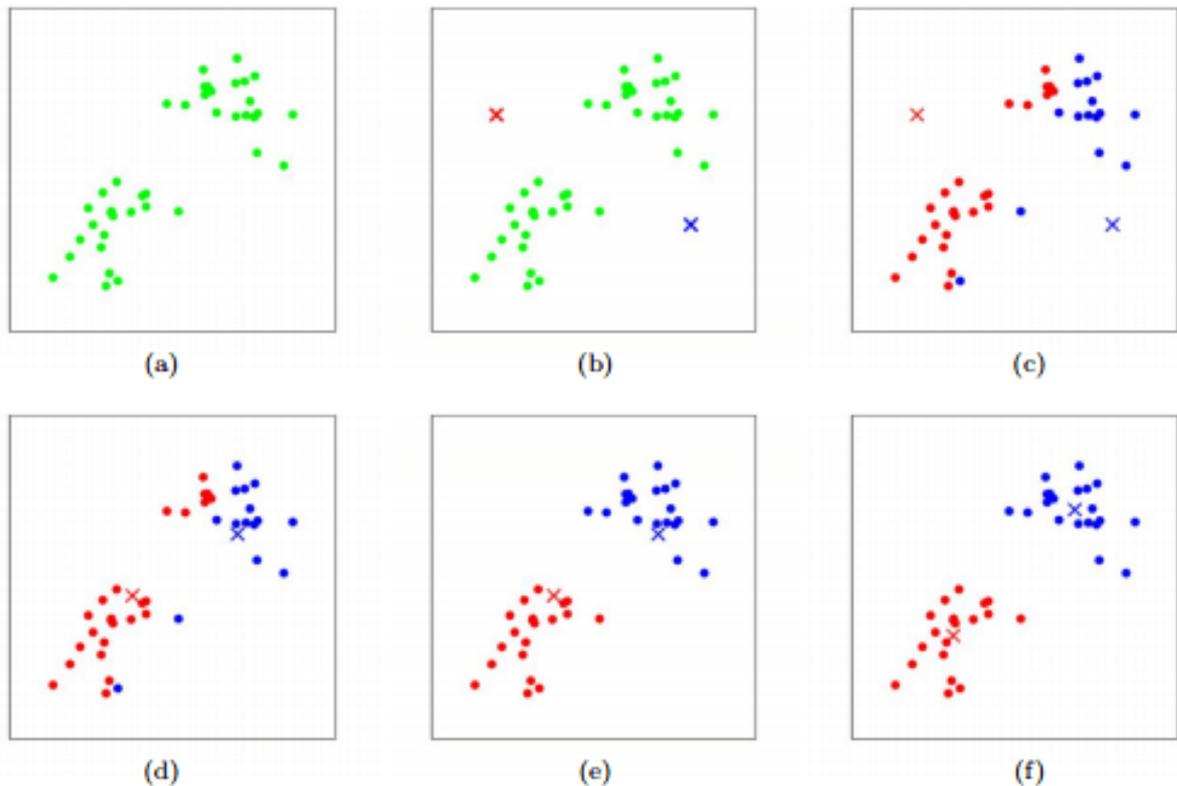


Figura 16 – Algoritmo *K-Means*. Os exemplos de treinamento e os centroides são representados como pontos e cruces, respectivamente. (a) Conjunto de dados original. (b) Inicialização aleatória dos  $k$  centroides. (c-f) Ilustração da execução de duas iterações do *K-Means*. Em cada iteração, atribuímos cada exemplo de treinamento ao centroide mais próximo (mostrado ao “pintar” cada ponto da mesma cor que o centroide do grupo ao qual foi atribuído). Então movemos cada centroide para a posição média dos pontos atribuídos a ele. Fonte: Michael I. Jordan <sup>7</sup>.

Embora o procedimento sempre termine, obedecendo ao critério de parada, o algoritmo de agrupamento *K-Means* não necessariamente encontrará a configuração ótima, correspondente ao mínimo objetivo global da função. O *K-Means* também é significativamente sensível aos centroides dos grupos randomicamente selecionados e a realização de repetidas execuções pode amenizar esse efeito.

<sup>7</sup> Disponível em: <<http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>>.

### 3.3.4.2 Mini Batch K-Means

O *Mini Batch K-Means*, proposto em [Sculley \(2010\)](#), é uma variante do algoritmo *K-Means* que utiliza mini lotes (*mini-batches*) para reduzir o tempo computacional, enquanto ainda tenta otimizar a mesma função objetivo. Os mini lotes são subconjuntos dos dados de entrada, randomicamente amostrados em cada iteração de treinamento. Esses mini lotes reduzem drasticamente a quantidade de computação necessária para convergir para uma solução local.

De maneira similar ao *K-Means* padrão, o algoritmo realiza iterações entre dois passos principais. No primeiro passo,  $b$  amostras são sorteadas aleatoriamente a partir do conjunto de dados, para formar um mini lote. Estes são então atribuídos ao centroide mais próximo. Na segunda etapa, os centroides são atualizados. Em contraste com *K-Means*, isso é feito em uma base por amostra. Para cada amostra no mini lote, o centroide atribuído é atualizado, tomando a média de *streaming* da amostra e todas as amostras anteriores atribuídas a esse centroide. Isso tem o efeito de diminuir a taxa de mudança de um centroide ao longo do tempo. Esses passos são executados até a convergência ou um número predeterminado de iterações ser atingido.

O *Mini Batch K-Means* converge mais rápido que o *K-Means*, mas a qualidade dos resultados é reduzida. Na prática, essa diferença de qualidade pode ser bem pequena, como pode ser visto na [Figura 17](#) e nos trabalhos de [Béjar \(2013\)](#) e [Feizollah et al. \(2014\)](#).

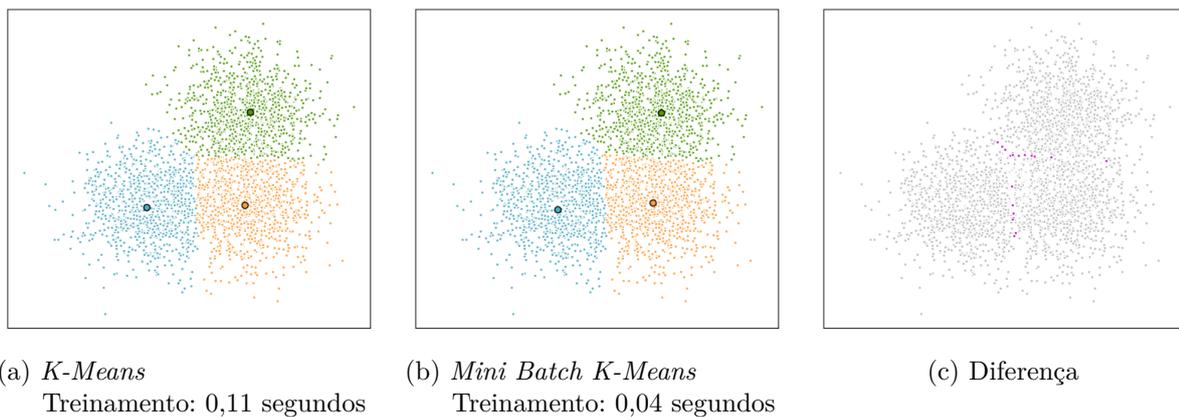


Figura 17 – Comparação entre os resultados obtidos para o agrupamento de um conjunto de dados com  $k = 3$ , utilizando os algoritmos *K-Means* e *Mini Batch K-Means*. O *K-Means* se mostrou mais lento do que o *Mini Batch K-Means*. Fonte: Elaborado pelo Autor.

### 3.3.5 Quantização dos Descritores

A quantização dos descritores é o procedimento de atribuir um vetor de características à palavra visual mais próxima de um vocabulário visual predefinido. Uma vez que o

dicionário visual é aprendido, cada descritor de uma imagem é quantificado e o histograma de ocorrências de palavras visuais serve como uma descrição global da imagem.

Então, os valores de histograma geralmente são dimensionados para  $[0 \ 1]$  e são utilizados para treinar ou testar o classificador. A eficiência dessa parte do modelo *Bag of Features* é crucial, uma vez que afeta os tempos de processamento. A complexidade da quantificação dos descritores depende principalmente das dimensões do descritor e da quantidade de palavras visuais.

### 3.3.6 Classificadores

A tarefa de classificar placas de trânsito é um problema de classificação multiclasse bastante conhecido e abordado por diversas aplicações práticas. Como apresentado nas Seções 3.2.1, 3.2.2 e 3.2.3, as bases de dados utilizadas possuem 62, 43 e 15 classes de placas de trânsito, respectivamente.

O modelo *Support Vector Machine* (SVM), introduzido por Cortes e Vapnik (1995), foi selecionado para a classificação devido a sua vasta utilização na literatura e robustez comprovada na composição do modelo BoF (HIBA; HAMID; OMAR, 2016; WANG et al., 2016; NGUYEN; YOSHITAKA, 2016; JIN; OMAR; JAWARD, 2016; ALTINOZ; ALTINOZ, 2017).

#### 3.3.6.1 *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) são métodos de aprendizado supervisionado que analisam dados e reconhecem padrões, e são utilizados para classificação e análise de regressão (CORTES; VAPNIK, 1995).

O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico.

Dado um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, o algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra. Um modelo SVM é uma representação de exemplos como pontos no espaço, mapeados de maneira que os exemplos de cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível. Os novos exemplos são então mapeados no mesmo espaço e preditos como pertencentes a uma categoria baseados em qual lado do espaço eles são colocados.

Como o modelo padrão do SVM apenas realiza classificação binária, o modelo SVM multiclasse será utilizado. Para obtenção desse modelo multiclasse, duas estratégias podem ser utilizadas: *One-versus-One* (O-v-O) e *One-versus-All* (O-v-A).

A estratégia **One-versus-One** (Um-contra-Um) treina  $\frac{N \times (N-1)}{2}$  classificadores binários para um problema multiclasse de  $N$  classes. Cada classificador recebe amostras de um par de classes do conjunto de treinamento e deve aprender a distinguí-las. Na etapa de predição, um esquema de votação é aplicado: todos os  $\frac{N \times (N-1)}{2}$  classificadores binários básicos são aplicados a uma amostra de teste e a classe que obtiver a maior quantidade de previsões é a classe predita pelos classificadores combinados (BISHOP, 2006, p. 339). A estratégia falha em seu objetivo quando os classificadores binários básicos empatam na votação (BISHOP, 2006, p. 183). A Figura 18 apresenta uma representação do SVM multiclasse com abordagem *One-versus-One* (O-v-O) para um problema com três classes distintas.

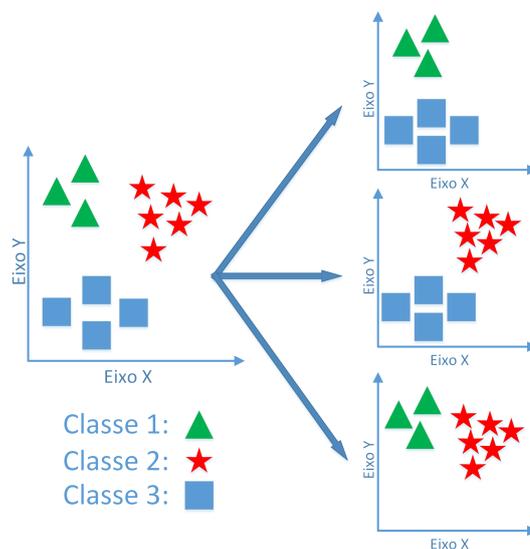


Figura 18 – SVM multiclasse com abordagem *One-versus-One* (O-v-O). Fonte: Elaborado pelo Autor.

A estratégia **One-versus-All** (Um-contra-Todos) envolve treinar  $N$  classificadores binários, em que  $N$  é a quantidade de classes do problema. Durante o treinamento, cada classificador binário recebe as amostras da classe representada como positivas e todas as outras como negativas. Essa estratégia exige que os classificadores binários básicos produzam uma nota de confiança de valor real para sua decisão, em vez de apenas o rótulo de classe. Utilizar somente rótulos discretos pode levar a ambiguidades, em que várias classes são preditas para uma única amostra (BISHOP, 2006, p. 182).

Embora essa estratégia seja popular, é uma heurística que sofre de vários problemas. Em primeiro lugar, a escala dos valores de confiança pode diferir entre os classificadores binários. Em segundo lugar, mesmo que a distribuição de classes seja equilibrada no conjunto de treinamento, os classificadores binários básicos observam distribuições desequilibradas, porque normalmente o conjunto de negativos que eles observam é muito maior do que o conjunto de positivos (BISHOP, 2006, p. 338). A Figura 19 apresenta uma representação do SVM multiclasse com abordagem *One-versus-All* (O-v-A) para um problema com três

classes distintas.

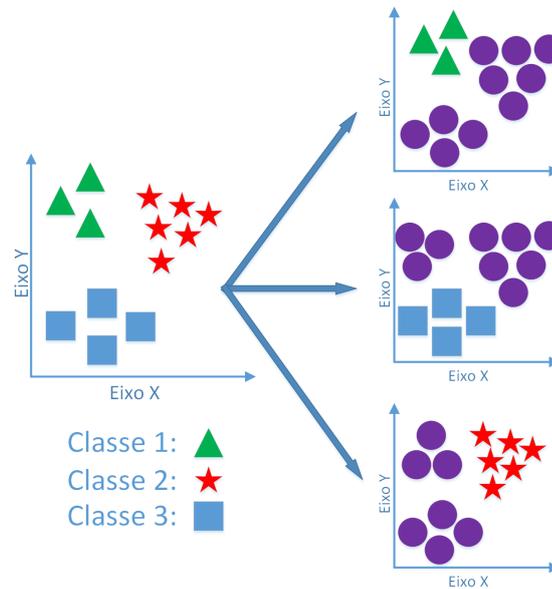


Figura 19 – SVM multiclasse com abordagem *One-versus-All* (O-v-A). Fonte: Elaborado pelo Autor.

### 3.4 Métricas de Avaliação dos Resultados

A maioria dos critérios de análise dos resultados de um classificador provém de uma matriz de confusão, que indica a quantidade de elementos classificados corretamente e incorretamente para cada um dos grupos existentes (FAWCETT, 2006).

A matriz de confusão é criada tendo como referência as predições realizadas pelo classificador e a real classe dos elementos. A matriz é gerada com base em quatro valores:

- Verdadeiro Positivo (VP), são os objetos da classe X que foram classificados corretamente como pertencentes a classe X;
- Verdadeiro Negativo (VN), são os objetos que não pertencem a classe X e foram classificados corretamente como não pertencentes a classe X;
- Falso Positivo (FP), são os objetos que não pertencem a classe X e foram classificados incorretamente como pertencentes a classe X;
- Falso Negativo (FN), são os objetos da classe X que foram classificados incorretamente como não pertencentes a classe X.

A partir desses valores, algumas taxas estatísticas podem ser calculadas para avaliar o desempenho da capacidade de classificação do modelo proposto. As taxas de Acurácia ( $A$ ) e índice *Kappa* ( $K$ ), métricas conhecidas na literatura, foram escolhidas para

a análise deste trabalho e são calculadas conforme definições dadas pelas Equações 3.1 e 3.2, respectivamente.

**Acurácia** é a porcentagem de casos corretamente classificados em um conjunto de teste (POWERS, 2007). É calculada a partir da Equação 3.1.

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN}. \quad (3.1)$$

**Índice Kappa** é um coeficiente de concordância para escalas nominais que mede o relacionamento entre a concordância, além da casualidade, e a discordância esperada (ROSENFELD; FITZPATRICK-LINS, 1986). É calculado a partir da Equação 3.2.

$$K = \frac{\theta_1 - \theta_2}{1 - \theta_2}, \text{ em que} \quad (3.2)$$

$$\theta_1 = \frac{VP + VN}{VP + VN + FP + FN}, \quad (3.3)$$

$$\theta_2 = \frac{[(VP + FN)(VP + FP)] + [(VN + FN)(VN + FP)]}{(VP + VN + FP + FN)^2}. \quad (3.4)$$

O índice *Kappa* é utilizado como uma medida apropriada da exatidão por representar inteiramente a matriz de confusão. O nível de precisão do índice *Kappa* utilizado para a avaliação dos resultados foi classificado conforme a Tabela 2, de acordo com o estabelecido em Landis e Koch (1977).

Tabela 2 – Nível de precisão da classificação, de acordo com o índice *Kappa*.

Índice <i>Kappa</i> (K)	Qualidade
$K \leq 0,2$	Ruim
$0,2 < K \leq 0,4$	Razoável
$0,4 < K \leq 0,6$	Bom
$0,6 < K < 0,8$	Muito Bom
$K \geq 0,8$	Excelente

### 3.5 Método de Avaliação dos Classificadores

Aprender os parâmetros de uma função de predição e testá-la nos mesmos dados é um erro metodológico: um classificador que apenas repete os rótulos das amostras que acabou de ver teria uma pontuação perfeita, mas não conseguiria prever nada de útil em dados ainda não vistos.

Essa situação é chamada de *overfitting*. Para evitá-lo, é prática comum, ao realizar um experimento de aprendizado de máquina (supervisionado), reter parte dos dados disponíveis como um conjunto de testes.

Ao avaliar diferentes configurações para estimadores, ainda há um risco de *overfitting* no conjunto de testes porque os parâmetros podem ser ajustados até que o estimador tenha um desempenho ideal. Dessa forma, o conhecimento sobre o conjunto de testes pode ser repassado para o modelo e as métricas de avaliação não relatam mais o desempenho da generalização.

Para resolver esse problema, ainda outra parte do conjunto de dados pode ser realizada como um chamado “conjunto de validação”: o treinamento prossegue no conjunto de treinamento, após o qual a avaliação é feita no conjunto de validação e quando o experimento parece ser bem-sucedido, a avaliação final pode ser feita no conjunto de testes.

No entanto, particionando os dados disponíveis em três conjuntos, o número de amostras que podem ser usadas para aprender o modelo reduzem drasticamente e os resultados podem depender de uma escolha aleatória específica para o par de conjuntos (treinamento, validação).

Uma solução para esse problema é um procedimento chamado validação cruzada (*cross-validation*). Um conjunto de testes ainda deve ser oferecido para avaliação final, mas o conjunto de validação não é mais necessário ao fazer a validação cruzada. Na abordagem básica, chamada *K-Fold Cross-Validation*, o conjunto de treinamento é dividido em  $k$  subconjuntos menores. O seguinte procedimento é seguido para cada um dos  $k$  subconjuntos:

- Um modelo é treinado usando  $k - 1$  dos subconjuntos como dados de treinamento;
- O modelo resultante é validado na parte restante dos dados (ou seja, é usado como um conjunto de testes para calcular uma medida de desempenho, como precisão).

A medida de desempenho relatada pelo *K-Fold* é então a média dos valores computados na repetição. Essa abordagem pode ser computacionalmente cara, mas não desperdiça muitos dados, o que é uma grande vantagem em problemas como a inferência inversa, em que o número de amostras é muito pequeno.

Além disso, alguns problemas de classificação podem exibir um grande desequilíbrio na distribuição das classes: por exemplo, pode haver várias vezes mais amostras negativas do que positivas. Nesses casos, recomenda-se usar amostragem estratificada para garantir que as frequências de classe relativas sejam preservadas aproximadamente em cada subconjunto de treinamento e validação.

O método escolhido para avaliação neste trabalho foi o *Stratified K-Fold Cross-*

*Validation.* Para definir um valor  $k$  que pudesse ser utilizado em todas as combinações e gerasse uma correta estratificação, as quantidades de imagens presentes nas classes das bases foram contadas e o valor  $k = 5$  foi definido.

## 3.6 Considerações Finais

Este capítulo introduziu informações sobre as propriedades das placas de trânsito. Adicionalmente, foram apresentadas as bases de imagens utilizadas, suas características e limitações. Além disso, foram descritos os algoritmos utilizados nesta pesquisa, como detectores de pontos de interesse, descritores de características, algoritmos de agrupamento e métodos de classificação. Por fim, foram apresentadas as métricas utilizadas para avaliar o desempenho do modelo proposto.

Considerando a importância de classificar placas de trânsito, foram realizados diversos experimentos com diferentes combinações de algoritmos. A escolha dos algoritmos foi determinada a partir de três requisitos principais: os mais utilizados, custo de processamento e trabalhos mais recentes. As principais análises feitas em relação à classificação foram a acurácia e o índice *Kappa*.

## 4 Modelo Proposto

Este trabalho propõe um modelo para a classificação de imagens de placas de trânsito. O contexto do problema envolve padrões previamente conhecidos (placas de trânsito) e a necessidade de um modelo capaz de aprender tais padrões. Sendo assim, a aplicação de uma técnica que utiliza aprendizagem supervisionada é possível. Diante do exposto, a utilização de um modelo baseado no *Bag of Features* (BoF) para a tarefa de classificação de placas de trânsito foi investigado.

Com base no que foi discutido na Seção 3.3, pode-se desenvolver uma visão geral do modelo proposto, contendo três etapas principais: uma para a detecção e descrição de características (extração de características), outra para a criação do dicionário visual (*bag of features*) e por fim a classificação das imagens de placas de trânsito (classificação). A Figura 20 apresenta a estrutura do modelo proposto neste trabalho com suas etapas delimitadas.

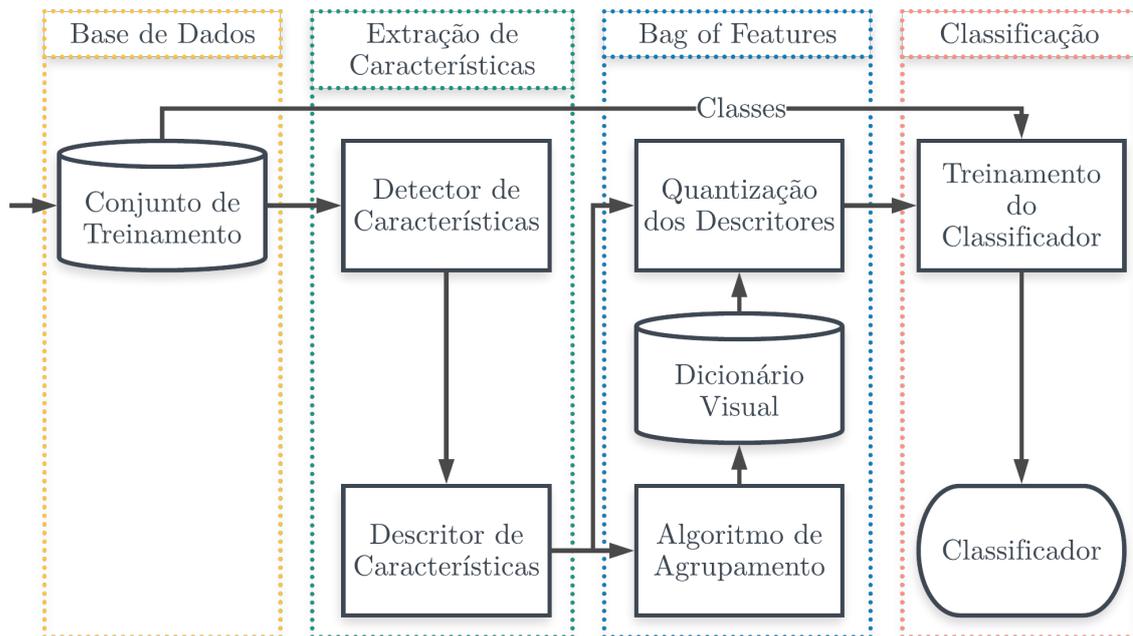


Figura 20 – Etapas do modelo proposto. Fonte: Elaborado pelo Autor.

### 4.1 Extração de Características

A etapa de extração de características do modelo proposto é dividida, conforme apresentado na Figura 20, em dois processos: detecção e descrição de características. As Seções 4.1.1 e 4.1.2 descrevem detalhadamente os dois processos, respectivamente.

### 4.1.1 Detecção de Características

No início do processo de detecção de características, as imagens do conjunto de treinamento são inseridas no modelo proposto. Logo após, um detector de características é aplicado nessas imagens. Neste trabalho, os detectores SIFT (LOWE, 1999) e FAST (ROSTEN; DRUMMOND, 2005) foram utilizados, conforme descrito na Seção 3.3.1.

O detector SIFT, apresentado na Seção 3.3.1.1, é bastante robusto, porém retorna baixa quantidade de pontos de interesse quando é aplicado em imagens pequenas ou com baixa resolução. Além disso, o SIFT não é considerado um detector de tempo real, tornando lento o processo de detectar pontos de interesse. Por esses motivos, o detector FAST, apresentado na Seção 3.3.1.2, foi escolhido como uma alternativa de detecção de pontos de interesse em tempo real. Além disso, o detector FAST geralmente retorna mais pontos de interesse do que o detector SIFT. Porém, apesar da velocidade de detecção, o FAST não é robusto na presença de altos níveis de ruídos e é dependente de um limiar.

Ambos detectores apresentaram dificuldades para realizar a detecção das características em algumas das imagens utilizadas. Em alguns casos, o detector retornou poucas ou nenhuma característica. Nesses casos, não foi possível utilizar essas imagens nos testes realizados.

### 4.1.2 Descrição de Características

Após a detecção, um descritor foi aplicado nos conjuntos de características detectados para cada imagem. Conforme apresentado na Seção 3.3.2, os descritores SIFT (LOWE, 1999) e BRIEF (CALONDER et al., 2010) foram utilizados.

O descritor SIFT, apresentado na Seção 3.3.2.1, é bastante robusto, porém demanda alto poder computacional para descrever as características detectadas. Entretanto, o descritor SIFT não é considerado descritor de tempo real, ou seja, não possui baixo tempo de resposta na tarefa de descrever imagens. Por esses motivos, o descritor BRIEF, apresentado na Seção 3.3.2.2, foi escolhido como uma alternativa de descrição em tempo real. Porém, como o BRIEF não é invariante a escala, um pouco da robustez é perdida em troca da velocidade e baixa necessidade de poder computacional.

O resultado da aplicação do descritor é um vetor de características para cada imagem.

## 4.2 *Bag of Features*

A etapa do *Bag of Features* é separada, conforme apresentado na Figura 20, em três processos: agrupamento, criação do dicionário visual e a quantização dos descritores. As Seções 4.2.1, 4.2.2 e 4.2.3 descrevem os três processos, respectivamente.

### 4.2.1 Agrupamento de Características

Após a geração dos vetores de características, as descrições mais representativas precisam ser identificadas e utilizadas na construção do dicionário visual. Para esta tarefa, um algoritmo de agrupamento (clusterização) é aplicado. Como apresentado na Seção 3.3.4, os algoritmos *K-Means* (MACQUEEN, 1967) e *Mini Batch K-Means* (SCULLEY, 2010) foram utilizados nesta pesquisa.

O algoritmo *K-Means* é apresentado na literatura como uma popular escolha para tarefas de agrupamento em tempo real. Porém, para grandes bases de imagens esse algoritmo se torna lento e demanda alto poder computacional. Visando contornar esses problemas, o algoritmo *Mini Batch K-Means* foi utilizado. Ele promete, sem perder a robustez no agrupamento, maior velocidade com menor necessidade de alto poder computacional. Além disso, oferece uma redução no consumo de armazenamento.

### 4.2.2 Criação do Dicionário Visual

Durante o agrupamento, cada grupo (*cluster*) criado representa uma “palavra visual” e o conjunto de todas as palavras visuais formam o dicionário visual. O dicionário visual é uma das estruturas mais importante do modelo BoF e a sua robustez deve ser assegurada para prover bom desempenho ao classificador.

As características de uma imagem podem não estar contidas ou podem não ser representadas se o dicionário for muito pequeno. Além disso, alguns ruídos e características de áreas não-alvo podem estar facilmente contidas no dicionário se ele for muito grande. O tamanho apropriado do dicionário deve ser selecionado de acordo com as circunstâncias específicas de cada base de imagens e essa escolha pode afetar o desempenho do classificador.

### 4.2.3 Quantização dos Descritores

Conforme visto na Figura 20, a quantização dos descritores recebe dados do processo de descrição de características e do dicionário visual. Uma vez que o dicionário visual é criado, cada vetor de características é quantificado e o histograma de ocorrências de palavras visuais serve como uma descrição global da imagem. Então, os valores de histograma geralmente são dimensionados para  $[0 \ 1]$  e são utilizados no treinamento do classificador.

A eficiência desta parte do modelo também é crucial, uma vez que afeta o tempo de treinamento. A complexidade da quantificação dos descritores depende principalmente das dimensões dos vetores gerados pelo descritor e da quantidade de palavras no dicionário visual.

## 4.3 Classificação

A etapa de classificação, conforme apresentada na [Figura 20](#), é composta apenas pelo processo de treinamento do classificador. Durante o processo, os descritores quantizados gerados na descrição e as classes presentes na base de imagens são utilizadas para treinar o classificador.

No início da fase de treinamento do estágio de classificação, os descritores quantizados gerados na descrição e os rótulos das imagens são utilizados para treinar o classificador. Como apresentado na [Seção 3.3.6](#), as estratégias multiclasse *One-vs-One* e *One-vs-All* do classificador SVM ([CORTES; VAPNIK, 1995](#)) foram utilizadas neste trabalho.

Conforme informado na [Seção 3.3.6](#), a abordagem *One-vs-One* cria um classificador individual para cada par de classes. Como as bases de imagens utilizadas neste trabalho possuem quantidades extensas de classes, uma grande quantidade de classificadores individuais são criados e estouros de memória foram frequentes na utilização dessa abordagem. Já na *One-vs-All*, apenas um classificador individual precisa ser criado para cada classe e isso reduz a quantidade de recurso computacional necessário para a sua execução. Além disso, os tempos de processamento demonstraram-se menores em relação a outra abordagem.

## 5 Resultados e Discussões

Neste capítulo são apresentados os resultados encontrados nos testes realizados com o objetivo de analisar o desempenho do modelo proposto para a classificação de placas de trânsito.

Todos os algoritmos foram implementados utilizando a linguagem de programação *Python* (ROSSUM, 1995). O *framework Scikit-Learn* (PEDREGOSA et al., 2011) foi utilizado para auxiliar no treinamento do classificador e no cálculo das métricas de avaliação de desempenho. A máquina utilizada para a realização dos testes possui 4 GB de memória RAM, processador *AMD Phenom(tm) II X4 B97* (2010) com 4 núcleos executando a 3,20 GHz cada e a distribuição *Linux Manjaro* (MÜLLER; DEVELOPERS, 2011).

Conforme apresentado na Seção 3.2, foram utilizadas três bases de imagens de placas de trânsito para a realização dos testes:

- BraTSD - 15 classes e 2.122 imagens;
- BTSD - 62 classes e 7.097 imagens;
- GTSRB - 43 classes e 51.839 imagens.

Devido ao baixo poder computacional disponível para a realização dos testes, por conter uma menor quantidade de classes e imagens, a BraTSD foi utilizada para analisar o desempenho das combinações de algoritmos propostas. A partir dos resultados obtidos nessa base, as melhores combinações de algoritmos foram aplicadas na BTSD e GTSRB.

Foi utilizado, em todos testes, o método estatístico *Stratified K-Fold Cross-Validation*, com o  $k$  igual a 5. A Acurácia (A) e o índice *Kappa* (K) foram as métricas utilizadas neste trabalho para a avaliação de desempenho dos algoritmos, por serem apropriadas para a análise de problemas de classificação multiclasse, conforme apresentado na Seção 3.4.

### 5.1 *Brazilian Traffic Sign Dataset* (BraTSD)

Neste experimento, todos os métodos apresentados na Seção 3.3 foram combinados e aplicados na BraTSD. O objetivo é encontrar a combinação de algoritmos que obtenha o melhor desempenho na classificação de imagens de placas de trânsito com o melhor aproveitamento do hardware fornecido.

A Tabela 3 apresenta os resultados obtidos na classificação das imagens da BraTSD com a utilização do detector e descritor SIFT, do algoritmo de agrupamento *K-Means* e

dos classificadores SVM *One-versus-One* e SVM *One-versus-All*. Por a BraTSD conter poucas imagens, foi possível variar a quantidade de *clusters* entre 500 e 10.000, permitindo assim, visualizar a influência do dicionário visual no resultado final.

Tabela 3 – Resultado da classificação com a utilização do descritor e detector SIFT, algoritmo de agrupamento *K-Means* e os classificadores SVM *One-versus-One* e SVM *One-versus-All* nas imagens da BraTSD.

<i>Clusters</i>	Descritor SIFT + Detector SIFT			
	<i>K-Means</i>			
	SVM <i>One-versus-One</i>		SVM <i>One-versus-All</i>	
	Acurácia (A%)	<i>Kappa</i> (K)	Acurácia (A%)	<i>Kappa</i> (K)
500	80,76%	0,7923	79,59%	0,7798
1.000	83,87%	0,8258	86,00%	0,8490
2.000	<b>84,19%</b>	<b>0,8291</b>	89,44%	0,8861
3.000	82,78%	0,8137	91,47%	0,9079
4.000	82,97%	0,8157	92,36%	0,9176
5.000	81,79%	0,8030	92,74%	0,9217
6.000	82,08%	0,8060	93,44%	0,9293
7.000	81,37%	0,7984	94,06%	0,9359
8.000	80,34%	0,7871	94,15%	0,9369
9.000	79,16%	0,7743	93,91%	0,7871
10.000	78,97%	0,7721	<b>94,38%</b>	<b>0,9394</b>

Na utilização do classificador SVM *One-versus-One*, o índice *Kappa* dos resultados varia entre 0,79 e 0,83, sendo considerados índices “muito bom” e “excelente”. A maior acurácia obtida para essa combinação foi de 84,19% com 2.000 *clusters*. Já com o classificador SVM *One-versus-All*, o índice *Kappa* dos resultados varia entre 0,77 e 0,94, sendo considerados índices “muito bom” e “excelente”. A maior acurácia obtida para essa combinação foi de 94,38% com 10.000 *clusters*.

Os resultados produzidos pela combinação contendo o classificador SVM *One-versus-All* foram superiores em relação aos resultados produzidos pelo classificador SVM *One-versus-One*. Por ser menos complexo, o classificador SVM *One-versus-All* consumiu menos recursos computacionais e executou em uma menor quantidade de tempo. Além disso, a variação no tamanho do dicionário visual teve reflexo no desempenho do modelo proposto para esse classificador.

O algoritmo de agrupamento *Mini Batch K-Means* foi utilizado, no lugar da sua versão padrão, com a mesma bateria de testes. A Tabela 4 apresenta os resultados obtidos na classificação das imagens da BraTSD com a utilização do detector e descritor SIFT, dos algoritmo de agrupamento *Mini Batch K-Means* e dos classificadores SVM *One-versus-One* e SVM *One-versus-All*.

Na utilização do classificador SVM *One-versus-One*, o índice *Kappa* dos resultados varia entre 0,75 e 0,79, sendo considerados índices “muito bons”. A maior acurácia obtida

para essa combinação foi de 80,66% com 5.000 *clusters*. Já com o classificador SVM *One-versus-All*, o índice *Kappa* dos resultados varia entre 0,74 e 0,94, sendo considerados índices “muito bom” e “excelente”. A maior acurácia obtida para essa combinação foi de 93,74% com 10.000 *clusters*.

O classificador SVM *One-versus-All* produziu novamente resultados superiores. Devido a maior robustez do algoritmo *K-Means* em relação ao *Mini Batch K-Means*, os resultados obtidos foram pouco inferiores se comparados ao experimento anterior, apresentado na Tabela 3. Porém, os experimentos realizados com o *Mini Batch K-Means* executaram em uma fração do tempo requerido pelo *K-Means*.

Tabela 4 – Resultado da classificação com a utilização do descritor e detector SIFT, algoritmo de agrupamento *Mini Batch K-Means* e os classificadores SVM *One-versus-One* e SVM *One-versus-All* nas imagens da BraTSD.

<i>Clusters</i>	Descritor SIFT + Detector SIFT			
	<i>Mini Batch K-Means</i>			
	SVM <i>One-versus-One</i>		SVM <i>One-versus-All</i>	
	Acurácia (A%)	<i>Kappa</i> (K)	Acurácia (A%)	<i>Kappa</i> (K)
500	77,70%	0,7592	76,81%	0,7498
1.000	78,31%	0,7656	81,18%	0,7970
2.000	79,43%	0,7775	86,28%	0,8520
3.000	78,50%	0,7674	87,99%	0,8705
4.000	79,71%	0,7805	90,62%	0,8988
5.000	<b>80,66%</b>	<b>0,7908</b>	91,08%	0,9038
6.000	79,62%	0,7795	90,99%	0,9029
7.000	79,25%	0,7753	92,26%	0,9166
8.000	80,10%	0,7846	93,16%	0,9262
9.000	78,64%	0,7686	93,16%	0,9263
10.000	77,41%	0,7552	<b>93,74%</b>	<b>0,9325</b>

O detector FAST e o descritor BRIEF foram utilizados, no lugar do descritor e detector SIFT, com a mesma bateria de testes. A Tabela 5 apresenta os resultados obtidos na classificação das imagens da BraTSD com a utilização do detector FAST, do descritor BRIEF, do algoritmo de agrupamento *Mini Batch K-Means* e dos classificadores SVM *One-versus-One* e SVM *One-versus-All*.

Na utilização do classificador SVM *One-versus-One*, o índice *Kappa* dos resultados varia entre 0,67 e 0,86, sendo considerados índices “bom” e “muito bom”. A maior acurácia obtida para essa combinação foi de 86,95% com 500 *clusters*. Já com o classificador SVM *One-versus-All*, o índice *Kappa* dos resultados varia entre 0,83 e 0,94, sendo considerados índices “excelentes”. A maior acurácia obtida para essa combinação foi de 94,43% com 8.000 *clusters*.

Novamente, o classificador SVM *One-versus-All* produziu resultados superiores. Os experimentos realizados com o detector FAST e o descritor BRIEF, por serem algoritmos de

tempo real, executaram em uma fração do tempo requerido pelo detector e descritor SIFT. Além disso, de modo geral, as taxas de acurácia e dos índices *Kappa* demonstraram-se superiores, quando comparadas com os resultados da Tabela 4.

Tabela 5 – Resultado da classificação com a utilização do detector FAST e descritor BRIEF, algoritmo de agrupamento *K-Means* e os classificadores SVM *One-versus-One* e SVM *One-versus-All* nas imagens da BraTSD.

<i>Clusters</i>	Detector FAST + Descritor BRIEF			
	<i>Mini Batch K-Means</i>			
	SVM <i>One-versus-One</i>		SVM <i>One-versus-All</i>	
	Acurácia (A%)	<i>Kappa</i> (K)	Acurácia (A%)	<i>Kappa</i> (K)
500	<b>86,95%</b>	<b>0,8588</b>	84,73%	0,8351
1.000	82,39%	0,8090	85,54%	0,8438
2.000	79,73%	0,7800	87,09%	0,8606
3.000	77,01%	0,7502	90,65%	0,8991
4.000	75,63%	0,7352	91,89%	0,9125
5.000	76,24%	0,7418	92,26%	0,9165
6.000	73,03%	0,7067	93,35%	0,9283
7.000	73,32%	0,7100	93,82%	0,9333
8.000	72,95%	0,7059	<b>94,43%</b>	<b>0,9399</b>
9.000	71,83%	0,6938	94,05%	0,9359
10.000	69,69%	0,6703	94,34%	0,9389

Ao comparar as maiores acurácias presentes nas Tabelas 3 (94,38%), 4 (93,74%) e 5 (94,43%), se observa que todas ficaram em torno de 94%. Entretanto, em termos de tempos de execução e consumo de poder computacional, os resultados são extremamente distintos.

Os algoritmos SIFT, *K-Means* e SVM *One-versus-One* necessitam de maior poder computacional e requerem um maior tempo para a execução do modelo. Já os algoritmos FAST, BRIEF, *Mini Batch K-Means* e SVM *One-versus-All* necessitam de menor poder computacional e requerem uma fração do tempo dos algoritmos anteriores para a execução do modelo. Para bases de pequeno porte a diferença não interfere nos tempos de execução, porém, em bases de médio e grande porte, a diferença é expressiva.

O gráfico da Figura 21 apresenta os resultados das Tabelas 3, 4 e 5 agrupados. De modo geral, os detectores e descritores retornam boas características e o desempenho final não foi afetado por esses algoritmos. Vale ressaltar, que os detectores e descritores considerados de tempo real consumiram menos recursos e possibilitaram a execução de testes em uma menor quantidade de tempo.

Os algoritmos de agrupamento *K-Means* e *Mini Batch K-Means* produziram dicionários visuais semelhantes em todas as situações. Entretanto, conforme apresentado na Seção 3.3.4, o *Mini Batch K-Means* reduziu o consumo de recursos em troca de resultados inferiores do que o *K-Means*.

Nos testes realizados, os classificadores representaram o fator de maior interferência no desempenho do modelo. O classificador SVM *One-versus-One*, considerado uma versão mais complexa e robusta, gerou resultados inferiores com um consumo alto de recursos, demonstrando não ser uma boa escolha. Além disso, o aumento no tamanho do dicionário visual comprometeu a sua performance. Em contrapartida, o classificador SVM *One-versus-All* gerou melhores resultados com um consumo reduzido de recursos. Além disso, o aumento no tamanho do dicionário visual possuiu clara influência nos resultados gerados.

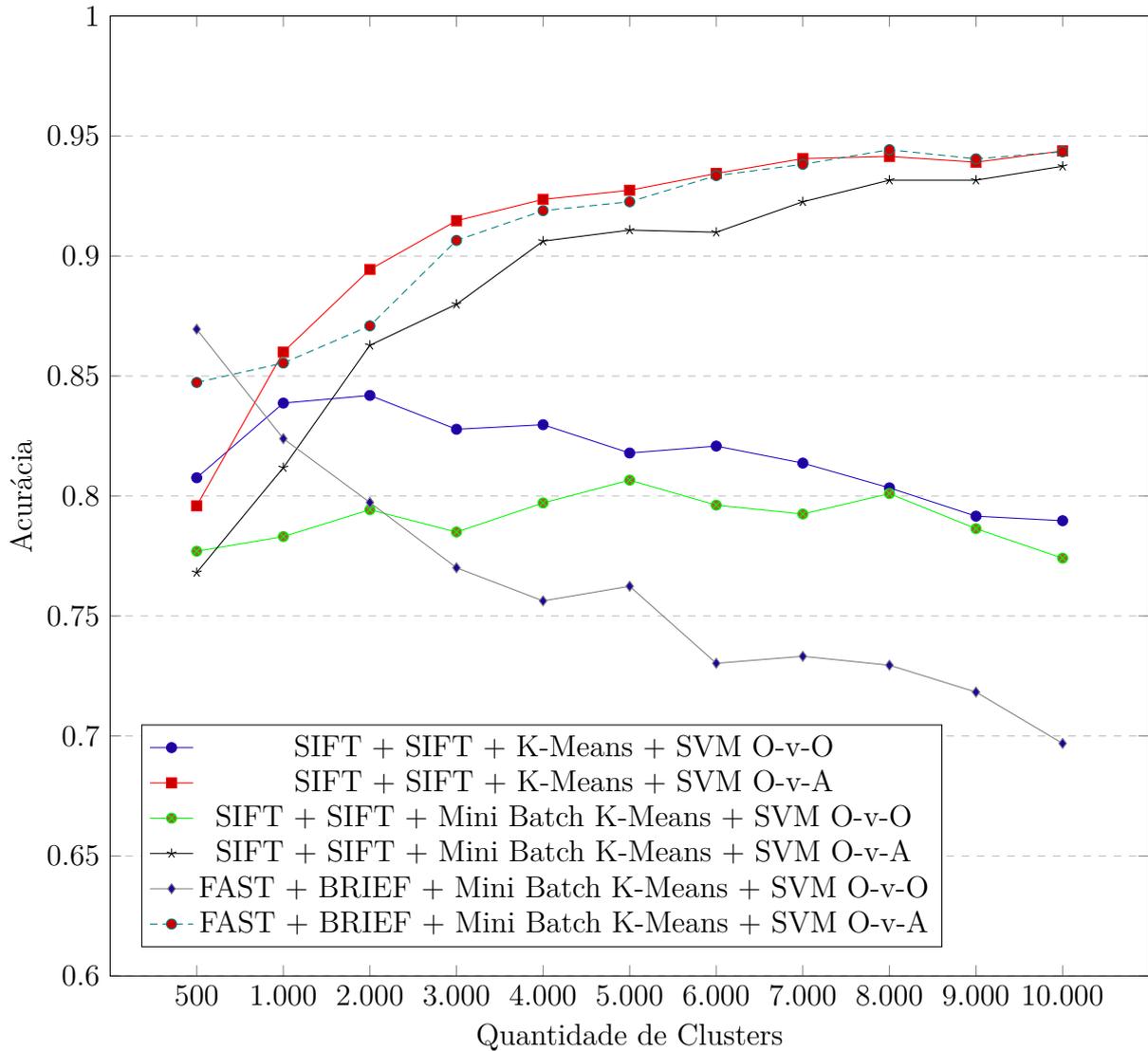


Figura 21 – Comparação do desempenho do modelo proposto na BraTSD.

Como informado anteriormente, as duas melhores combinações foram escolhidas para serem executadas nas bases BTSD e GTSRB. As combinações selecionadas foram:

- Descritor SIFT + Detector SIFT + *K-Means* + SVM *One-versus-All*;
- Descritor FAST + Detector BRIEF + *Mini Batch K-Means* + SVM *One-versus-All*.

## 5.2 Belgium Traffic Sign Dataset (BTSD)

A base de imagens de Sinais de Trânsito da Bélgica (BTSD) contem 62 classes, possuindo 4.577 imagens no particionamento de treinamento e 2.520 imagens no particionamento de teste, totalizando 7.097 imagens. Esses particionamentos foram idealizados pelos criadores da base para facilitar a realização de experimentos (TIMOFTE; ZIMMERMANN; GOOL, 2009).

Porém, para o *Stratified K-Fold Cross-Validation* ser utilizado e a padronização entre os testes ser mantida, os particionamentos de treinamento e teste foram agrupados em uma única partição. Além disso, o *K-Fold* com  $k$  igual a 5 foi aplicado em todos os testes realizados. Como a BTSD possui aproximadamente três vezes mais imagens do que a BraTSD, a quantidade de *clusters* precisou ser reduzida para permitir a realização dos testes no *hardware* disponível em tempo hábil. Foram utilizados tamanhos entre 100 e 600 *clusters*.

A Tabela 6 apresenta os resultados obtidos na classificação da BTSD. Foram utilizadas as duas melhores combinações da Seção 5.1. Na utilização do detector e descritor SIFT com o *K-Means*, o índice *Kappa* dos resultados varia entre 0,52 e 0,64, sendo considerados índices “bom” e “muito bom”. A maior acurácia obtida para essa combinação foi de 65,39% com 600 *clusters*. Já com o detector FAST e descritor BRIEF com o *Mini Batch K-Means*, o índice *Kappa* dos resultados varia entre 0,92 e 0,96, sendo considerados índices “excelentes”. A maior acurácia obtida para essa combinação foi de 96,67% com 500 *clusters*.

Tabela 6 – Desempenho do modelo proposto na BTSD.

Clusters	SIFT + SIFT		FAST + BRIEF	
	<i>K-Means</i>		<i>Mini Batch K-Means</i>	
	SVM One-versus-All			
	Acurácia (A%)	<i>Kappa</i> (K)	Acurácia (A%)	<i>Kappa</i> (K)
100	54,89%	0,5212	94,93%	0,9249
200	58,55%	0,5616	95,95%	0,9401
300	60,38%	0,5802	96,09%	0,9423
400	63,48%	0,6129	95,88%	0,9392
500	64,59%	0,6243	<b>96,67%</b>	<b>0,9509</b>
600	<b>65,39%</b>	<b>0,6327</b>	95,88%	0,9398

O gráfico da Figura 22 exhibe a comparação entre resultados apresentados na Tabela 6 onde é perceptível que ambas combinações são afetadas pelo tamanho do dicionário visual. A segunda combinação, composta em sua maioria por algoritmos de tempo real, obteve resultados superiores em relação ao desempenho para todas as quantidades de *clusters* utilizadas. Além disso, a quantidade de poder computacional necessária para a sua execução é mínima em relação a primeira combinação.

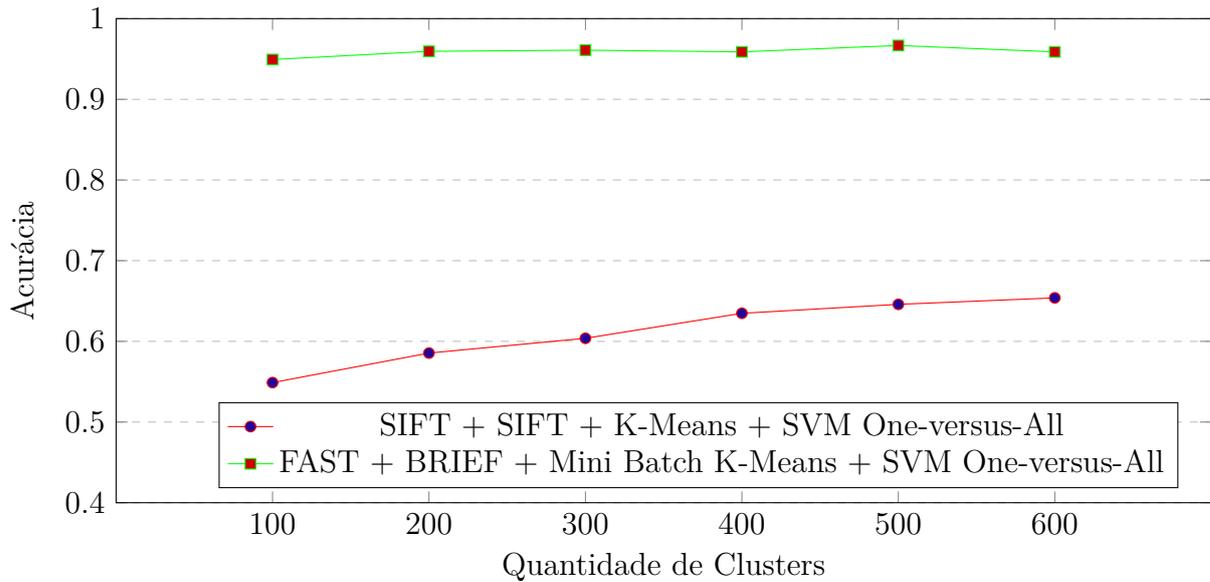


Figura 22 – Comparação do desempenho geral do modelo proposto na BTSD.

### 5.3 German Traffic Sign Recognition Benchmark (GTSRB)

A base de imagens de Sinais de Trânsito da Alemanha (GTSRB) contém 43 classes, possuindo 39.209 imagens no particionamento de treinamento e 12.630 imagens no particionamento de teste, totalizando 51.839 imagens. Esses particionamentos foram idealizados pelos criadores da base para facilitar a realização de experimentos (STALLKAMP et al., 2012).

Porém, para o *Stratified K-Fold Cross Validation* ser utilizado e a padronização entre os testes ser mantida, os particionamentos de treinamento e teste foram agrupados em uma única partição. Além disso, o *K-Fold* com  $k$  igual a 5 foi aplicado em todos os testes realizados. Como a GTSRB possui aproximadamente vinte e quatro vezes mais imagens do que a BraTSD, a quantidade de *clusters* precisou ser reduzida para permitir a realização dos testes no *hardware* disponível em tempo hábil. Foram utilizados tamanhos entre 100 e 600 *clusters*.

A Tabela 7 apresenta os resultados obtidos na classificação da GTSRB. Foram utilizadas as duas melhores combinações da Seção 5.1. Na utilização do detector e descritor SIFT com o *K-Means*, o índice *Kappa* dos resultados varia entre 0,44 e 0,58, sendo considerados índices “bom”. A maior acurácia obtida para essa combinação foi de 59,51% com 600 *clusters*. Já com o detector FAST e descritor BRIEF com o *Mini Batch K-Means*, o índice *Kappa* dos resultados varia entre 0,71 e 0,83, sendo considerados índices “muito bom” e “excelente”. A maior acurácia obtida para essa combinação foi de 84,27% com 600 *clusters*.

O gráfico da Figura 23 exibe a comparação entre resultados apresentados na Tabela 7. É perceptível que ambas combinações são afetadas pelo tamanho do dicionário

Tabela 7 – Desempenho do modelo proposto na GTSRB.

Clusters	SIFT + SIFT		FAST + BRIEF	
	<i>K-Means</i>		<i>Mini Batch K-Means</i>	
	SVM One-versus-All			
	Acurácia (A%)	<i>Kappa</i> (K)	Acurácia (A%)	<i>Kappa</i> (K)
100	46,86%	0,4456	74,12%	0,7109
200	53,15%	0,5123	80,89%	0,7866
300	56,46%	0,5469	83,31%	0,8136
400	59,17%	0,5754	83,86%	0,8198
500	58,80%	0,5718	84,22%	0,8238
600	<b>59,51%</b>	<b>0,5795</b>	<b>84,27%</b>	<b>0,8245</b>

visual e que por restrições de *hardware* não foi possível utilizar tamanhos de dicionários visuais maiores.

A segunda combinação, composta em sua maioria por algoritmos de tempo real, obteve resultados superiores em relação ao desempenho para todas as quantidades de *clusters* utilizadas. Além disso, a quantidade de poder computacional necessária para a sua execução é mínima em relação a primeira combinação.

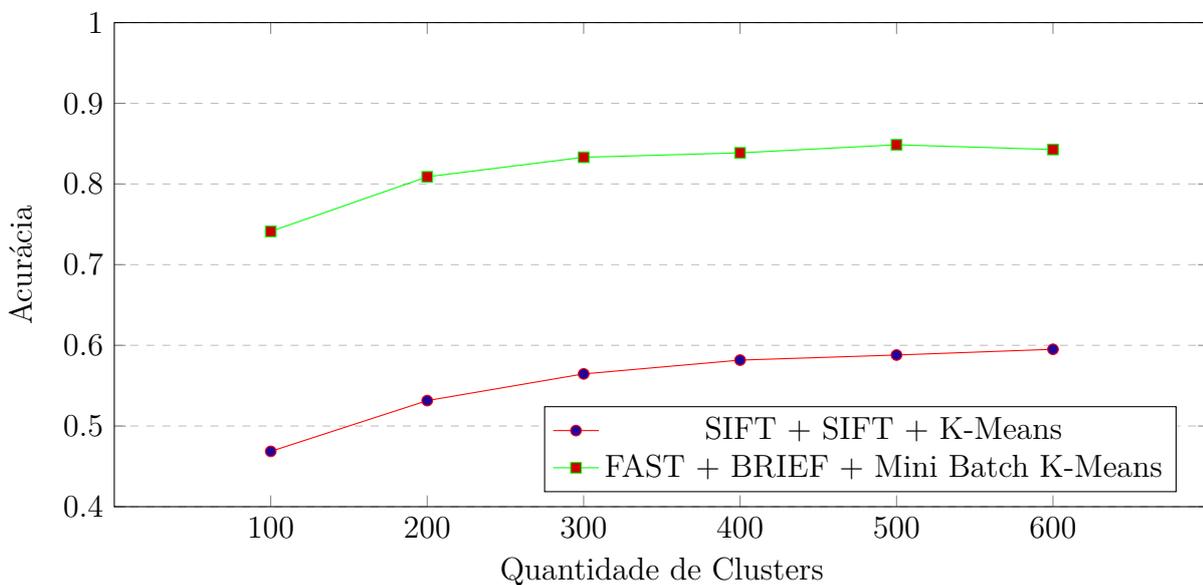


Figura 23 – Comparação do desempenho geral do modelo proposto na GTSRB.

## 5.4 Discussão e Considerações Finais

Este capítulo apresentou os resultados do modelo proposto baseado na classificação de placas de trânsito. Os resultados obtidos demonstram a performance do modelo ao utilizar diversas combinações de algoritmos. Devido a restrições de recursos, métodos dito

de tempo real e que necessitam de menor poder computacional foram postos a prova contra métodos clássicos da literatura.

Uma análise comparativa foi realizada utilizando diferentes métodos com o objetivo de justificar a escolha dos algoritmos utilizados no método proposto. Foram testados os detectores SIFT e FAST, os descritores SIFT e BRIEF, os algoritmos de agrupamento *K-Means* e *Mini Batch K-Means* e os classificadores SVM multiclasse com abordagem *One-versus-One* e *One-versus-All*. Foram realizados experimentos em três bases de imagens de portes distintos: BraTSD, BTSD e GTSRB.

Na BraTSD, onde a quantidade de imagens é pequena, o tamanho do dicionário visual foi variado entre 500 e 10.000 palavras. Foi encontrada uma acurácia máxima de 94,43%, obtendo um desempenho “excelente” com um índice *Kappa* igual a 0,9325 na combinação dos algoritmos FAST + BRIEF + *Mini Batch K-Means* + SVM *One-versus-All* com um dicionário de 8.000 palavras visuais. Além de obter os melhores resultados nas métricas analisadas, essa combinação de algoritmos foi capaz de executar em uma fração do tempo requisitado por outras combinações. Esse fato demonstra que tal combinação pode ser utilizada em *hardwares* de baixo custo com reduzido poder computacional.

Ainda na BraTSD, outras combinações obtiveram resultados próximos da melhor combinação encontrada. Com uma acurácia de 94,38%, obtendo um desempenho “excelente” com um índice *Kappa* igual a 0,9394 na combinação dos algoritmos SIFT + SIFT + *K-Means* + SVM *One-versus-All* com um dicionário de 10.000 palavras visuais. Apesar do resultado ser muito próximo da melhor combinação, o consumo de recurso realizado é muito superior, impossibilitando a sua utilização em *hardwares* de baixo custo com reduzido poder computacional.

Na BTSD, o tamanho do dicionário visual foi variado entre 100 e 600 palavras. Foi encontrada uma acurácia de 96,67%, obtendo um desempenho “excelente” com um índice *Kappa* igual a 0,9509 na combinação dos algoritmos FAST + BRIEF + *Mini Batch K-Means* + SVM *One-versus-All* com um dicionário de 500 palavras visuais. Novamente, essa combinação foi capaz de executar em uma fração do tempo requisitado por outras combinações.

Diferentemente dos resultados encontrados na BraTSD, a combinação de algoritmos SIFT + SIFT + *K-Means* + SVM *One-versus-All* não gerou resultados próximos da melhor combinação na BTSD. Foi encontrada uma acurácia de 65,39%, obtendo um desempenho “muito bom” com um índice *Kappa* igual a 0,6327.

Na GTSRB, o tamanho do dicionário visual foi variado entre 100 e 600 palavras. Foi encontrada uma acurácia de 84,27%, obtendo um desempenho “excelente” com um índice *Kappa* igual a 0,8245 na combinação dos algoritmos FAST + BRIEF + *Mini Batch K-Means* + SVM *One-versus-All* com um dicionário de 600 palavras visuais. Novamente,

essa combinação foi capaz de executar em uma fração do tempo requisitado por outras combinações.

Diferentemente dos resultados encontrados na BraTSD, a combinação de algoritmos SIFT + SIFT + *K-Means* + SVM *One-versus-All* não gerou resultados próximos da melhor combinação na GTSRB. Foi encontrada uma acurácia de 59,51%, obtendo um desempenho “bom” com um índice *Kappa* igual a 0,5795.

Foi perceptível o impacto que o classificador tem sobre o modelo proposto. Em todos os casos que o SVM *One-versus-One* foi utilizado, o desempenho do modelo ficou retraído. Além disso, esse algoritmo utiliza muito poder computacional e requer grandes quantidades de tempo para ser executado.

## 6 Conclusão e Trabalhos Futuros

Um modelo de classificação de placas de trânsito pode ser utilizado em diversas tarefas que normalmente necessitam da presença de um especialista humano. Esta pesquisa apresentou uma proposta de modelo de classificação de placas de trânsito utilizando descritores locais e *Bag of Features*.

### 6.1 Conclusão

Este estudo mostra a possibilidade de utilizar técnicas de inteligência computacional para a classificação de placas de trânsito. A principal contribuição está no fato de considerar que a utilização de algoritmos de tempo real pode prover excelente desempenho. Além disso, apresentou-se uma avaliação de diferentes combinações de algoritmos em diversas bases de imagens.

A *Brazilian Traffic Sign Dataset* (BraTSD), uma base com 2.122 imagens de placas de trânsito capturadas nas vias de Teresina, capital do Piauí, foi apresentada neste trabalho e está disponível em <https://github.com/hugos94/traffic-signs-database>. Essa base serviu para demonstrar qualitativamente o funcionamento do método proposto. Para uma comparação quantitativa com outros trabalhos da literatura, foram utilizadas as bases BTSD e GTSRB.

A composição dos algoritmos FAST, BRIEF, *Mini Batch K-Means* e SVM *One-vs-All* obtiveram os resultados mais promissores para o problema analisado. Foram encontradas acurácias máximas de 84,27% na base de placas de trânsito da Alemanha, 96,67% para a base de imagens da Bélgica e 94,43% na base de placas de trânsito do Brasil. Os índices *Kappa* obtidos pelos melhores resultados de cada base de imagens foram considerados “Excelentes”. Desse modo, com o aprimoramento das técnicas utilizadas, o modelo proposto pode obter resultados ainda mais expressivos dos que foram apresentados.

### 6.2 Trabalhos Futuros

Os resultados apresentados são promissores, mas podem ser melhorados. Na continuidade da pesquisa, as seguintes tarefas podem ser analisadas com o intuito de aperfeiçoar o método proposto de classificação de placas de trânsito: realização de ajustes nos hiperparâmetros dos algoritmos; utilização de métodos de detecção e descrição mais robustos e rápidos; uso de algoritmos de agrupamento com abordagens diferentes; aplicação de classificadores com metodologias alternativas; mesclagem entre as bases de dados utilizadas;

e a realização de tratamentos específicos nas palavras do dicionário visual.

# Referências

- ABBAS, F.; FAN, P. A hybrid low-latency d2d resource allocation scheme based on cellular v2x networks. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. [s.n.], 2018. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICCW.2018.8403512>>. Citado na página 1.
- ALSEIARI, F. A. A.; AUNG, Z. Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining. In: *2015 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. [s.n.], 2015. p. 148–153. Disponível em: <<https://doi.org/10.1109/ICSGCE.2015.7454287>>. Citado na página 24.
- ALTINOZ, O. T.; ALTINOZ, M. O. Classification of modern art movements with computational methods: Initial results. In: *2017 25th Signal Processing and Communications Applications Conference (SIU)*. [s.n.], 2017. p. 1–4. Disponível em: <<https://doi.org/10.1109/SIU.2017.7960640>>. Citado na página 27.
- ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, v. 46, n. 3, p. 175–185, 1992. Disponível em: <<https://doi.org/10.2307/2685209>>. Citado na página 6.
- ARIFIN, N. A.; IRAWAN, B.; SETIANINGSIH, C. Traffic sign recognition application using speeded-up robust features (surf) and support vector machine (svm) based on android. In: *2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. [s.n.], 2017. p. 52–58. Disponível em: <<https://doi.org/10.1109/APWiMob.2017.8284004>>. Citado na página 3.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: \_\_\_\_\_. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 404–417. ISBN 978-3-540-33833-8. Disponível em: <[https://doi.org/10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32)>. Citado na página 7.
- BÉJAR, J. K-means vs mini batch k-means : A comparison. In: . [s.n.], 2013. Disponível em: <<https://upcommons.upc.edu/bitstream/handle/2117/23414/R13-8.pdf>>. Citado na página 26.
- BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM*, ACM, New York, NY, USA, v. 18, n. 9, p. 509–517, set. 1975. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/361002.361007>>. Citado na página 5.
- BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738. Disponível em: <<https://www.springer.com/br/book/9780387310732>>. Citado na página 28.

- BONJORN, N.; FOUKALAS, F.; POP, P. Enhanced 5g v2x services using sidelink device-to-device communications. In: *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. [s.n.], 2018. p. 1–7. Disponível em: <<https://doi.org/10.23919/MedHocNet.2018.8407085>>. Citado na página 1.
- BOSCH, A.; ZISSERMAN, A.; MUNOZ, X. Representing shape with a spatial pyramid kernel. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*. New York, NY, USA: ACM, 2007. (CIVR '07), p. 401–408. ISBN 978-1-59593-733-9. Disponível em: <<http://doi.acm.org/10.1145/1282280.1282340>>. Citado na página 5.
- BRASIL, R. F. do. *LEI Nº 9.503, DE 23 DE SETEMBRO DE 1997 - Código de Trânsito Brasileiro (CTB)*. 1997. <[http://www.planalto.gov.br/ccivil\\_03/leis/L9503.htm](http://www.planalto.gov.br/ccivil_03/leis/L9503.htm)>. Acessado em: 21-04-2018. Citado na página 1.
- BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>. Citado na página 5.
- CALONDER, M. et al. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 34, n. 7, p. 1281–1298, July 2012. ISSN 0162-8828. Disponível em: <<https://doi.org/10.1109/TPAMI.2011.222>>. Citado na página 21.
- CALONDER, M. et al. Brief: Binary robust independent elementary features. In: *Proceedings of the 11th European Conference on Computer Vision: Part IV*. Berlin, Heidelberg: Springer-Verlag, 2010. (ECCV'10), p. 778–792. ISBN 3-642-15560-X, 978-3-642-15560-4. Disponível em: <<http://dl.acm.org/citation.cfm?id=1888089.1888148>>. Citado 6 vezes nas páginas 15, 4, 19, 21, 22 e 34.
- CHEN, L. W.; HO, Y. F. Centimeter-grade metropolitan positioning for lane-level intelligent transportation systems based on the internet of vehicles. *IEEE Transactions on Industrial Informatics*, p. 1–1, 2018. ISSN 1551-3203. Disponível em: <<https://doi.org/10.1109/TII.2018.2854901>>. Citado na página 1.
- CHEN, W. et al. Platoon control for connected vehicles based on the v2x communications: Design and implementation. In: *2018 Chinese Control And Decision Conference (CCDC)*. [s.n.], 2018. p. 6552–6557. Disponível em: <<https://doi.org/10.1109/CCDC.2018.8408282>>. Citado na página 1.
- CHOI, S.; HAN, S. Fast image retrieval with grid-based keypoint detector and binary descriptor. In: *2014 International Conference on Information and Communication Technology Convergence (ICTC)*. [s.n.], 2014. p. 679–680. ISSN 2162-1233. Disponível em: <<https://doi.org/10.1109/ICTC.2014.6983253>>. Citado na página 16.
- CIRESAN, D. C.; MEIER, U.; SCHMIDHUBER, J. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012. Disponível em: <<http://arxiv.org/abs/1202.2745>>. Citado na página 6.
- CORNO, M. Design, analysis, and validation of a haptic-based driver support system for traction control. *IEEE Transactions on Intelligent Transportation Systems*, v. 14, n. 4, p. 1849–1859, Dec 2013. ISSN 1524-9050. Disponível em: <<https://doi.org/10.1109/TITS.2013.2268316>>. Citado na página 1.

- CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 20, n. 3, p. 273–297, set. 1995. ISSN 0885-6125. Disponível em: <<http://dx.doi.org/10.1023/A:1022627411411>>. Citado 6 vezes nas páginas 4, 5, 6, 7, 27 e 36.
- COULIBALY, M. et al. Development of a demonstrator: smart-parking. In: *2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON)*. [s.n.], 2018. p. 172–176. Disponível em: <<https://doi.org/10.1109/MELCON.2018.8379088>>. Citado na página 1.
- CSURKA, G. et al. Visual categorization with bags of keypoints. In: *In Workshop on Statistical Learning in Computer Vision, ECCV*. [s.n.], 2004. p. 1–22. Disponível em: <<http://people.eecs.berkeley.edu/~efros/courses/AP06/Papers/csurka-eccv-04.pdf>>. Citado 2 vezes nas páginas 7 e 22.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [s.n.], 2005. v. 1, p. 886–893 vol. 1. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2005.177>>. Citado na página 5.
- DESHPANDE, A. V.; SUBASHINI, M. M. A survey of different image processing methods for the design and development of an efficient traffic sign board recognition system. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. [s.n.], 2017. p. 340–343. Disponível em: <<http://doi.org/10.1109/RTEICT.2017.8256612>>. Citado na página 2.
- ERHAN, C. et al. Traffic sign detection and recognition fusing feature descriptors. In: *2013 21st Signal Processing and Communications Applications Conference (SIU)*. [s.n.], 2013. p. 1–4. Disponível em: <<https://doi.org/10.1109/SIU.2013.6531427>>. Citado na página 19.
- FAN, Y.; ZHANG, W. Traffic sign detection and classification for advanced driver assistant systems. In: *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. [s.n.], 2015. p. 1335–1339. Disponível em: <<https://doi.org/10.1109/FSKD.2015.7382137>>. Citado na página 3.
- FANG, C.-Y.; CHEN, S.-W.; FUH, C.-S. Road-sign detection and tracking. *IEEE Transactions on Vehicular Technology*, v. 52, n. 5, p. 1329–1341, Sept 2003. ISSN 0018-9545. Disponível em: <<https://doi.org/10.1109/TVT.2003.810999>>. Citado na página 9.
- FANG, C. Y. et al. A road sign recognition system based on dynamic visual model. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*. [s.n.], 2003. v. 1, p. I-750–I-755 vol.1. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2003.1211428>>. Citado na página 9.
- FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861 – 874, 2006. ISSN 0167-8655. ROC Analysis in Pattern Recognition. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S016786550500303X>>. Citado na página 29.
- FEI-FEI, L.; PERONA, P. A bayesian hierarchical model for learning natural scene categories. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern*

*Recognition (CVPR'05)*. [s.n.], 2005. v. 2, p. 524–531 vol. 2. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2005.16>>. Citado 2 vezes nas páginas 19 e 22.

FEIZOLLAH, A. et al. Comparative study of k-means and mini batch k-means clustering algorithms in android malware detection using network traffic analysis. In: *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*. [s.n.], 2014. p. 193–197. Disponível em: <<https://doi.org/10.1109/ISBAST.2014.7013120>>. Citado 2 vezes nas páginas 24 e 26.

FISHER, R. A. The statistical utilization of multiple measurements. *Annals of Eugenics*, v. 8, n. 4, p. 376–386, August 1938. Disponível em: <<http://doi.org/10.1111/j.1469-1809.1938.tb02189.x>>. Citado 2 vezes nas páginas 5 e 6.

FITRIYANI, S. R.; MURFI, H. The k-means with mini batch algorithm for topics detection on online news. In: *2016 4th International Conference on Information and Communication Technology (ICoICT)*. [s.n.], 2016. p. 1–5. Disponível em: <<https://doi.org/10.1109/ICoICT.2016.7571914>>. Citado na página 24.

FREEMAN, W. T.; ROTH, M. *Orientation Histograms for Hand Gesture Recognition*. Cambridge, MA 02139, 1994. Disponível em: <<http://www.merl.com/publications/TR94-03/>>. Citado na página 5.

F.R.S., K. P. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor and Francis, v. 2, n. 11, p. 559–572, 1901. Disponível em: <<https://doi.org/10.1080/14786440109462720>>. Citado na página 20.

FUSTER, J. M. The prefrontal cortex—an update time is of the essence. *Neuron*, v. 30, p. 319–333, 2001. Disponível em: <[http://doi.org/10.1016/S0896-6273\(01\)00285-9](http://doi.org/10.1016/S0896-6273(01)00285-9)>. Citado na página 7.

GIONIS, A.; INDYK, P.; MOTWANI, R. Similarity search in high dimensions via hashing. In: *Proceedings of the 25th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. (VLDB '99), p. 518–529. ISBN 1-55860-615-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=645925.671516>>. Citado na página 20.

GUAN, H. et al. Robust traffic-sign detection and classification using mobile lidar data with digital images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 11, n. 5, p. 1715–1724, May 2018. ISSN 1939-1404. Disponível em: <<https://doi.org/10.1109/JSTARS.2018.2810143>>. Citado na página 3.

HAMMING, R. W. Error detecting and error correcting codes. *The Bell System Technical Journal*, v. 29, n. 2, p. 147–160, April 1950. ISSN 0005-8580. Disponível em: <<https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>>. Citado na página 21.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: *Proceedings of the Fourth Alvey Vision Conference*. [s.n.], 1988. p. 147–151. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.4816&rep=rep1&type=pdf>>. Citado na página 16.

- HIBA, C.; HAMID, Z.; OMAR, A. An improved breast tissue density classification framework using bag of features model. In: *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. [s.n.], 2016. p. 405–409. Disponível em: <<https://doi.org/10.1109/CIST.2016.7805081>>. Citado na página 27.
- HOOSE, N. *Computer image processing in traffic engineering*. Research Studies Press, 1991. (Research Studies Press in Traffic Engineering). ISBN 9780863801136. Disponível em: <<https://books.google.com.br/books?id=IMlhAQAAIAAJ>>. Citado na página 9.
- HOTELLING, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, Warwick and York, Baltimore, MA, v. 24, p. 417–441, 1933. Disponível em: <<http://dx.doi.org/10.1037/h0071325>>. Citado na página 20.
- HOUBEN, S. et al. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. [s.n.], 2013. p. 1–8. ISSN 2161-4407. Disponível em: <<http://doi.org/10.1109/IJCNN.2013.6706807>>. Citado 2 vezes nas páginas 5 e 12.
- HSU, C. C.; LIN, C. W. Unsupervised convolutional neural networks for large-scale image clustering. In: *2017 IEEE International Conference on Image Processing (ICIP)*. [s.n.], 2017. p. 390–394. Disponível em: <<https://doi.org/10.1109/ICIP.2017.8296309>>. Citado na página 24.
- HSU, C. C.; LIN, C. W. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, v. 20, n. 2, p. 421–429, Feb 2018. ISSN 1520-9210. Disponível em: <<https://doi.org/10.1109/TMM.2017.2745702>>. Citado na página 24.
- HU, W. et al. Fast branch convolutional neural network for traffic sign recognition. *IEEE Intelligent Transportation Systems Magazine*, v. 9, n. 3, p. 114–126, Fall 2017. ISSN 1939-1390. Disponível em: <<https://doi.org/10.1109/MITS.2017.2709780>>. Citado 2 vezes nas páginas 7 e 8.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing*, v. 70, n. 1, p. 489 – 501, 2006. ISSN 0925-2312. Neural Networks. Disponível em: <<https://doi.org/10.1016/j.neucom.2005.12.126>>. Citado na página 7.
- HUANG, Z. et al. An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Transactions on Cybernetics*, v. 47, n. 4, p. 920–933, April 2017. ISSN 2168-2267. Disponível em: <<https://doi.org/10.1109/TCYB.2016.2533424>>. Citado 2 vezes nas páginas 7 e 8.
- INDYK, P.; MOTWANI, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM, 1998. (STOC '98), p. 604–613. ISBN 0-89791-962-9. Disponível em: <<http://doi.acm.org/10.1145/276698.276876>>. Citado na página 20.
- ITE, I. of T. E.; WOLSHON, B.; PANDE, A. *Traffic Engineering Handbook*. Wiley, 2016. (Institute of Transportation Engineers). ISBN 9781118762301. Disponível em: <<https://books.google.com.br/books?id=V1XKCQAAQBAJ>>. Citado na página 11.

- JANG, C.; KIM, Y. K. A feasibility study of vehicle pose estimation using road sign information. In: *2016 16th International Conference on Control, Automation and Systems (ICCAS)*. [s.n.], 2016. p. 397–401. Disponível em: <https://doi.org/10.1109/ICCAS.2016.7832351>. Citado na página 19.
- JIANG, G.-Y.; CHOI, T. Y. Robust detection of landmarks in color image based on fuzzy set theory. In: *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on*. [s.n.], 1998. v. 2, p. 968–971 vol.2. Disponível em: <https://doi.org/10.1109/ICOSP.1998.770774>. Citado na página 11.
- JIANG, Y.-G.; NGO, C.-W.; YANG, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*. New York, NY, USA: ACM, 2007. (CIVR '07), p. 494–501. ISBN 978-1-59593-733-9. Disponível em: <http://doi.acm.org/10.1145/1282280.1282352>. Citado na página 24.
- JIN, C. M.; OMAR, Z.; JAWARD, M. H. A mobile application of american sign language translation via image processing algorithms. In: *2016 IEEE Region 10 Symposium (TENSYMP)*. [s.n.], 2016. p. 104–109. Disponível em: <https://doi.org/10.1109/TENCONSpring.2016.7519386>. Citado na página 27.
- JIN, J.; FU, K.; ZHANG, C. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, v. 15, n. 5, p. 1991–2000, Oct 2014. ISSN 1524-9050. Disponível em: <http://doi.org/10.1109/TITS.2014.2308281>. Citado na página 7.
- JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: *Proceedings of the 10th European Conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 1998. (ECML'98), p. 137–142. ISBN 3-540-64417-2, 978-3-540-64417-0. Disponível em: <https://doi.org/10.1007/BFb0026683>. Citado na página 22.
- JURIŠIĆ, F.; FILKOVIĆ, I.; KALAFATIĆ, Z. Multiple-dataset traffic sign classification with onecnn. In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. [s.n.], 2015. p. 614–618. Disponível em: <https://doi.org/10.1109/ACPR.2015.7486576>. Citado 2 vezes nas páginas 6 e 8.
- KASHEVNIK, A. et al. Context-based driver support system development: Methodology and case study. In: *2017 21st Conference of Open Innovations Association (FRUCT)*. [s.n.], 2017. p. 162–171. Disponível em: <https://doi.org/10.23919/FRUCT.2017.8250179>. Citado na página 1.
- KASHIHARA, K. A driver support system to prevent traffic accidents caused by optical illusions. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [s.n.], 2014. p. 3561–3566. ISSN 1062-922X. Disponível em: <https://doi.org/10.1109/SMC.2014.6974482>. Citado na página 1.
- KIM, H. et al. Lane positioning in highways based on road-sign tracking using kalman filter. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [s.n.], 2014. p. 2379–2384. ISSN 1062-922X. Disponível em: <https://doi.org/10.1109/SMC.2014.6974282>. Citado na página 19.

KUON, I.; TESSIER, R.; ROSE, J. Fpga architecture: Survey and challenges. *Found. Trends Electron. Des. Autom.*, Now Publishers Inc., Hanover, MA, USA, v. 2, n. 2, p. 135–253, fev. 2008. ISSN 1551-3939. Disponível em: <<http://dx.doi.org/10.1561/1000000005>>. Citado na página 7.

KUS, M. C.; GOKMEN, M.; ETANER-UYAR, S. Traffic sign recognition using scale invariant feature transform and color classification. In: *2008 23rd International Symposium on Computer and Information Sciences*. [s.n.], 2008. p. 1–6. Disponível em: <<https://doi.org/10.1109/ISCIS.2008.4717875>>. Citado 2 vezes nas páginas 17 e 19.

LALONDE, M.; LI, Y. *Survey of the state of the art for sub-project 2.4: Road sign recognition*. [S.l.], 1995. Technical report, Collection scientifique et technique, CRIM-IIT-95/09-35. Citado na página 11.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics*, [Wiley, International Biometric Society], v. 33, n. 1, p. 159–174, 1977. ISSN 0006341X, 15410420. Disponível em: <<http://www.jstor.org/stable/2529310>>. Citado na página 30.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. [s.n.], 2006. v. 2, p. 2169–2178. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2006.68>>. Citado na página 24.

LI, L. m. et al. Feature detection and matching for traffic sign images. In: *Proceedings of the 10th World Congress on Intelligent Control and Automation*. [s.n.], 2012. p. 4628–4632. Disponível em: <<https://doi.org/10.1109/WCICA.2012.6359356>>. Citado 2 vezes nas páginas 17 e 19.

LINDEBERG, T. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 30, n. 2, p. 79–116, nov. 1998. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1023/A:1008045108935>>. Citado na página 16.

LOSCIALE, M. et al. A novel icn-based communication bus for intelligent transportation systems. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. [s.n.], 2018. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICCW.2018.8403577>>. Citado na página 1.

LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision - Volume 2 - Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999. (ICCV '99), p. 1150–. ISBN 0-7695-0164-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=850924.851523>>. Citado 4 vezes nas páginas 7, 17, 19 e 34.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 2, p. 91–110, nov. 2004. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>>. Citado 2 vezes nas páginas 19 e 20.

LÓPEZ, J. R. et al. A genetic programming approach for driving score calculation in the context of intelligent transportation systems. *IEEE Sensors Journal*, p. 1–1, 2018. ISSN 1530-437X. Disponível em: <<https://doi.org/10.1109/JSEN.2018.2856112>>. Citado na página 1.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967. p. 281–297. Disponível em: <<http://projecteuclid.org/euclid.bsmsp/1200512992>>. Citado 3 vezes nas páginas 7, 24 e 35.

MAJI, S.; MALIK, J. *Fast and Accurate Digit Classification*. [S.l.], 2009. Disponível em: <<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-159.html>>. Citado na página 5.

MALIK, Z.; SIDDIQI, I. Detection and recognition of traffic signs from road scene images. In: *2014 12th International Conference on Frontiers of Information Technology*. [s.n.], 2014. p. 330–335. Disponível em: <<https://doi.org/10.1109/FIT.2014.68>>. Citado 2 vezes nas páginas 17 e 19.

MARTINEZ, A. M.; KAK, A. C. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 2, p. 228–233, Feb 2001. ISSN 0162-8828. Disponível em: <<https://doi.org/10.1109/34.908974>>. Citado 2 vezes nas páginas 5 e 6.

MATHIAS, M. et al. Traffic sign recognition – how far are we from the solution? In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. [s.n.], 2013. p. 1–8. ISSN 2161-4393. Disponível em: <<https://doi.org/10.1109/IJCNN.2013.6707049>>. Citado 3 vezes nas páginas 5, 6 e 8.

MöGELMOSE, A.; TRIVEDI, M. M.; MOESLUND, T. B. Traffic sign detection and analysis: Recent studies and emerging trends. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. [s.n.], 2012. p. 1310–1314. ISSN 2153-0017. Disponível em: <<http://doi.org/10.1109/ITSC.2012.6338900>>. Citado na página 2.

MÜLLER, P.; DEVELOPERS, M. *Manjaro Linux*. 2011. Acessado em: 20-04-2017. Disponível em: <<https://manjaro.org>>. Citado na página 37.

NGUYEN, N.; YOSHITAKA, A. Classification and temporal localization for human-human interactions. In: *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. [s.n.], 2016. p. 74–81. Disponível em: <<https://doi.org/10.1109/BigMM.2016.87>>. Citado na página 27.

NOWAK, E.; JURIE, F.; TRIGGS, B. Sampling strategies for bag-of-features image classification. In: *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*. Berlin, Heidelberg: Springer-Verlag, 2006. (ECCV'06), p. 490–503. ISBN 3-540-33838-1, 978-3-540-33838-3. Disponível em: <[http://dx.doi.org/10.1007/11744085\\_38](http://dx.doi.org/10.1007/11744085_38)>. Citado na página 19.

PARODI, P.; PICCIOLI, G. A feature-based recognition scheme for traffic scenes. In: *Intelligent Vehicles '95 Symposium., Proceedings of the*. [s.n.], 1995. p. 229–234. Disponível em: <<https://doi.org/10.1109/IVS.1995.528285>>. Citado na página 11.

PASSALIS, N.; TEFAS, A. Information clustering using manifold-based optimization of the bag-of-features representation. *IEEE Transactions on Cybernetics*, PP, n. 99, p. 1–12, 2016. ISSN 2168-2267. Disponível em: <<https://doi.org/10.1109/TCYB.2016.2623581>>. Citado na página 24.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Disponível em: <<http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>>. Citado na página 37.

PENG, K.; LEUNG, V. C. M.; HUANG, Q. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, v. 6, p. 11897–11906, 2018. Disponível em: <<https://doi.org/10.1109/ACCESS.2018.2810267>>. Citado na página 24.

PENG, X. et al. Traffic sign recognition with transfer learning. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [s.n.], 2017. p. 1–7. Disponível em: <<https://doi.org/10.1109/SSCI.2017.8285332>>. Citado na página 3.

PETERMEIJER, S. M. et al. The effect of haptic support systems on driver performance: A literature survey. *IEEE Transactions on Haptics*, v. 8, n. 4, p. 467–479, Oct 2015. ISSN 1939-1412. Disponível em: <<https://doi.org/10.1109/TOH.2015.2437871>>. Citado na página 1.

PHU, K. T.; OO, L. L. Traffic sign recognition system using feature points. In: *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. [s.n.], 2018. p. 1–6. Disponível em: <<https://doi.org/10.1109/RCIS.2018.8406656>>. Citado 2 vezes nas páginas 1 e 3.

POWERS, D. M. W. Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. *School of Informatics and Engineering Technical Reports (24p ver. of ECAI'2008 Evaluation Evaluation)*, Flinders University, Adelaide, Australia, v. 2, n. SIE-07-001, p. 37–63, 01 2007. ISSN 2229-3981. Disponível em: <[http://www.academia.edu/3090500/Evaluation\\_From\\_Precision\\_Recall\\_and\\_F-Measure\\_to\\_ROC\\_Informedness\\_Markedness\\_and\\_Correlation](http://www.academia.edu/3090500/Evaluation_From_Precision_Recall_and_F-Measure_to_ROC_Informedness_Markedness_and_Correlation)>. Citado na página 30.

QASAIMAH, M.; SAGAHYROON, A.; SHANABLEH, T. Fpga-based parallel hardware architecture for real-time image classification. *IEEE Transactions on Computational Imaging*, v. 1, n. 1, p. 56–70, March 2015. ISSN 2333-9403. Disponível em: <<https://doi.org/10.1109/TCI.2015.2424077>>. Citado 2 vezes nas páginas 6 e 8.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, n. 1, p. 81–106, Mar 1986. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1007/BF00116251>>. Citado na página 18.

RATNASARI, R.; IRAWAN, B.; SETIANINGSIH, C. Traffic sign recognition application using scale invariant feature transform method and support vector machine based on android. In: *2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. [s.n.], 2017. p. 47–51. Disponível em: <<https://doi.org/10.1109/APWiMob.2017.8284003>>. Citado 2 vezes nas páginas 3 e 19.

RIESENHUBER, M.; POGGIO, T. Hierarchical models of object recognition in cortex. v. 2, n. 11, p. 1019–1025, November 1999. Disponível em: <<http://dx.doi.org/10.1038/14819>>. Citado na página 6.

ROSENFELD, G. H.; FITZPATRICK-LINS, K. A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering and Remote Sensing (PE&RS)*, USGS Publications Warehouse, v. 52, n. 2, p. 223–227, 1986. Disponível em: <<http://pubs.er.usgs.gov/publication/70014667>>. Citado na página 30.

ROSSUM, G. *Python Reference Manual*. Amsterdam, The Netherlands, The Netherlands, 1995. Disponível em: <<https://ir.cwi.nl/pub/5008>>. Citado na página 37.

ROSTEN, E.; DRUMMOND, T. Fusing points and lines for high performance tracking. In: *IEEE International Conference on Computer Vision*. [s.n.], 2005. v. 2, p. 1508–1511. Disponível em: <[http://www.edwardrosten.com/work/rosten\\_2005\\_tracking.pdf](http://www.edwardrosten.com/work/rosten_2005_tracking.pdf)>. Citado 3 vezes nas páginas 17, 18 e 34.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: *European Conference on Computer Vision*. [s.n.], 2006. v. 1, p. 430–443. Disponível em: <[http://www.edwardrosten.com/work/rosten\\_2006\\_machine.pdf](http://www.edwardrosten.com/work/rosten_2006_machine.pdf)>. Citado na página 18.

ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 1, p. 105–119, Jan 2010. ISSN 0162-8828. Disponível em: <<https://doi.org/10.1109/TPAMI.2008.275>>. Citado 2 vezes nas páginas 3 e 18.

ROXAS, E. A. et al. Vision-based traffic sign compliance evaluation using convolutional neural network. In: *2018 IEEE International Conference on Applied System Invention (ICASI)*. [s.n.], 2018. p. 120–123. Disponível em: <<https://doi.org/10.1109/ICASI.2018.8394502>>. Citado 2 vezes nas páginas 2 e 3.

SALEM, F.; ELHILLALI, Y.; NIAR, S. Efficient modelling of ieee 802.11p mac output process for v2x interworking enhancement. *IET Networks*, v. 7, n. 4, p. 210–219, 2018. ISSN 2047-4954. Disponível em: <<https://doi.org/10.1049/iet-net.2017.0228>>. Citado na página 1.

SALTON, G. The state of retrieval system evaluation. *Inf. Process. Manage.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 28, n. 4, p. 441–449, march 1992. ISSN 0306-4573. Disponível em: <[https://doi.org/10.1016/0306-4573\(92\)90002-H](https://doi.org/10.1016/0306-4573(92)90002-H)>. Citado na página 22.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, november 1975. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/361219.361220>>. Citado na página 22.

SCULLEY, D. Web-scale k-means clustering. In: *Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA: ACM, 2010. (WWW '10), p. 1177–1178. ISBN 978-1-60558-799-8. Disponível em: <<http://doi.acm.org/10.1145/1772690.1772862>>. Citado 4 vezes nas páginas 4, 24, 26 e 35.

SELVER, M. A. et al. Camera based driver support system for rail extraction using 2-d gabor wavelet decompositions and morphological analysis. In: *2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*. [s.n.], 2016. p. 270–275. Disponível em: <<https://doi.org/10.1109/ICIRT.2016.7588744>>. Citado na página 1.

SERMANET, P.; LECUN, Y. Traffic sign recognition with multi-scale convolutional networks. In: *The 2011 International Joint Conference on Neural Networks*. [s.n.], 2011. p. 2809–2813. ISSN 2161-4393. Disponível em: <<http://doi.org/10.1109/IJCNN.2011.6033589>>. Citado na página 6.

SERRE, T. et al. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. In: *AI Memo 2005-036, CBCL Memo*. [s.n.], 2007. Disponível em: <<http://serre-lab.clps.brown.edu/wp-content/uploads/2012/08/GetTRDoc.pdf>>. Citado na página 17.

SHEIKH, M. A. A.; KOLE, A.; MAITY, T. Traffic sign detection and classification using colour feature and neural network. In: *2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI)*. [s.n.], 2016. p. 307–311. Disponível em: <<https://doi.org/10.1109/ICICPI.2016.7859723>>. Citado na página 3.

SHEN, Z. et al. A car following based rate control algorithm for vanet in intelligent transportation systems. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. [s.n.], 2018. p. 1–6. Disponível em: <<https://doi.org/10.1109/WCNC.2018.8376976>>. Citado na página 1.

SIVIC, J.; ZISSERMAN, A. Video google: a text retrieval approach to object matching in videos. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. [s.n.], 2003. p. 1470–1477 vol.2. Disponível em: <<https://doi.org/10.1109/ICCV.2003.1238663>>. Citado na página 24.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, JMLR.org, v. 15, n. 1, p. 1929–1958, jan. 2014. ISSN 1532-4435. Disponível em: <<http://dl.acm.org/citation.cfm?id=2627435.2670313>>. Citado na página 6.

STALLKAMP, J. et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, v. 32, p. 323–332, 2012. ISSN 0893-6080. Selected Papers from IJCNN 2011. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608012000457>>. Citado 3 vezes nas páginas 15, 14 e 43.

SUPRIYANTO, C.; LUTHFIARTA, A.; ZENIARJA, J. An unsupervised approach for traffic sign recognition based on bag-of-visual-words. In: *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*. [s.n.], 2016. p. 1–4. Disponível em: <<https://doi.org/10.1109/ICITEED.2016.7863253>>. Citado 2 vezes nas páginas 7 e 8.

SüRüCü, E. A.; DOĞAN, H. Traffic sign recognition with hierarchical convolutional neural network. In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. [s.n.], 2018. p. 1–4. Disponível em: <<https://doi.org/10.1109/SIU.2018.8404702>>. Citado 2 vezes nas páginas 2 e 3.

TAKAKI, M.; FUJIYOSHI, H. Traffic sign recognition using sift features. *Journal of the Institute of Electrical Engineers of Japan C (Electronic, Information and Systems Division)*, v. 129, n. 5, p. 824–831, 01 2009. Disponível em: <<http://doi.org/10.1541/ieejieiss.129.824>>. Citado 2 vezes nas páginas 17 e 19.

TARAMOV, A.; SHILOV, N. A systematic review of proactive driver support systems and underlying technologies. In: *2017 20th Conference of Open Innovations Association (FRUCT)*. [s.n.], 2017. p. 448–459. Disponível em: <<https://doi.org/10.23919/FRUCT.2017.8071347>>. Citado na página 1.

TIMOFTE, R.; GOOL, L. V. Sparse representation based projections. In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011. p. 61.1–61.12. ISBN 1-901725-43-X. Disponível em: <<http://doi.org/10.5244/C.25.61>>. Citado na página 6.

TIMOFTE, R.; GOOL, L. V. Iterative nearest neighbors for classification and dimensionality reduction. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [s.n.], 2012. p. 2456–2463. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2012.6247960>>. Citado na página 6.

TIMOFTE, R.; ZIMMERMANN, K.; GOOL, L. V. Multi-view traffic sign detection, recognition, and 3d localisation. In: *2009 IEEE Computer Society Workshop on Application of Computer Vision (WACV)*. Snowbird, Utah, USA: [s.n.], 2009. p. 1–8. ISBN 978-1-4244-5496-9. ISSN 1550-5790. Disponível em: <<https://doi.org/10.1109/WACV.2009.5403121>>. Citado 5 vezes nas páginas 15, 5, 12, 13 e 42.

TIMOFTE, R.; ZIMMERMANN, K.; GOOL, L. V. Multi-view traffic sign detection, recognition, and 3d localisation. *Machine Vision and Applications*, v. 25, n. 3, p. 633–647, Apr 2014. ISSN 1432-1769. Disponível em: <<https://doi.org/10.1007/s00138-011-0391-3>>. Citado na página 13.

TOPRAK, T. et al. Camera based pedestrian detection for railway driver support systems. In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. [s.n.], 2018. p. 1–4. Disponível em: <<https://doi.org/10.1109/SIU.2018.8404574>>. Citado na página 1.

TRAFFIC, U. N. C. on R. *TRANSPORT AND COMMUNICATIONS*. Viena: Convention on Road Traffic, 1968. 1-66 p. Disponível em: <<https://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf>>. Citado na página 9.

TRANSPORT, D. for. *The Traffic Signs Regulations and General Directions 2016*. 1. ed. STATUTORY INSTRUMENTS, 2016. v. 1. (1, v. 1). ISBN 9780580796159. Disponível em: <<http://www.legislation.gov.uk/uksi/2016/362/contents/made>>. Citado na página 9.

TUYTELAARS, T. Dense interest points. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [s.n.], 2010. p. 2281–2288. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2010.5539911>>. Citado na página 16.

VELVIZHY, P.; PAVITHRA; KANNAN, A. Automatic food recognition system for diabetic patients. In: *2014 Sixth International Conference on Advanced Computing (ICoAC)*. [s.n.], 2014. p. 329–334. ISSN 2377-6927. Disponível em: <<https://doi.org/10.1109/ICoAC.2014.7229735>>. Citado na página 24.

- VIDYA, R.; PRIYANKKA, R. P. J.; KUMAR, G. N. A system learning of connection with humans by online social networking — a rapport by means of creating usable customer intelligence from social media data: Clustering the social communities. In: *2014 International Conference on Science Engineering and Management Research (ICSEMR)*. [s.n.], 2014. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICSEMR.2014.7043642>>. Citado na página 24.
- VITABILE, S.; GENTILE, A.; SORBELLO, F. A neural network based automatic road signs recognizer. In: *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*. [s.n.], 2002. v. 3, p. 2315–2320. ISSN 1098-7576. Disponível em: <<https://doi.org/10.1109/IJCNN.2002.1007503>>. Citado na página 11.
- VITABILE, S.; SORBELLO, F. Pictogram road signs detection and understanding in outdoor scenes. In: VERLY, J. (Ed.). *Enhanced and Synthetic Vision 1998*. [s.n.], 1998. v. 3364, p. 359–370. Disponível em: <<http://adsabs.harvard.edu/abs/1998SPIE.3364..359V>>. Citado na página 11.
- WANG, C. Research and application of traffic sign detection and recognition based on deep learning. In: *2018 International Conference on Robots Intelligent System (ICRIS)*. [s.n.], 2018. p. 150–152. Disponível em: <<https://doi.org/10.1109/ICRIS.2018.00047>>. Citado 2 vezes nas páginas 2 e 3.
- WANG, F. et al. Gesture recognition based on bof and its application in human-machine interaction of service robot. In: *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. [s.n.], 2016. p. 115–120. Disponível em: <<https://doi.org/10.1109/CYBER.2016.7574806>>. Citado na página 27.
- WANG, M. S.; ZHANG, Z. R. Fpga implementation of hog based multi-scale pedestrian detection. In: *2018 IEEE International Conference on Applied System Invention (ICASI)*. [s.n.], 2018. p. 1099–1102. Disponível em: <<https://doi.org/10.1109/ICASI.2018.8394472>>. Citado na página 1.
- WEI, L.; RUNGE, L.; XIAOLEI, L. Traffic sign detection and recognition via transfer learning. In: *2018 Chinese Control And Decision Conference (CCDC)*. [s.n.], 2018. p. 5884–5887. Disponível em: <<https://doi.org/10.1109/CCDC.2018.8408160>>. Citado 2 vezes nas páginas 2 e 3.
- WRIGHT, J. et al. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 2, p. 210–227, Feb 2009. ISSN 0162-8828. Disponível em: <<https://doi.org/10.1109/TPAMI.2008.79>>. Citado na página 6.
- YANG, B.; ZHOU, L.; DENG, Z. C-hmax: Artificial cognitive model inspired by the color vision mechanism of the human brain. *Tsinghua Science and Technology*, v. 18, n. 1, p. 51–56, Feb 2013. Disponível em: <<https://doi.org/10.1109/TST.2013.6449407>>. Citado 2 vezes nas páginas 6 e 8.
- YUAN, Y.; MENG, M. Q. H. Polyp classification based on bag of features and saliency in wireless capsule endoscopy. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. [s.n.], 2014. p. 3930–3935. ISSN 1050-4729. Disponível em: <<https://doi.org/10.1109/ICRA.2014.6907429>>. Citado na página 24.

ZAKLOUTA, F.; STANCIULESCU, B. Real-time traffic-sign recognition using tree classifiers. *IEEE Transactions on Intelligent Transportation Systems*, v. 13, n. 4, p. 1507–1514, Dec 2012. ISSN 1524-9050. Disponível em: <<https://doi.org/10.1109/TITS.2012.2225618>>. Citado 2 vezes nas páginas 5 e 8.

ZAVADIL, J.; TUMA, J.; SANTOS, V. M. F. Traffic signs detection using blob analysis and pattern recognition. In: *Proceedings of the 13th International Carpathian Control Conference (ICCC)*. [s.n.], 2012. p. 776–779. Disponível em: <<https://doi.org/10.1109/CarpathianCC.2012.6228752>>. Citado na página 19.

ZENG, Y. et al. Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Transactions on Intelligent Transportation Systems*, v. 18, n. 6, p. 1647–1653, June 2017. ISSN 1524-9050. Disponível em: <<https://doi.org/10.1109/TITS.2016.2614916>>. Citado 2 vezes nas páginas 1 e 3.

ZHU, Z. et al. Traffic-sign detection and classification in the wild. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2016. p. 2110–2118. Disponível em: <<https://doi.org/10.1109/CVPR.2016.232>>. Citado na página 3.