



# **UNIVERSIDADE FEDERAL DO MARANHÃO**

**Programa de Pós-Graduação em Ciência da Computação**

**JULIO CESAR DA SILVA SOARES**

***METODOLOGIA DE DETECÇÃO E RECONHECIMENTO DE SEMÁFOROS UTILIZANDO REDES NEURAS ARTIFICIAIS***

**São Luís - MA  
2016**

JULIO CESAR DA SILVA SOARES

**METODOLOGIA DE DETECÇÃO E RECONHECIMENTO DE SEMÁFOROS  
UTILIZANDO REDES NEURAS ARTIFICIAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Prof. Dr. Areolino de Almeida Neto (Orientador)  
Universidade Federal do Maranhão – UFMA

São Luís - MA  
2016

FICHA CATALOGRÁFICA  
Serviço de Processamento Técnico da Universidade Federal do Piauí  
Biblioteca Setorial de Floriano

S676m Soares, Julio Cesar da Silva.  
Metodologia de detecção e reconhecimento de semáforos  
utilizando redes neurais artificiais. [Manuscrito] /Júlio Cesar da Silva  
Soares – São Luis - MA, 2016.  
84 fls.

Cópia de computador (printout).  
Dissertação (Mestrado) Programa de Pós – Graduação em Ciência da  
Computação, Universidade Federal do Maranhão - UFMA.  
“Orientação: Prof. Dr. Areolino de Almeida Neto”

1. detecção de semáforos. 2. *Advanced Driver Support Systems*. 3. Histograma retroprojetado. 4. Visão computacional e redes neurais artificiais. I.Título.

CDD 004

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pelo dom e condução da minha vida.

Em seguida aos meus pais pela criação que me foi dada, incentivada ao trabalho, estudo e cidadania. Em especial ao meu pai que Deus o chamou antes de ver o fim desta luta. À minha esposa, Jaqueline Monteiro da Mota Soares, e ao meu filho, Josué Cairú Monteiro da Mota Soares, pelo companheirismo e força.

A minha gratidão aos meus amigos que residem em São Luís, por todo o apoio e o acolhimento.

Agradeço ao meu orientador, prof. Areolino de Almeida Neto e coorientador, prof. Tiago Bonini Borchardt, por todo o conhecimento passado a mim durante todo esse período e ao prof. Anselmo Cardoso de Paiva por todo o apoio concedido.

Ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Maranhão, pela oportunidade de desenvolver este trabalho.

Aos companheiros de mestrado, Emanuel Cleyton Macedo Lemos, Diego Santiago de Meneses Carvalho, Jorge Cordeiro, Gilberto Nunes Neto e Otílio Paulo da Silva Neto, pelo compartilhamento de ideias e companheirismo.

À Fundação de Amparo à Pesquisa do Maranhão, pelo auxílio financeiro concedido à dissertação.

À Universidade Federal do Piauí, instituição da qual faço parte, pela liberação e apoio financeiro.

## RESUMO

As vias urbanas estão cada vez mais complexas e o acréscimo no fluxo de veículos nas cidades de médio e grande porte vem contribuindo para a elevação do número de acidentes. Pesquisas apontam que os sinais de trânsito são eficientes na redução do número de acidentes. A implantação de sinais de trânsito apresentam vantagens relevantes, mas por outro lado revelam alguns problemas, como a dificuldade na detecção de sinais de trânsito pelos condutores em vias urbanas. Este fato está relacionado à quantidade de informações visuais nas vias, ao estresse dos motoristas e/ou à fadiga visual destes, que fazem os motoristas desviarem sua atenção da sinalização. Estas razões motivaram muitas pesquisas nos últimos anos, sobre o tema veículos inteligentes. Assim, o presente trabalho propõe uma metodologia para detectar e reconhecer semáforos de trânsito para ser aplicada em veículos inteligentes, podendo contribuir para os *Advanced Driver Support Systems* - ADAS (Sistema Avançado de Auxílio ao Motorista), e que auxilie os motoristas, em especial aqueles com deficiência parcial da visão. Além disso, o sistema desenvolvido é capaz de identificar o estado do semáforo e indicar ao condutor se ele deve parar ou prosseguir, contribuindo assim para a redução de acidentes de trânsito.

Para o desenvolvimento do algoritmo de detecção, utilizaram-se técnicas de processamento de imagens, através de histograma retroprojetado e limiarização global para detectar pontos de luzes. A limiarização local é aplicada para o cálculo de simetria entre o raio e o centro dos pontos de luzes, com a finalidade de segmentar o corpo do semáforo, onde se obteve uma taxa média de detecção de 99%. As características dos semáforos foram extraídas utilizando os atributos de Haralick, com a inclusão de informações de cor e forma. Os dados gerados pela extração de características foram pré-processados utilizando a técnica de SMOTE para balancear a base de dados. O reconhecimento e a identificação do estado do semáforo foram realizados por uma rede neural artificial do tipo *Multilayer Perceptron* (MLP). No treinamento da rede se utilizou o algoritmo de aprendizagem *backpropagation* e a separação de dados para treinamento e validação. Os resultados da validação mostraram uma taxa média de reconhecimento de 98%.

Palavras-Chave: detecção de semáforos, *Advanced Driver Support Systems*, histograma retroprojetado, visão computacional e redes neurais artificiais.

## **ABSTRACT**

Urban roads are very complex. The increase in the flow of vehicles in the cities has contributed to traffic accidents. Researches for accident reduction show that the traffic lights are effective in reducing accidents. Traffic lights can minimize the occurrence of accidents at intersections and crosswalks. The implementation of traffic light signals shows significant advantages, otherwise reveals some problems such as the failure to detect road signs by drivers on urban roads. This fact is related to excessive visual information, the stress of the drivers and/or eyestrain makes the drivers lose their attention. These reasons motivated researches about intelligent vehicles. This work aims to develop a methodology to detect and recognize traffic lights, to be applied in smart vehicles. This methodology can contribute to the Advanced Driver Support Systems (ADAS), which assists drivers, especially those with partial vision impairment.

Image processing techniques are used to develop the detection methodology. Back project and global thresholding are combined to find light points. Local thresholding techniques are applied to calculate the symmetry between the radius and the center of the light points to segment the traffic light body. The first step got an average rate of 99% of detection. The features of the traffic lights are extracted using Haralick texture measures, with the inclusion of color and shape information. The data generated by the feature extraction step were pre-processed using the SMOTE technique to balance the database. The recognition and identification of the traffic lights state are made by an artificial neural network using Multilayer-Perceptron (MLP). The backpropagation learning algorithm are used in the network training. The validation results show an average recognition rate of 98%.

*Keywords: traffic lights detection, Advanced Driver Support Systems, back project, computer vision, artificial neural network.*

## LISTA DE FIGURAS

FIGURA 1.1 SEMÁFORO DO TIPO CIRCULAR, NA POSIÇÃO VERTICAL .....	17
FIGURA 2.1 EM CINZA OS VIZINHOS DE P NA (A) VIZINHANÇA DE 4, (B) VIZINHANÇA DIAGONAL E (C) VIZINHANÇA DE 8.....	24
FIGURA 2.2 EXEMPLOS DE CONECTIVIDADES: (A) IMAGEM ORIGINAL, (B) CONECTIVIDADE N4(P) E (C) CONECTIVIDADE N8(P).....	25
FIGURA 2.3 EXEMPLO DE DILATAÇÃO.....	26
FIGURA 2.4 EXEMPLO DE EROSIÃO .....	26
FIGURA 2.5 PASSO A PASSO DA ABERTURA E FECHAMENTO .....	27
FIGURA 2.6 EXEMPLO DE HISTOGRAMA COM OITO NÍVEIS DE CINZA .....	28
FIGURA 2.7 MODELO DE COR HSV .....	29
FIGURA 2.8 LIMIAZIZAÇÃO ADAPTATIVA .....	32
FIGURA 2.9 IMAGEM DE ENTRADA, (B) OBJETO MODELO E (C) RESULTADO OBTIDO PELA TÉCNICA DE <i>BACKPROJECT</i> . .....	34
FIGURA 2.10 CLASSIFICAÇÃO SUPERVISIONADA.....	43
FIGURA 2.11 CLASSIFICAÇÃO NÃO SUPERVISIONADA .....	44
FIGURA 2.12 MODELO DE UM NEURÔNIO MCP.....	47
FIGURA 2.13 ARQUITETURA DE UMA REDE NEURAL ARTIFICIAL ORGANIZADA EM CAMADAS ....	49
FIGURA 2.14 TOPOLOGIAS DE RNA.....	50
FIGURA 2.15 GRÁFICO ROC.....	56
FIGURA 2.16 UMA CURVA ROC CRIADA PELA LIMIAZIZAÇÃO.....	57
FIGURA 3.1 ETAPAS DA METODOLOGIA .....	59
FIGURA 3.2 EXEMPLOS DE IMAGENS: (A) IMAGEM ORIGINAL (B) IMAGEM RECORTADA (C) IMAGEM EM HSV.....	60
FIGURA 3.3 LUZES DO SEMÁFORO, (A) LUZ VERDE E (B) LUZ VERMELHA .....	61
FIGURA 3.4 IMAGEM SEGMENTADA .....	61
FIGURA 3.5 IMAGEM RESULTANTE DOS FILTROS MORFOLÓGICOS .....	62
FIGURA 3.6 DETECTANDO SEMÁFORO .....	64
FIGURA 4.1 CURVA ROC.....	75

## LISTA DE TABELAS

TABELA 2.1 EXEMPLO DE HISTOGRAMA .....	28
TABELA 2.2 MATRIZ DO OBJETO.....	34
TABELA 2.3 HISTOGRAMA DA MATRIZ DO OBJETO.....	34
TABELA 2.4 MATRIZ DA IMAGEM DE ENTRADA .....	35
TABELA 2.5 MATRIZ <i>BACKPROJECT</i> .....	35
TABELA 2.6 FORMATO DE UMA GLCM PARA UMA IMAGEM COM 3 NÍVEIS DE CINZA .....	38
TABELA 2.7 EQUAÇÕES DAS FUNÇÕES DE ATIVAÇÃO MAIS COMUNS .....	48
TABELA 2.8 MATRIZ DE CONFUSÃO PARA DUAS CLASSES.....	54
TABELA 2.9 TABELA DE LIMIARES ROC.....	57
TABELA 2.10 QUALIDADE DO CLASSIFICADOR EM RELAÇÃO À ÁREA DA CURVA ROC .....	58
TABELA 4.1 ESTIMATIVAS DE DETECÇÃO. ....	70
TABELA 4.2 DIVISÃO DA BASE DE DADOS .....	71
TABELA 4.3 BALANCEAMENTO POR <i>SMOTE</i> E <i>SPREAD</i> .....	71
TABELA 4.4 DESEMPENHO DO BALANCEAMENTO .....	71
TABELA 4.5 INCLUSÃO DE ATRIBUTOS.....	73
TABELA 4.6 DIVISÃO POR <i>SPLITS</i> .....	73
TABELA 4.7 MATRIZ DE CONFUSÃO - TREINO .....	74
TABELA 4.8 TESTE COM A CAMADA ESCONDIDA.....	74
TABELA 4.9 TABELA DE DADOS DA CURVA ROC.....	74
TABELA 4.10 BALANCEAMENTO <i>SMOTE</i> - VALIDAÇÃO.....	75
TABELA 4.11 MATRIZ DE CONFUSÃO – VALIDAÇÃO.....	76
TABELA 4.12 COMPARAÇÃO DOS RESULTADOS.....	78



## **LISTA DE ABREVIATURAS E SIGLAS**

A	Área
ADAS	<i>Advanced Driver Support Systems</i> (Sistema Avançado de Auxílio ao Motorista)
AUC	<i>Area under the curve</i> (Área sob a curva ROC)
AV	<i>Autonomous Vehicle</i> (Veículo Autônomo)
BP	<i>Backpropagation</i>
C	Circularidade
CamShift	<i>Continuously adaptive mean-Shift</i>
CIE	<i>International Commission on Illumination ou Commission Internationale de L'Eclairage</i>
CIELab	Espaço de cores desenvolvido pela CIE onde L é a luminância, e os demais a cromaticidade
CONTRAN	Conselho Nacional de Trânsito
DENATRAN	Departamento Nacional de Trânsito
ESP	Especificidade
FPR	<i>False Positive Rate</i> (Taxa de falso positivo)
H	<i>Hue</i>
HMM	<i>Hidden Markov Models</i>
HSI	<i>Hue, Saturation and Intensity</i> (Matiz, Saturação e Intensidade)
HSL	<i>Hue, Saturation and Lightness</i> (Matiz, Saturação e Iluminação)
HSV	<i>Hue, Saturation and Value</i> (Matiz, Saturação e Iluminação)
IA	Inteligência Artificial
MCP	Neurônio de McCulloch e Pitts
MLP	<i>Multi-Layer Perceptron</i>
ND	Não detectado
ONU	Organização das Nações Unidas
p	<i>pixels</i>
P	<i>Precision</i> (Precisão)
R	<i>Recall</i>
RBF	<i>Radial Basis Function</i>

RGB	Espaço de cores representado por vermelho, verde e azul ( <i>Red, Green and Blue</i> )
RNA	Rede Neural Artificial
ROC	<i>Receiver Operating Characteristic</i>
ROI	<i>Region of Interest</i> (Região de Interesse)
SEN	Sensibilidade
SMOTE	<i>Synthetic Minority Oversampling Technique</i>
TPR	<i>True Positive Rate</i> (Taxa de verdadeiro positivo)
YCbCr.	Espaço de cores, onde Y é a luminância, Cb é a cromaticidade azul e Cr é a cromaticidade vermelha
YUV	Espaço de cores, onde Y é a luminância, U é a cromaticidade azul e V é a cromaticidade vermelha

## SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	OBJETIVOS .....	17
1.2	TRABALHOS RELACIONADOS .....	17
1.3	ORGANIZAÇÃO DO TRABALHO.....	22
2	FUNDAMENTAÇÃO TEÓRICA .....	23
2.1	IMAGENS DIGITAIS .....	23
2.1.1	VIZINHANÇA .....	23
2.1.2	CONECTIVIDADE.....	24
2.2	PROCESSAMENTO DE IMAGENS DIGITAIS .....	25
2.2.1	PROCESSAMENTO MORFOLÓGICO.....	25
2.2.2	PROCESSAMENTO DE HISTOGRAMA .....	27
2.2.3	MODELOS DE CORES.....	29
2.2.4	CONVERSÃO DE MODELOS CORES: RGB PARA HSV .....	29
2.3	SEGMENTAÇÃO .....	30
2.3.1	MÉTODO DE OTSU.....	31
2.3.2	LIMIARIZAÇÃO ADAPTATIVA OU LOCAL .....	31
2.3.3	SEGMENTAÇÃO POR HISTOGRAMA RETROPROJETADO.....	32
2.4	EXTRAÇÃO DE CARACTERÍSTICAS .....	36
2.4.1	DESCRITORES DE FORMAS.....	36
2.4.2	DESCRITORES DE TEXTURA.....	37
2.4.3	DESCRITORES DE HARALICK.....	38
2.4.4	DESCRITORES COR .....	41
2.5	RECONHECIMENTO DE PADRÕES.....	42
2.5.1	CLASSIFICAÇÃO SUPERVISIONADA .....	43
2.5.2	CLASSIFICAÇÃO NÃO SUPERVISIONADA.....	43
2.6	DESBALANCEAMENTO DE DADOS.....	44
2.6.1	PROBLEMAS COM OS MÉTODOS DE BALANCEAMENTO.....	45

2.6.2	OS MÉTODOS SELECIONADOS.....	46
2.7	REDES NEURAS ARTIFICIAIS .....	46
2.7.1	ARQUITETURAS.....	48
2.7.2	PROCESSOS DE APRENDIZAGEM.....	51
2.7.3	PERCEPTRON DE MÚLTIPLAS CAMADAS .....	51
2.8	ANÁLISE ROC .....	53
2.8.1	DESEMPENHO DOS CLASSIFICADORES.....	54
2.8.2	ESPAÇO ROC.....	55
2.8.3	CURVA NO ESPAÇO ROC .....	56
2.8.4	ÁREA SOB A CURVA ROC.....	58
2.8.5	CONSIDERAÇÕES FINAIS .....	58
3	METODOLOGIA PROPOSTA .....	59
3.1	BASE DE DADOS .....	59
3.2	PREPROCESSAMENTO.....	60
3.3	SEGMENTAÇÃO .....	61
3.4	REDUÇÃO DE FALSOS POSITIVOS.....	62
3.5	EXTRAÇÃO DE CARACTERÍSTICAS .....	64
3.6	BALANCEAMENTO DE DADOS .....	65
3.7	REDES NEURAS ARTIFICIAIS .....	66
3.8	CONSIDERAÇÕES FINAIS.....	66
4	EXPERIMENTOS.....	70
4.1	DETECÇÃO DOS SEMÁFOROS. ....	70
4.2	PREPARAÇÃO DOS DADOS PARA APRENDIZADO .....	70
4.3	INCLUSÃO DE ATRIBUTOS.....	71
4.4	TREINAMENTO DA REDE NEURAL .....	72
4.5	TESTE DE VALIDAÇÃO .....	75
4.6	DISCUSSÃO DOS RESULTADOS .....	76
4.7	COMPARAÇÃO COM TRABALHOS RELACIONADOS .....	78
5	CONCLUSÃO.....	79

5.1	CONTRIBUIÇÕES DESTE TRABALHO .....	80
5.2	TRABALHOS FUTUROS .....	80
6	REFERÊNCIAS BIBLIOGRÁFICAS .....	82

## 1 INTRODUÇÃO

Os cenários do trânsito em vias urbanas estão mais complexos, elevando o número de acidentes. Segundo dados da Organização das Nações Unidas (ONU), em um mês o trânsito brasileiro mata, em média, 4,1 mil pessoas e causa a invalidez permanente de outras 43,2 mil (BRASIL, 2014).

Os números consolidados são mais expressivos que os registrados pela guerra entre palestinos e israelenses, que em julho deixou mais de 1,3 mil mortos e seis mil feridos (BRASIL, 2014). Algumas razões para estes números são:

- o fluxo cada vez maior de carros nas cidades de médio e grande porte;
- a falta de planejamento e manutenção das vias de tráfego;
- e semáforos sem adaptabilidade às variações no dia-a-dia.

A complexidade das cenas nas vias urbanas, o elevado número de informações visuais, o estresse e/ou fadiga visual fazem os motoristas desviarem sua atenção dos sinais de trânsito. Razões estas que motivaram muitas pesquisas nos últimos anos sobre os temas veículos inteligentes e ADAS (*Advanced Driver Assistance Systems* - Sistema Avançado de Auxílio ao Motorista).

Os sinais de trânsito são ferramentas eficientes para redução destes números, de acordo com o manual dos semáforos do Departamento Nacional de Trânsito – DENATRAN (2011). A implantação de semáforos em cruzamento ou em locais onde tem faixas de pedestre, são favoráveis à redução da severidade dos acidentes e resultam na minimização dos números com vítimas fatais (COELHO; FREITAS; MOREIRA, 2008).

De acordo com o Conselho Nacional de Trânsito (CONTRAN) (DENATRAN, 2011), a sinalização semafórica tem por finalidade transmitir aos usuários a informação sobre o direito de passagem em interseções e/ou seções de via onde o espaço viário é disputado por dois ou mais movimentos conflitantes, ou advertir sobre a presença de situações na via que possam comprometer a segurança dos usuários. Levando em conta as diferentes combinações de forma, cor e sinal integrantes da sinalização semafórica, estas possuem significados distintos e transmitem informações específicas ao condutor e pedestre.

Os sistemas avançados de suporte ao motorista devem auxiliar no cumprimento das regras de trânsitos em um ambiente urbano, levando em consideração que devem responder em tempo real, para serem viáveis na aplicação de veículos automotivos (KUO; PAI; LI, 2011). Alguns dos principais sistemas empregados para garantir estes requisitos são: sistema

de posicionamento global (GPS), radares, *scanners*, sensores e visão computacional.

A detecção e identificação dos semáforos em ambientes urbanos complexos, não são atividades triviais. Em geral, as pesquisas feitas para detectar e identificar semáforos levam em conta informações de cor e forma das luzes. Como exemplo, o trabalho de Gong, *et al.* (2010), de Charette e Nashashibi (DE CHARETTE; NASHASHIBI, 2009) e Li, *et al.* (LI *et al.*, 2011) trabalharam com semáforos veiculares de forma circular, enquanto o trabalho de Cai, Li, e Gu (CAI; LI; GU, 2012) é restrito a semáforos em forma de seta.

As pesquisas apresentaram problemas pontuais na detecção de semáforos luminosos em ambiente urbano, como:

- são facilmente confundidos com luzes traseiras de freio de carros, letreiros luminosos de lojas e *outdoors* (WANG *et al.*, 2011);
- tipos de lâmpadas utilizadas nos semáforos, podendo ser do tipo LED ou lâmpadas fluorescentes. As lâmpadas fluorescentes podem apresentar uma intensidade de luz baixa, enquanto as do tipo LED podem ter iluminação parcial (DIAZ-CABRERA; CERRI; MEDICI, 2015);
- a variação das cores das luzes utilizadas pelos semáforos, que além de não serem puras, podem sofrer alterações, como a cor vermelha, que tende a uma cor alaranjada, e a cor verde tende para tons azuis;
- a variação da iluminação e condições climáticas ambientais que influenciam diretamente na detecção da cor dos pontos de luzes. Algumas pesquisas utilizam dois algoritmos para detectar semáforos, um para o dia e outro para o turno da noite. Há outros que não trabalham com imagens e/ou vídeos noturnos, como exemplos os trabalhos de Cai, Li e Gu (CAI; LI; GU, 2012) e Wang *et al.*, (WANG *et al.*, 2011);
- obstrução parcial dos semáforos por objetos e/ou veículos de grande porte, que dificultam na identificação das formas das luzes (WANG *et al.*, 2011);
- o ângulo de visão dos veículos em relação aos semáforos é outro fator de dificuldade (WANG *et al.*, 2011).

Outros fatores considerados na detecção dos semáforos em ambiente urbano são: o modo como são montados (diferentes em cada região), a orientação e a quantidade de semáforos nas vias de tráfego (JOHN *et al.*, 2014).

## 1.1 OBJETIVOS

Esta dissertação de mestrado tem como objetivo desenvolver uma metodologia para detecção e reconhecimento de semáforos, que poderá ser aplicada em veículos inteligentes. Espera-se contribuir para os sistemas ADAS, através da aplicação de técnicas de visão computacional, onde o sistema pode ser resumido em veículo, câmera e algoritmo de detecção de semáforos.

Além disso, este trabalho visa auxiliar motorista com deficiência parcial de visão ou com fadiga visual, trazendo maior segurança e proporcionando conforto na condução dos veículos. Esta segurança será obtida porque a metodologia tem o propósito de identificar o semáforo a uma distância segura e reconhecer o estado deste, indicando ao condutor se ele deve parar ou prosseguir na via. Com isto, espera-se contribuir para a redução do número de acidentes e minimizar as infrações cometidas diariamente pelos motoristas.

Este trabalho limita-se a reconhecer os semáforos do tipo veicular, que tenham formatos circulares e orientação vertical, como pode ser visto na Figura 1.1, levando-se em consideração que estes são formados por luzes: vermelha, amarela e verde. Enfatizando que os veículos devem cumprir as instruções dos semáforos em meio urbano.

Adicionalmente, deseja-se também investigar a eficiência das redes neurais artificiais como abordagem para reconhecimento de padrões, e comparar os seus resultados com aqueles obtidos por outras abordagens presentes na literatura.

Figura 1.1 Semáforo do tipo circular, na posição vertical



Fonte: Adaptada de Sinalização (2014).

## 1.2 TRABALHOS RELACIONADOS

No estado da arte se encontram várias técnicas de reconhecimento de semáforos. Uma delas é baseada em modelos adaptativos de semáforos. De Charette e Nashashibi (2009) trabalham com o casamento de modelos correspondentes. Nesse trabalho algumas



preocupações são citadas como: tempo de processamento relativamente alto; necessidade de criação de modelos para o reconhecimento de novos tipos de semáforo e a dificuldade em estimar a posição do veículo em relação aos semáforos reconhecidos. As taxas de reconhecimento deste trabalho foram de 95,38% de precisão e 98,41% de *recall*. A técnica desenvolvida por Gong, *et al.*, (2010) usou o algoritmo de reconhecimento de semáforos via aprendizado de máquina com as técnicas de Haar e AdaBoost. Nesse trabalho, para evitar falsos positivos e perda de segmento, aplicou-se o rastreamento de alvos através do algoritmo CamShift (*Continuously adaptive mean-Shift*), que usa o histograma do espaço de cor HSV (*Hue, Saturation and Value*, ou Matiz, Saturação e Iluminação) como modelo alvo.

A técnica introduzida por Li *et al.*, (2011) reconhece os semáforos em ambientes urbanos, converte a imagem original de RGB para escala de cinza e aplica o operador *top-hat* resultando em imagens binárias. Para validar regiões circulares detectadas, foram aplicados filtros morfológicos. A identificação da cor das luzes dos semáforos utilizou o modelo de cor HSV e o histograma de cor no canal H. A base de dados utilizada foi dividida em três grupos: imagens em dias nublados, ensolarados e em dias com condições climáticas variadas. Os experimentos apontam que o algoritmo teve bons resultados em condições meteorológicas complexas e a taxa de reconhecimento obtida foi 97% em média.

Outra técnica, desenvolvida por Wang *et al.*, (2011), trabalha com casamento de modelos usando correlação cruzada normalizada. Os autores para chegarem ao reconhecimento, utilizaram limiarização de cores em HSI (*Hue, Saturation and Intensity*, ou Matiz, Saturação e Intensidade) para segmentação da imagem, além disso, utilizaram os operadores morfológicos para remoção de ruídos. Validaram a região candidata de acordo com as propriedades de formas dos semáforos. A taxa de reconhecimento obtida foi em média 97,2% de precisão e 96,5% de *recall*.

Cai, Li e Gu (2012) propuseram um algoritmo para detectar e reconhecer os semáforos do tipo seta. O algoritmo é dividido em dois módulos: detecção e reconhecimento. No módulo detecção, a imagem de entrada em RGB passa pelo processo de binarização, depois são aplicadas à técnica de rotulação e filtragem morfológica para encontrar as regiões de interesse. As regiões encontradas são os retângulos pretos, correspondentes ao corpo do semáforo. Posteriormente essas regiões são convertidas para o formato YCbCr, onde se utiliza o canal Cr para detectar as luzes do tipo seta dos semáforos. No módulo reconhecimento, foram utilizados os algoritmos 2D *Gabor wavelet transform* e 2D Análise de Componentes Independentes para extrair características relacionadas a distribuição de Gauss. Os resultados

obtiveram uma taxa de reconhecimento de 91% em média.

Chen, Shi e Zou (2014) propuseram um sistema unificado de visão que reconhece semáforos do tipo circular e do tipo seta de forma robusta e em tempo real. O algoritmo está dividido em três partes. Na primeira, a imagem de entrada é convertida de RGB para RGB normalizado e modelo de cor HSI; Ainda na primeira parte, são usados o filtro de espaço de cor e o filtro de componentes conectados para obtenção das regiões candidatas. Na segunda parte, é aplicado em cada região candidata o Histograma Orientado a Gradiente em Múltiplas camadas (MHOG), para extrair as características em diferentes escalas e depois se aplica o classificador linear Máquinas de Vetores de Suporte (SVM), treinado para detectar semáforos. Na terceira parte, para cada semáforo detectado, são extraídas as características através do Histograma Orientado a Gradiente (HOG) tradicional para cada luz do semáforo. Para determinar a direção dos semáforos detectados, utilizou-se o classificador SVM linear. Os resultados obtidos foram 99,18% de precisão e 98,96% de *recall*.

O trabalho desenvolvido por John *et al.*, (2014) empregou aprendizagem profunda e redes neurais convolucionais. Esta técnica foi empregada em diferentes condições de iluminação para dia e noite, além de usar um sensor GPS para indicar os semáforos mapeados. Nesse trabalho, aplicaram-se diversos espaços de cores como cinza normalizada, LAB e HSV. Os autores encontraram problemas como identificação de múltiplos semáforos e eliminação das necessidades de obtenção de limiares de cores empíricos. Os resultados apresentaram uma taxa de reconhecimento de 99,4%.

Sathiya, Balasubramanian e Priya (2014) descrevem um método para detectar e reconhecer semáforos e seus intervalos de contagem regressiva. O trabalho é dividido em duas partes: a primeira parte está relacionada à detecção de semáforos que utiliza limiar de cor para o modelo RGB; A segunda parte reconhece os dígitos do *display* de contagem regressiva, encontrado junto ao semáforo. A segmentação através do limiar de cor, extrai os dígitos através de varredura horizontal e vertical, a imagem é redimensionada para um tamanho padrão e convertida para imagens binárias. Através dos estados ligado ou desligado dos segmentos do *display*, o reconhecimento pode ser realizado. O método é capaz de detectar semáforos e temporizadores em tempo razoável. Os resultados dos experimentos apresentam uma precisão de 96% para detecção de semáforos e 88% do reconhecimento dos dígitos do *display*.

Gomez *et al.*, (2014) propõem uma metodologia que liga o processamento de imagens com uma rotina de estimação de estados formada pela *Hidden Markov Models*

(HMM). Este método permite determinar o estado atual do sinal detectado, com base nos estados obtidos por processamento de imagem, com o objetivo de obter o melhor desempenho na determinação dos estados dos semáforos. Para essa solução utilizou-se o algoritmo *Virteb*.

A metodologia é dividida em etapas: inicialmente a imagem de entrada é capturada por uma câmera instalada no veículo autônomo (AV) e em seguida se aplica o processamento de imagens para detectar regiões de interesse, quando nesse momento se aplicam a morfologia gradiente e o *thresholding* Otsu para binarizar a imagem, por fim, através das características geométricas dos semáforos, é realizada a detecção. Depois de detectar o semáforo, é preciso determinar a sua localização. O próximo passo é a determinação do estado do semáforo. Então, o modelo de cor HSV é utilizado para determinar a cor da luz do semáforo. Na etapa de estimação do estado do semáforo, é aplicada a técnica HMM de quatro estados, que representa os estados dos semáforos: vermelho, amarelo, verde e não detectado (ND). Com o método proposto, obteve-se 90,55% de precisão na detecção do estado do semáforo, em comparação com 78,54% obtidos usando apenas a primeira etapa, que compreende o processamento de imagem até detecção.

Diaz-Cabrera, Cerri e Medici (2015) propuseram a criação de um sistema robusto para detectar semáforos suspensos durante o dia e a noite, e estimar a distância em tempo real. O sistema utiliza segmentação de cor com espaços de cores RGB simples e RGB normalizado, depois as amostras são agrupadas em *clusters*, através de processos *fuzzy* combinados com regras sequenciais (uma regra usada para a noite e outra para o dia). As regras sequenciais estão relacionadas à transformada de *Hough* para detectar círculos e à aplicação do detector de bordas *canny* para auxiliar na detecção das bordas dos círculos. Operadores morfológicos e regras de filtros para caixa delimitadora dos semáforos foram utilizados para reduzir a quantidade de falsos positivos. O uso do GPS e de um mapa da região auxiliou na redução de falsas detecções. A estimativa da distância levou em conta o mapeamento de perspectiva inversa juntamente com o filtro de Kalman não linear e o filtro de partículas. Os resultados de detecção durante o dia foram 98,88% de precisão e 97,1 % de *rechamada* e durante a noite foram 99,70% de precisão e 98,93% de *rechamada*.

Almagambetov, Velipasalar e Baitassova (2015) apresentam em seu trabalho um método robusto, com base em padrões de tráfego, e computacionalmente eficiente para detectar os estados dos semáforos, contribuindo para condutores de veículos automotivos, que

são afetados por *Color Vision Deficiency* (CVD)<sup>1</sup>. Nesse trabalho, foram empregados vários mecanismos para reduzir as detecções erradas. O algoritmo utiliza duas câmeras instaladas em um veículo, onde a câmera do lado do motorista captura imagens para serem processadas, enquanto a câmera do lado do passageiro serve como “controle”. Quando as condições de trânsito são verificadas, ambas são sincronizadas. As imagens obtidas utilizam o modelo de cor YUV, onde inicialmente são aplicados os limiares de cores nos canais UV e depois são aplicados operadores morfológicos para eliminar ruídos nas imagens.

Ainda nesse trabalho, filtros de características são aplicados para encontrar as regiões de interesses. Cada região de interesse é analisada utilizando variáveis de rotulação de cor e utilizando o histograma no canal V para determinar a cor das luzes dos semáforos, pois há uma preocupação em não poder diferenciar a cor vermelha da amarela. Como solução, utilizou-se o modelo de cor HSV para assegurar esta diferenciação. O sistema é capaz também de detectar luzes defeituosas de semáforos e semáforos do tipo seta. A detecção pode ser realizada de forma segura a uma distância de 120 metros aproximadamente. A taxa de detecção alcançou 97,5% de precisão (ALMAGAMBETOV; VELIPASALAR; BAITASSOVA, 2015).

Em resumo, as pesquisas estão preocupadas em melhorar a taxa de reconhecimento, além de priorizar alguns fatores, como o tempo de processamento do algoritmo e a estimativa entre a posição do veículo e os semáforos. O tempo de processamento é relevante para aplicações em tempo real, enquanto a estimativa da posição do veículo em relação aos semáforos é interessante para veículos autônomos, para atuação com a velocidade desenvolvida.

Um detector de semáforo instalado em um dispositivo móvel pode funcionar no interior de um automóvel, fixo por um suporte específico em local estratégico, e possibilitar a interação com outros componentes integrados aos dispositivos móveis como sensores e câmera.

O objetivo do presente trabalho foi desenvolver uma metodologia para detectar e reconhecer os semáforos utilizando a visão computacional, fazendo uso de câmeras de vídeo e/ou imagem de baixo custo, montada no interior do veículo, utilizando informações de cor e

---

<sup>1</sup> Deficiência de visão de cores é a incapacidade de distinguir certos tons de cor. O termo "daltonismo" também é usado para descrever esta condição visual.

formas para detecção de pontos luminosos e redes neurais artificiais para realizar o reconhecimento.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte forma, a fim de possibilitar seu melhor entendimento: no Capítulo 2, é abordada a fundamentação teórica, onde estão descritos os temas relevantes para que se tenha o entendimento do trabalho. No Capítulo 3, é comentada a metodologia desenvolvida. O Capítulo 4 descreve os experimentos realizados, e por fim, no Capítulo 5, apresenta-se a conclusão desta pesquisa, com destaque para as contribuições e suas possíveis continuções.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos referentes a um projeto de reconhecimento de padrões, os quais foram selecionados para o desenvolvimento deste trabalho. Os conceitos abrangentes são: imagens digitais, processamento de imagens, segmentação de imagens, extração de características, reconhecimento de padrões, redes neurais e finalizando com teste e validação de classificadores.

Além destes conceitos foi feita uma breve introdução sobre balanceamento de dados e as influências no desempenho dos classificadores.

### 2.1 IMAGENS DIGITAIS

Uma imagem pode ser descrita matematicamente por uma função  $f(x,y)$  da intensidade luminosa, sendo seu valor, em qualquer ponto de coordenadas espaciais  $(x,y)$ , proporcional ao brilho (ou nível de cinza) da imagem naquele ponto (FILHO; NETO, 1999).

A imagem ao ser digitalizada deve ser representada por uma estrutura de dados, sendo muito comum a representação por uma matriz  $N \times M$ , onde  $N$  e  $M$  são valores inteiros, como pode ser observado na Equação 2.1 (FILHO; NETO, 1999),

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,0) & \dots & f(0,0) \\ f(1,0) & f(1,0) & \dots & f(1,0) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (2.1)$$

A função  $f(x, y)$  que representa uma imagem digital possui diversos elementos na matriz e cada elemento dessa matriz representa um elemento da imagem, ou seja, um ponto da imagem, chamado de *pixel*. Em uma imagem digitalizada, cada elemento pode assumir um valor no conjunto  $\{0, 1, \dots, L-1\}$ , onde  $L$  é a quantidade possível de níveis de cinza, normalmente uma potência de dois, sendo que o valor 0 corresponde à cor preta e o valor  $L-1$  corresponde à cor branca.

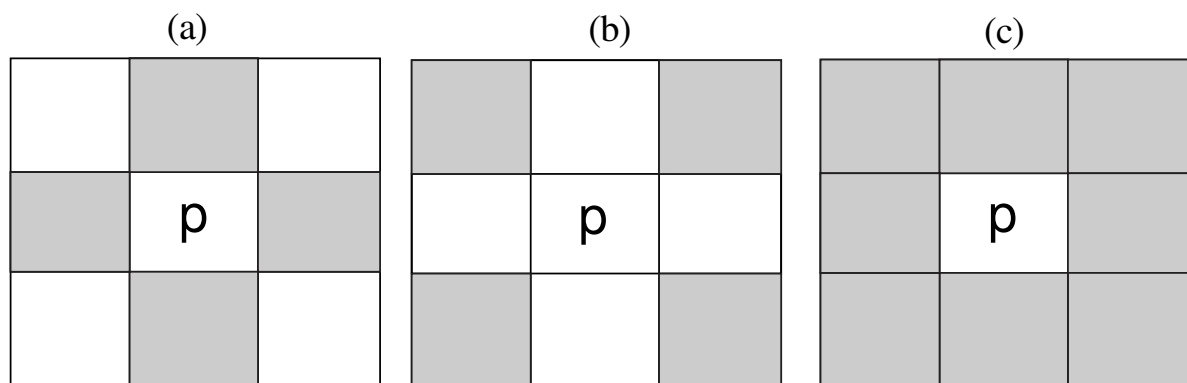
#### 2.1.1 VIZINHANÇA

Nas coordenadas  $(x, y)$  de uma imagem digital, um *pixel*  $p$  possui dois vizinhos horizontais  $\{(x+1, y), (x-1, y)\}$  e dois verticais  $\{(x, y+1), (x, y-1)\}$ , que formam o conjunto

conhecido como vizinhança de 4 do *pixel*  $p$ , representado por  $\{N4(p)\}$ . O *pixel*  $p$  também possui quatro vizinhos nas diagonais  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$ ,  $(x-1, y-1)$ , representados por  $ND(p)$ . A união de  $N4(p)$  e  $ND(p)$  compõe a vizinhança de 8 ( $N8(p)$ ) do *pixel*  $p$ . Na Figura 2.1 se ilustram as vizinhanças de  $p$ .

É importante ressaltar que em certas posições, como nas bordas das imagens, por exemplo, o *pixel*  $p$  não terá todos os seus vizinhos. Ao trabalhar com algumas técnicas de processamento de imagens, deve-se fazer alguns ajustes para obter um resultado uniforme (GONZALEZ; WOODS, 2000).

Figura 2.1 Em cinza os vizinhos de  $p$  na (a) vizinhança de 4, (b) vizinhança diagonal e (c) vizinhança de 8.



Fonte: Adaptada de Marques Filho e Neto (1999).

### 2.1.2 CONECTIVIDADE

A conectividade de *pixels* permite dizer quais *pixels* podem ser ditos conectados uns aos outros, sendo importante na localização de regiões semelhantes. Para dizer que dois *pixels*  $p1$  e  $p2$  são conectados, eles necessitam satisfazer os seguintes critérios (GONZALEZ; WOODS, 2000):

1. Ser adjacente, ou seja, devem estar em  $N8(p)$  ou  $N4(p)$  e;
2. Atender critérios de classificação, seus níveis de cinza devem ser iguais.

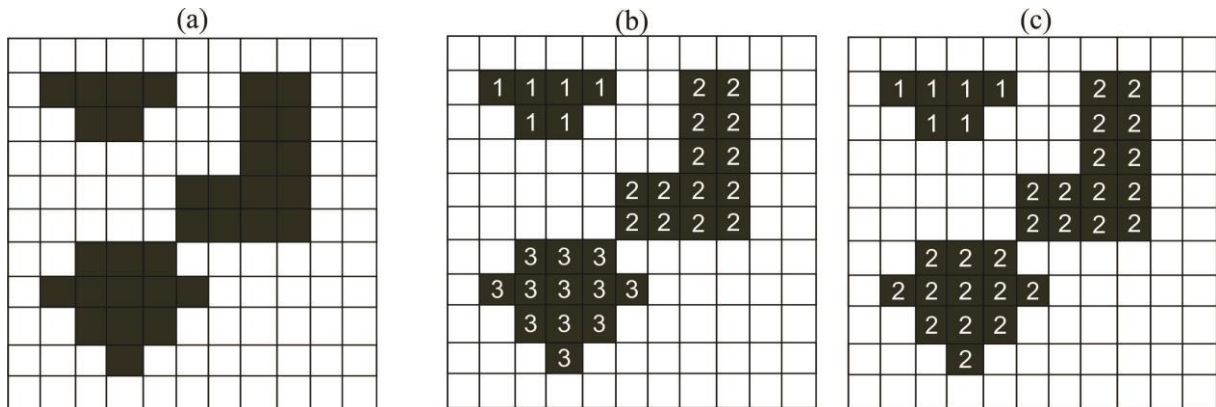
Tal conceito é importante para estabelecer bordas de componentes e objetos.

Cada subconjunto de *pixels* conectados em uma imagem é chamado de componente conexo. Na seção seguinte são mostrados alguns exemplos.

Através da conectividade, pode-se fazer a rotulação de componentes conexos. Esta técnica consiste em atribuir um valor inteiro distinto diferente de zero para cada subconjunto de *pixels* conexos de uma imagem binária. Dessa forma cada componente é separado facilmente da imagem. A Figura 2.2 ilustra a aplicação do método de rotulação de componentes conexos, onde se tem exemplos de componentes conexos em (b) com

vizinhança de 4 e (c) com vizinhança de 8.

Figura 2.2 Exemplos de conectividades: (a) imagem original, (b) conectividade N4(p) e (c) conectividade N8(p)



Fonte: Adaptado de Pedrini e Schwartz (2008).

## 2.2 PROCESSAMENTO DE IMAGENS DIGITAIS

O processamento de imagens digitais abordado nesta subseção está relacionado às operações morfológicas, operações de histograma, modelos de cores e finalizando com a conversão de cor RGB para HSV.

### 2.2.1 PROCESSAMENTO MORFOLÓGICO

Chamam-se de processamento morfológico as técnicas que realizam transformações sobre imagens que podem resultar em filtragens, afinamento, junção ou separação de elementos (regiões que têm formas, apresentadas nas imagens). Sua teoria vem da morfologia matemática, que realiza estudos sobre formas de objetos utilizando a teoria de conjuntos.

As operações morfológicas foram originalmente desenvolvidas para imagens binárias, mas também existem técnicas para operações sobre imagens em tons de cinza e imagens coloridas.

Existem diversas operações morfológicas, sendo base para muitas delas a erosão e a dilatação. Em definição matemática formal, considerando  $A$  e  $B$  como conjuntos no espaço bidimensional  $Z^2$  (espaço que representa imagens binárias) e  $\emptyset$  como conjunto vazio, a dilatação é definida pela Equação 2.2 e a erosão, pela Equação 2.3.

$$A \oplus B = \{Z | (B)_z \cap A^c = \emptyset\} \quad (2.2)$$

$$A \ominus B = \{Z | (B)_z \subseteq A\} \quad (2.3)$$

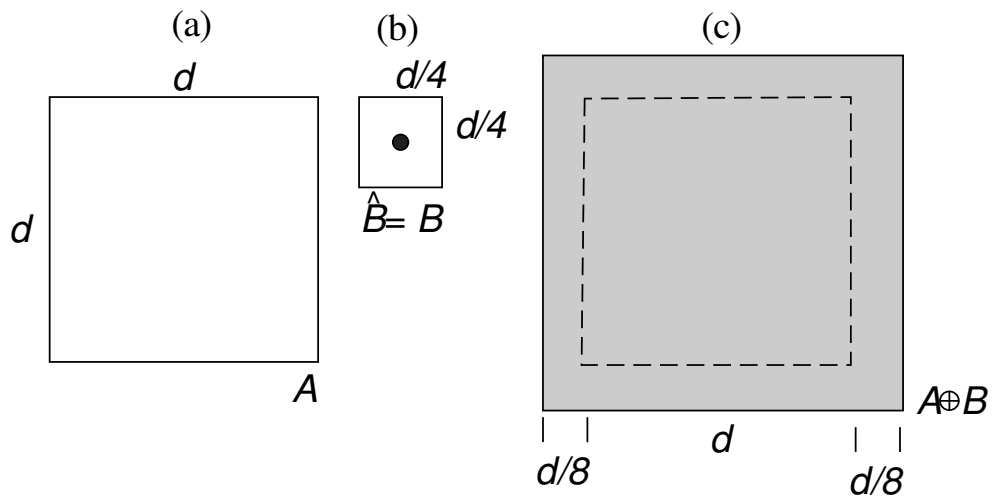
Na Figura 2.3, têm-se o conjunto original (a), o elemento estruturante quadrado (b) e



o resultado da dilatação (c). Como se pode perceber, a operação do elemento estruturante (b) no conjunto original (a) resulta na expansão da parte cinza em (c), que é representado pela dilatação  $A$  por  $B$ .

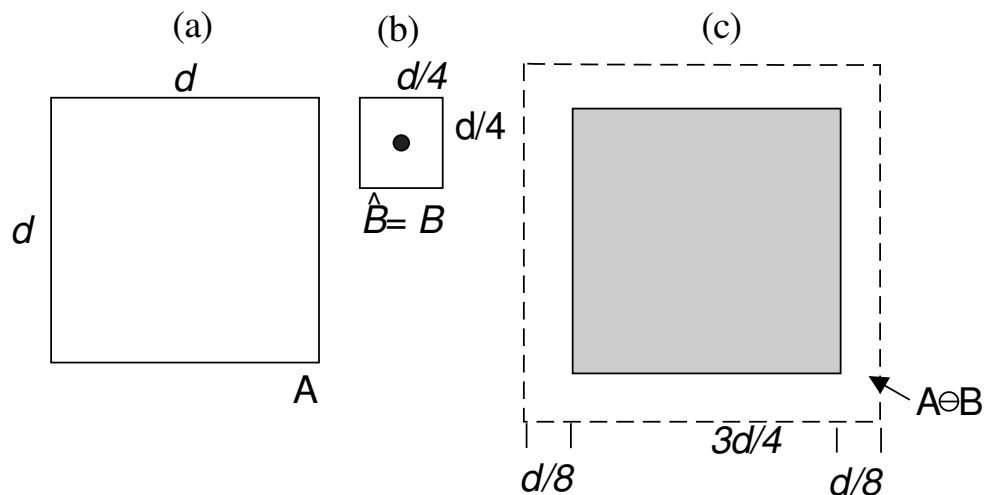
Na Figura 2.4, têm-se o conjunto original (a), o elemento estruturante quadrado (b) e o resultado da operação de erosão (c). Como se pode perceber, a operação do elemento estruturante (b) no conjunto original (a) resulta na redução da parte cinza em (c), que é representado por erosão de  $A$  por  $B$ .

Figura 2.3 Exemplo de dilatação



Fonte: Adaptada de Gonzalez e Woods (2000).

Figura 2.4 Exemplo de erosão



Fonte: Adaptada de Gonzalez e Woods (2000).

Em imagens binárias, a operação de abertura do conjunto  $A$  por um elemento estruturante  $B$  é definida pela Equação 2.4,

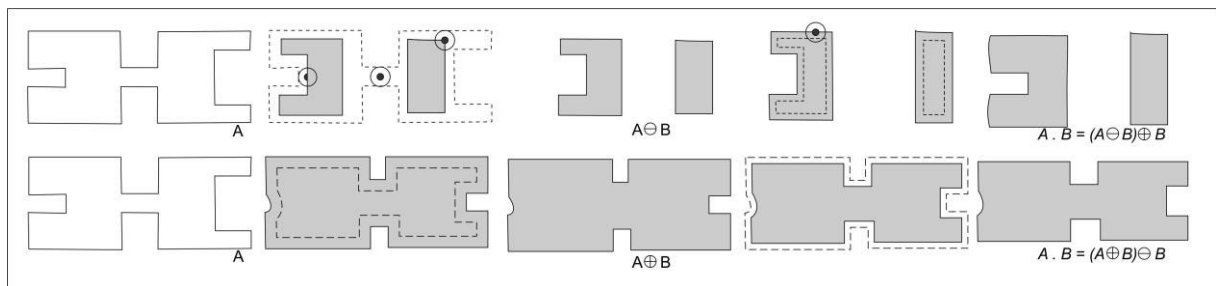
$$A \circ B = (A \ominus B) \oplus B \quad (2.4)$$

Isto é, aplica-se erosão, após, a dilatação. Como resultado, obtêm-se a suavização de contornos, a formação de ilhas e a eliminação de pontos pequenos. Já na operação de fechamento em imagens binárias do conjunto  $A$  por um elemento estruturante  $B$ , utiliza-se como definição a Equação 2.5:

$$A \cdot B = (A \oplus B) \ominus B \quad (2.5)$$

Neste caso, aplica-se dilatação, em seguida, erosão. Com isso, é possível conectar pequenas regiões, suavizar contornos e eliminar buracos. Pode-se ver na Figura 2.5 o passo a passo dos operadores de abertura e fechamento.

Figura 2.5 Passo a passo da abertura e fechamento



Fonte: Adaptada de Gonzalez e Woods (2000).

Na Figura 2.5, verifica-se na linha superior um exemplo de uma operação de abertura, enquanto na parte inferior um exemplo de fechamento para o mesmo objeto original e o mesmo elemento estruturante.

## 2.2.2 PROCESSAMENTO DE HISTOGRAMA

O Histograma de uma imagem representa a distribuição estatística entre os níveis de cinza e a quantidade (ou frequência) de *pixels* em uma imagem. Do ponto de vista estatístico, o histograma de uma imagem digital representa a função de distribuição de probabilidade (FDP) dos níveis de cinza presentes na imagem.

Através da visualização do histograma de uma imagem, pode-se obter uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura) (FILHO; NETO, 1999).

O histograma de uma imagem é obtido através da Equação 2.6:

$$P_r(r_k) = \frac{n_k}{n} \quad (2.6)$$

onde:

- $k = 0, 1, \dots, L-1$ ,  $L$  é o número de níveis de cinza da imagem, geralmente  $L = 256$ ;
- $n$  é o número total de *pixel* da imagem;
- $P_r(r_k)$  é a probabilidade do  $k$ -ésimo nível de cinza;
- $n_k$  é o número de *pixels* cujo nível de cinza corresponde a  $k$ ;
- $r_k$  é o valor de intensidade dos *pixels* que varia de  $0 \leq r_k \leq 1$ .
- 0 representa a parte mais escura da imagem e;
- 1 representa a parte mais clara da imagem.

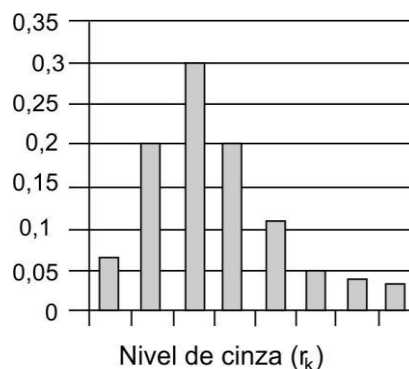
Um histograma apresenta várias características importantes. A primeira delas é que cada  $P_r(r_k)$  fornece, como sugere a notação, a probabilidade de um *pixel* da imagem apresentar nível de cinza  $r_k$ . Portanto, um histograma é uma função de distribuição de probabilidades e, como tal, deve obedecer aos axiomas e teoremas da teoria de probabilidade. Por exemplo, pode-se ver que na Tabela 2.1, a soma dos valores de  $P_r(r_k)$  é 1.

Tabela 2.1 Exemplo de histograma

Nível de cinza ( $r_k$ )	$n_k$	$P_r(r_k)$
0	1120	0,068
1/7	3214	0,196
2/7	4850	0,296
3/7	3425	0,209
4/7	1995	0,122
5/7	784	0,048
6/7	541	0,033
1	455	0,028
Total	16384	1

Fonte: Adaptada de Marques Filho e Neto (1999).

Figura 2.6 Exemplo de histograma com oito níveis de cinza



Fonte: Adaptada de Marques Filho e Neto (1999).

Entre as principais aplicações dos histogramas estão a melhoria de contraste de uma

imagem, compressão, segmentação ou ainda a descrição dessa imagem.

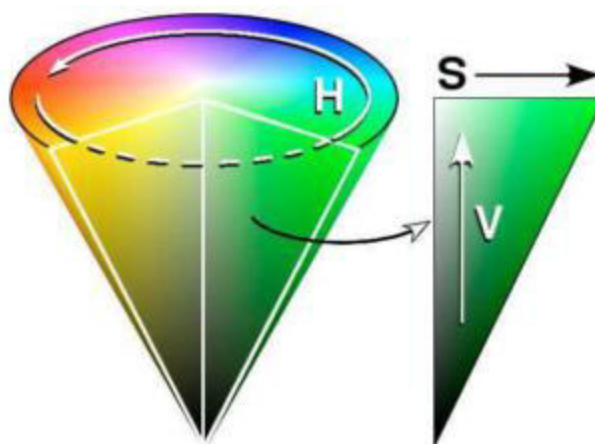
### 2.2.3 MODELOS DE CORES

O objetivo dos modelos de cores é facilitar a especificação das cores em alguma forma padronizada, amplamente aceita. Essencialmente, um modelo de cores é uma especificação de um sistema no qual cada cor é representada por um único ponto.

A maioria dos modelos de cores utilizados atualmente é orientada ou em direção ao hardware (como no caso dos monitores e impressoras coloridas) ou em direção a aplicações envolvendo a manipulação de cores (como a criação de imagens coloridas para uma animação).

Em termos de processamento digital de imagens, os modelos de cores mais comuns são: RGB e HSI (GONZALEZ; WOODS, 2000). O modelo de cor HSI ou HSV é representado por três componentes ( $H$ ,  $S$  e  $V$ ). O primeiro componente  $H$  representa a matiz, responsável pelo tipo de cor pura, enquanto que o componente  $S$  representa a saturação, que fornece uma medida do grau que uma cor pura é diluída pela luz branca. O último componente  $V$  está associado à iluminação, que representa o brilho ao longo do eixo de cinza (por exemplo, do branco ao preto).

Figura 2.7 Modelo de cor HSV



Fonte: Adaptada de Gonzalez e Woods (2000).

### 2.2.4 CONVERSÃO DE MODELOS CORES: RGB PARA HSV

Seja uma cor definida por RGB, onde  $R$ ,  $G$  e  $B$  normalizados entre 0 e 1. Sejam  $MAX$  e  $MIN$  os valores correspondentes, respectivamente, ao maior e ao menor valor presente em todos os canais. A transformação para os parâmetros  $H$ ,  $S$  e  $V$  dessa cor pode ser determinada

pelas Equações 2.7 a 2.9.

$$H = \begin{cases} 60 * \frac{G - B}{MAX - MIN} + 0, se MAX = R e G \geq B \\ 60 * \frac{G - B}{MAX - MIN} + 360, se MAX = R e G < B \\ 60 * \frac{G - B}{MAX - MIN} + 120, se MAX = G \\ 60 * \frac{G - B}{MAX - MIN} + 240, se MAX = B \end{cases} \quad (2.7)$$

$$S = \frac{MAX - MIN}{MAX} \quad (2.8)$$

$$V = MAX \quad (2.9)$$

### 2.3 SEGMENTAÇÃO

Segmentação pode ser definida como a simplificação da imagem, reduzindo-a em seus componentes básicos, ou objetos, sem descartar suas características relevantes (BAXES, 1994). A definição de “características relevantes” depende do requisito particular da imagem. Uma operação de segmentação é qualquer operação que destaca, ou de alguma forma isola, objetos individuais dentro de uma determinada imagem. A segmentação subdivide uma imagem em seus elementos ou objetos, especificando que o nível ao qual cada uma destas subdivisões é levada, depende do problema que é resolvido, isto é, a segmentação deve parar quando os objetos de interesse em uma aplicação tiverem sido isolados (GONZALEZ; WOODS, 2000).

Algoritmos de segmentação para imagens monocromáticas geralmente são baseados em uma ou duas propriedades básicas de nível de cinza: descontinuidade e similaridade. Na primeira categoria, a estratégia consiste em dividir uma imagem baseada nas grandes mudanças do nível de cinza, e as principais áreas de interesse desta categoria são descobertas de pontos isolados e descoberta de linhas e bordas em uma imagem. As principais estratégias na segunda categoria estão baseadas na divisão da imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios predefinidos. A limiarização, crescimento de região e a divisão e fusão de regiões são exemplos dos métodos dessa categoria. O conceito de segmentação de imagem baseado na descontinuidade ou similaridade dos valores de nível de cinza dos *pixels* é aplicável tanto em imagens estáticas quanto dinâmicas (GONZALEZ; WOODS, 2000).

### 2.3.1 MÉTODO DE OTSU

O método de Otsu (1979) é considerado um eficiente método de binarização automática de imagens. O método assume que a imagem está dividida em duas classes, objeto e fundo, e tenta encontrar o limiar que melhor divide essas classes por meio de múltiplas iterações.

Para isso, calcula-se a variância entre as classes objeto e fundo para todos os possíveis limiares. O primeiro passo para implementar o método é computar o histograma  $h(G_i)$  da imagem que será binarizada, onde  $h$  representa o histograma e  $G_i$  os níveis de cinza.

As Equações 2.10, 2.11, 2.12, 2.13 e 2.14 são calculadas para cada valor de  $K$  variando de 1 até  $G_{max}-1$ . Nas equações abaixo,  $G_{max}$  representa o nível máximo de cinza na imagem e  $K$  representa o nível de cinza que particiona o histograma da imagem em duas classes  $C_1$  e  $C_2$ . Assim,  $P_1(K)$  é a probabilidade do nível de cinza  $k$  ser da classe  $c_1$

$$P_1(K) \left( \sum_{i=0}^K h(G_i) \right) \quad (2.10)$$

E  $P_2(K)$  é a probabilidade do nível de cinza  $k$  ser da classe  $c_2$

$$P_2(k) \left( \sum_{i=K+1}^{G_{max}-1} h(G_i) \right) \quad (2.11)$$

O histograma é aproximado por duas funções Gaussianas ( $m_1$  e  $m_2$ ):

$$m_1 \left( \frac{1}{P_1(k)} \sum_{i=0}^K i, h(G_i) \right) \quad (2.12)$$

$$m_2 \left( \frac{1}{P_2(k)} \sum_{i=K+1}^{G_{max}-1} i, h(G_i) \right) \quad (2.13)$$

O valor do limiar, que representa a máxima variância ( $\sigma$ ) entre as duas classes  $C_1$  e  $C_2$ , é calculado pela Equação 2.14 e é escolhido como o valor ótimo de "Threshold". Então, determina-se a variância ( $\sigma$ ) intraclasses em relação ao nível de cinza  $k$  por:

$$\sigma^2 = \frac{[m_1(k) \cdot P_2(k) - m_2(k) \cdot P_1(k)]^2}{P_1(k) \cdot P_2(k)} \quad (2.14)$$

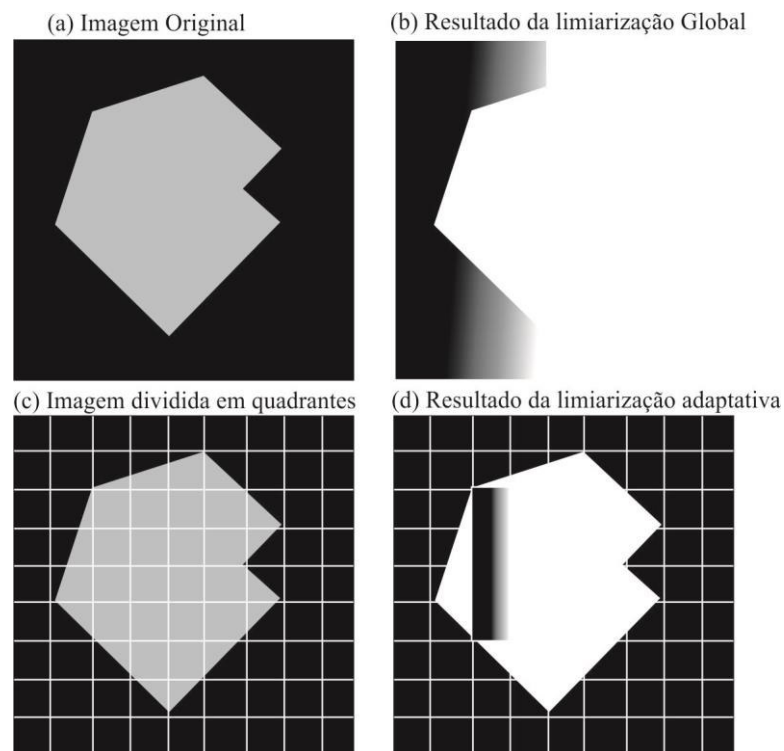
### 2.3.2 LIMIAZIZAÇÃO ADAPTATIVA OU LOCAL

A técnica de limiarização adaptativa ou local consiste em aplicar limiarização global

simples sobre regiões menores e, desta forma, obter limiares mais precisos. De acordo com a Figura 2.8, a imagem original foi dividida em quatro quadrantes e cada um novamente dividido em outros quatro quadrantes. O resultado da técnica de limiarização adaptativa aplicada sobre cada região pode ser visto na Figura 2.8 (d). Resultados ainda melhores podem ser obtidos no exemplo, caso sejam aplicadas novas divisões nas regiões centrais, que não foram bem-sucedidas em sua limiarização.

A técnica de limiares locais apresenta bons resultados, porém segundo Vaz (2011), a maior dificuldade dos limiares locais é a determinação do tamanho da janela para definição do limiar local. Talvez por isso existam várias técnicas para determinar o tamanho da janela (BAGGIO, DANIEL LÉLIS; ESCRIVÁ, DAVID MILLÁN; MAHMOOD, NAUREEN; SHILKROT, ROY; EMAMI, SHERVIN; LEVGEN, KHVEDCHENIA; SARAGIH, 2012).

Figura 2.8 Limiarização adaptativa



Fonte: Adaptada de Vaz (2011).

### 2.3.3 SEGMENTAÇÃO POR HISTOGRAMA RETROPROJETADO

*Backproject* ou retroprojeção de histograma da imagem é usado em muitos domínios associados à extração de características de cor. A partir de uma imagem em um canal de cor é obtido o histograma *backproject* desta imagem, que resulta na modificação do histograma da imagem original, que redistribui os *pixels* através da probabilidade de que os *pixels* buscados pertençam aos *pixels* dos objetos de interesse (KHAN; FARBIZ, 2010).

O conceito de histograma retropropagado é construído sobre o pressuposto de que o histograma do objeto ( $O$ ) tem faixa estreita e o histograma da imagem ( $M$ ) é amplamente distribuído, isto é, estreita refere-se a histogramas de um simples objeto, enquanto amplo seria o histograma de uma imagem com este objeto e com outros também.

A retroprojeção tem a finalidade de utilizar o histograma do objeto ( $O$ ) para identificar em imagem de entrada os *pixels* que podem pertencer ao objeto em questão. Para isso é aplicado à distribuição acumulativa (*Cumulative Distribution Function* - CDF), que pode ser vista nas Equações 2.15 e 2.16.

$$cdf_x(i) = \sum_{j=0}^i P_x(j) \quad (2.15)$$

$$k = \frac{L}{i} \quad (2.16)$$

onde  $k$  é a constante que resulta da quantidade de níveis de cinza ( $L$ ) da imagem dividido pela quantidade de *bins*, representada por  $i$ , (*bins* - predefinição do número de intervalos em que os valores da imagem serão representados).

A redistribuição é realizada de acordo com a quantidade de *bins*:

$$cdf_y(i) = iK \quad (2.17)$$

$$cdf_y(y') = cdf_y(T(k)) = cdf_x(k) \quad (2.18)$$

A função  $T$  mapeia a intensidade de cada *pixel* para o intervalo de limiares, de acordo com a quantidade de *bins* predefinida por  $i$ ,  $y'$  são valores de *pixel* que provavelmente pertençam aos *pixels* do objeto. O resultado desta redistribuição deve ser normalizado de acordo com o histograma do objeto, conforme as Equações 2.19 e 2.20,

$$n = \max\{y'\} = L \quad (2.19)$$

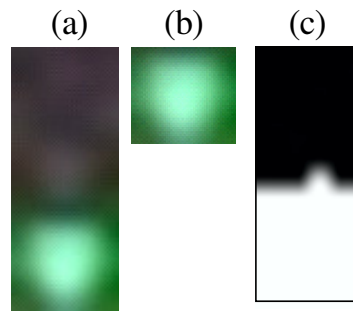
$$x'.n = y'.L \quad (2.20)$$

onde  $n$  representa o máximo entre os  $y'$ , na normalização este recebe a quantidade máxima de nível de cinza e os demais valores  $x'$ , são normalizados, utilizando regra de três simples, apresentado na Equação 2.20.

Abaixo um exemplo prático da aplicação do algoritmo *backproject* para reconhecimento de um semáforo com estado verde, utilizando o modelo de cor *HSV* e o histograma do canal de cor  $H$  (*Hue*).



Figura 2.9 Imagem de entrada, (b) objeto modelo e (c) resultado obtido pela técnica de *Backproject*.



Fonte: Elaborada pelo autor.

Neste exemplo, inicialmente é apresentada ao algoritmo uma imagem de entrada e um objeto modelo, que se deseja localizar nessa imagem, de acordo com as Figura 2.9 (a) e (b), respectivamente.

A matriz da Tabela 2.2 representa os valores de intensidade dos *pixels* no canal *H* do objeto. Para este exemplo, foram definidos 5 *bins* e o canal *H* desse modelo de cor varia de 0 a 180 tons. O resultado dessa primeira parte deste exemplo pode ser visto nessa tabela.

Tabela 2.2 Matriz do objeto.

71	73	76	75	71	68
71	72	76	75	70	67
70	71	75	74	70	67
67	69	74	73	68	64
61	65	72	70	63	59

Tabela 2.3 Histograma da matriz do objeto.

Distribuição de <i>bins</i>						
Bins	1	2	3	4	5	
Limiares	35	71	107	143	180	
CDF	0	19	11	0	0	
Normalização						
Valores normalizados	0	180	104,2105	0	0	
						Min
						Max
						0
						89

Depois da normalização, é executada a segunda parte do algoritmo, que utiliza a tabela de distribuição como referência para buscar os valores que fazem parte do intervalo dos *bins* da Tabela 2.3.

Na Tabela 2.3, pode-se verificar que cada *bin* tem um intervalo de intensidade de *pixel* de acordo com a Equação 2.17. Para o exemplo, estes valores podem ser vistos abaixo:

- *bins* (1) trabalha com um intervalo ( $0 \leq \text{bins} (1) < 36$ );
- *bins* (2) trabalha com um intervalo ( $36 \leq \text{bins} (1) < 72$ );
- *bins* (3) trabalha com um intervalo ( $72 \leq \text{bins} (1) < 108$ );

- *bins* (4) trabalha com um intervalo ( $108 \leq \textit{bins} (1) < 144$ );
- *bins* (5) trabalha com um intervalo ( $144 \leq \textit{bins} (1) < 180$ ).

O resultado da primeira etapa encontrou apenas *pixels* que estão nos intervalos dos *bins* 2 e 3. *Min* e *Max* representam, respectivamente, a menor e a maior quantidade cumulativa de *pixels* encontrados no histograma do objeto. A normalização utiliza os valores mínimo e máximo para normalizar os valores dos *bins*, que resultaram nos valores 104 e 180, aproximadamente, para os *bins* 2 e 3, respectivamente.

A retroprojeção é aplicada na matriz da imagem de entrada, representada pela Tabela 2.4, buscando apenas os valores que estão dentro dos intervalos citados. O resultado pode ser visualizado na Tabela 2.5, onde os valores da matriz são substituídos pelos valores normalizados dos *bins* do modelo e os demais valores que não correspondem aos intervalos são substituídos por zeros.

Tabela 2.4 Matriz da imagem de entrada

163	164	159	157	150	144
163	164	159	157	150	147
163	164	159	159	155	155
163	163	158	158	156	156
170	170	160	156	160	170
3	177	160	156	0	11
5	0	169	170	5	11
12	13	20	36	26	25
35	46	70	72	50	46
60	67	75	74	64	58
70	73	77	76	69	66
71	73	77	76	70	67
70	73	77	76	69	65
65	70	76	75	67	62
52	60	74	72	58	52

Tabela 2.5 Matriz *backproject*

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	180	0	0
0	180	180	104	180	180
180	180	104	104	180	180
180	104	104	104	180	180
180	104	104	104	180	180
180	104	104	104	180	180
180	180	104	104	180	180
180	180	104	104	180	180

Ao final deste processo, tem-se a imagem que pode ser visualizada na Figura 2.9 (c).

## 2.4 EXTRAÇÃO DE CARACTERÍSTICAS

Uma vez que a imagem é segmentada, o próximo passo para a análise das características é buscar as informações individuais de cada objeto, que são obtidas com base no uso de descritores. Com estas medidas é possível comparar as informações dos objetos com medidas previamente conhecidas para enfim classificar o objeto em uma das várias categorias (BAXES, 1994). Existem vários tipos de descritores capazes de descrever as características de um objeto, como descritores de forma, textura ou cor.

### 2.4.1 DESCRITORES DE FORMAS

Segundo Pitas (2000), descrição de forma é um problema importante tanto em análise quanto em síntese de imagens. A representação da forma obtida por técnicas de análise de imagens pode ser usada para aplicações de reconhecimento de objetos, em aplicações gráficas e representações de objetos bi e tridimensionais. Elas podem ser obtidas através de modelagem do objeto e são usadas para a síntese de imagens digitais. As representações devem descrever vários esquemas de forma bidimensional e suas aplicações ocorrem em análise de imagens, visão computacional e reconhecimento de padrões.

Formas bidimensionais podem ser descritas de duas maneiras diferentes: o primeiro método visa usar o contorno do objeto e suas características (como o tamanho do contorno, curvatura, assinatura, descritores de Fourier, etc.), e o segundo visa descrever a região ocupada pelo objeto no plano da imagem. O primeiro método está diretamente ligado à detecção de linhas e bordas e os esquemas descritores resultantes são chamados de representações externas. Estes produzem uma compacta representação da forma. O segundo método é conectado às técnicas de segmentação de região. Os esquemas de segmentação conduzem diretamente para esquemas de representação do objeto. Descritores de região (como área, momentos e esqueletização) são usados extensivamente em aplicações de reconhecimento de objetos.

A área de uma região é definida como o número de *pixels* na região. O perímetro de uma região é determinado pelo tamanho (comprimento) da sua fronteira. Embora a área e o perímetro sejam por vezes utilizados como descritores, eles são usados principalmente em situações em que o tamanho das regiões de interesse é invariante. A utilização mais frequente desses dois indicadores é na medida da capacidade de uma região, definida como

(perímetro)<sup>2</sup>/área. Um descritor um pouco diferente (por conta de um multiplicador escalar) da compacidade é a razão de circularidade, definida como a razão entre a área de uma região e área de um círculo (a forma mais compacta) tendo o mesmo perímetro. A área  $A$  de um círculo com perímetro  $P$  é  $P^2/4\pi$ . Portanto, a razão de circularidade  $R_c$  é dada pela Equação 2.21 (GONZALEZ; WOODS, 2000).

$$R_c = \frac{4\pi A}{P^2} \quad (2.21)$$

#### 2.4.2 DESCRITORES DE TEXTURA

A textura constitui uma característica diretamente relacionada com as propriedades físicas que a superfície de um objeto apresenta (lisa, rugosa, macia, áspera ou ondulada). Ela descreve o padrão de variação de tons de cinza ou cor numa determinada área. Trata-se de um termo intuitivo e de grandes utilidades, mas que, apesar de sua grande importância, não possui uma definição precisa. Uma textura caracteriza-se pela repetição de um modelo sobre uma região, sendo este modelo repetido em sua forma exata ou com pequenas variações. Através de sua análise é possível distinguir regiões que apresentem as mesmas características de reflectância, e, portanto, mesmas cores em determinada combinação de bandas. Isso torna a textura um excelente descritor regional, contribuindo para uma melhor precisão dos processos de reconhecimento, descrição e classificação de imagens. (SANTOS, 2009) (HARALICK; SHANMUGAM; DINSTEN, 1973).

As três principais abordagens usadas no processamento de imagens para descrever a textura de uma região são (JAIN, 1989):

- **Estatística:** utiliza um conjunto de medidas estatísticas locais para representar a textura de uma imagem, como suave, rugosa, granulada e assim por diante. Métodos estatísticos produzem melhores resultados quando aplicados sobre texturas naturais.
- **Estrutural:** as regiões da textura são consideradas como padrões bidimensionais compostos por primitivas que se relacionam. Um exemplo básico seriam linhas paralelas espaçadas regularmente.
- **Espectral:** baseada em propriedades de espectros de frequência, obtidos através de transformadas tais como Fourier e Wavelets. São usadas principalmente para detectar a periodicidade global em uma imagem pela identificação de picos de alta energia no espectro.

### 2.4.3 DESCRITORES DE HARALICK

Haralick (1979) propôs uma metodologia para descrição de texturas baseada em estatística de segunda ordem, onde são definidas características provenientes do cálculo de matrizes em níveis de cinza, denominadas Gray Level Co-occurrence Matrix (GLCM) ou “matrizes de coocorrência em nível de cinza”. A matriz de coocorrência consiste de uma contagem de quantas combinações diferentes de níveis de cinza ocorrem em uma imagem, em uma determinada direção. Para obtenção de tais matrizes, considera-se a variação da distância ( $d$ ) e direção ( $\theta$ ), entre *pixels* vizinhos. Normalmente são utilizados quatro direcionamentos:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  e  $135^\circ$ .

Supondo uma imagem com apenas três níveis de cinza (0, 1 ou 2), a GLCM referente a ela teria o formato apresentado na Tabela 2.6, sendo que  $V_{i,j}$  indica quantas ocorrências de *pixels* de referência com nível de cinza  $i$ , cujo *pixel* vizinho apresenta nível de cinza  $j$ , há na imagem.

Tabela 2.6 Formato de uma GLCM para uma imagem com 3 níveis de cinza

		Valor do pixel vizinho		
		0	1	2
Valor do pixel de referência	0	V(0,0)	V(0,1)	V(0,2)
	1	V(1,0)	V(1,1)	V(1,2)
	2	V(2,0)	V(2,1)	V(2,2)

É necessário que a GLCM seja normalizada, apresentando probabilidades de ocorrência de combinações, em vez de números absolutos de ocorrências. Essas probabilidades  $P_{i,j}$  são calculadas segundo a Equação 2.22, para  $N_{NC}$  níveis de cinza.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{p=0}^{N_{NC}-1} \sum_{q=0}^{N_{NC}-1} V_{p,q}} \quad (2.22)$$

Cada um dos 14 descritores de Haralick é calculado usando-se os valores da GLCM de uma forma diferente. Antes de calcular os descritores, é preciso determinar alguns valores auxiliares, os quais são definidos pelas Equações 2.23 a 2.29. Os 14 descritores são dados pelas Equações 2.30 a 2.43. Nos cálculos envolvendo logaritmos, soma-se  $\epsilon$ , um valor

arbitrário positivo muito pequeno, ao argumento do log, a fim de evitar a indefinição  $\log(0)$ .

$$P_x(i) = \sum_{j=0}^{N_{NC}-1} P_{i,j} \quad (2.23)$$

$$P_y(j) = \sum_{i=0}^{N_{NC}-1} P_{i,j} \quad (2.24)$$

$$P_{x+y}(k) = \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} \quad (2.25)$$

com  $i + j = k$  e  $k = 0, 1, \dots, 2N_{NC} - 2$ , onde  $P_x(i)$  e  $P_x(j)$  correspondem à  $i$ -ésima e à  $j$ -ésima entradas na matriz de probabilidades obtida somando os elementos na linha de  $P_{i,j}$ .

$$P_{x-y}(k) = \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} \quad (2.26)$$

com  $|i - j| = k$  e  $k = 0, 1, \dots, N_{NC} - 1$ .

$$HXY1 = - \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} \log(P_x(i)P_y(j) + \varepsilon) \quad (2.27)$$

$$HXY2 = - \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_x(i)P_y(j) \log(P_x(i)P_y(j) + \varepsilon) \quad (2.28)$$

$$Q(i,j) = \sum_{k=0}^{N_{NC}-1} \frac{P_{i,k}P_{j,k}}{P_x(i)P_y(j)} \quad (2.29)$$

A seguir são apresentadas as equações relativas à Segundo Momento Angular (SMA), Contraste e Correlação, respetivamente.

Segundo Momento Angular (SMA ou  $f_1$ ): tanto o segundo momento angular (SMA), quanto a energia usam  $P_{i,j}$  como pesos. Valores elevados de SMA ou energia ocorrem quando a imagem está muito ordenada

$$f_1 = \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j}^2 \quad (2.30)$$

Contraste ( $f_2$ ): é uma estimativa das variações locais dos níveis de cinza entre pares de pixels. Essa medida é também chamada de soma do quadrado da variância.

Medidas relacionadas a contraste usam pesos relacionados com a distância da

diagonal da GLCM. Valores na diagonal da GLCM representam pouco contraste, aumentando quando a distância da diagonal aumenta.

$$f_2 = \sum_{n=0}^{N_{NC}-1} n^2 \left( \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} \right) \quad (2.31)$$

Correlação ( $f_3$ ): o coeficiente de correlação utilizado em estatística é uma medida do grau de associação linear (negativa ou positiva) entre duas variáveis quantitativas. A correlação em textura mede a dependência linear de pixels em relação à sua vizinhança.

$$f_3 = \frac{\sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.32)$$

onde  $\mu_x$  e  $\sigma_x$  são, respectivamente, a média e o desvio padrão de  $P_x$  e  $\mu_y$  e  $\sigma_y$ , a média e o desvio padrão de  $P_y$ .

Variância ( $f_4$ ): a variância baseia-se na dispersão dos dados numéricos em torno de um valor médio, de valores dentro da GLCM. Entretanto, a variância da GLCM trata especificamente das combinações do *pixel* de referência e do seu vizinho.

$$f_4 = \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} (i - \mu)^2 P_{i,j} \quad (2.33)$$

sendo  $\mu$  a média dos níveis de cinza presentes na imagem.

Momento inverso da diferença ( $f_5$ ): quando os valores altos da GLCM estiverem próximos da diagonal, este descritor possuirá também valor alto.

$$f_5 = \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} \frac{P_{i,j}}{1 + (i - \mu)^2} \quad (2.34)$$

Média da soma ( $f_6$ ):

$$f_6 = \sum_{i=2}^{2N_{NC}} iP_{x+y}(i - 2) \quad (2.35)$$

Variância da soma ( $f_7$ ):

$$f_7 = \sum_{i=2}^{2N_{NC}} (i - f_6)^2 P_{x+y}(i - 2) \quad (2.36)$$

Entropia ( $f_8$ ) é normalmente denominada como uma medida de primeiro grau, mas

deveria propriamente ser uma medida de zero grau. Como a função logarítmica não é definida para zero e valores negativos, assume-se que  $0 * \log_0 = 0$ .

$$f_8 = - \sum_{i=0}^{N_{NC}-1} \sum_{j=0}^{N_{NC}-1} P_{i,j} \log(P_{i,j} + \varepsilon) \quad (2.37)$$

Entropia da soma ( $f_9$ ):

$$f_9 = - \sum_{i=0}^{2N_{NC}-2} P_{x+y}(i) \log(P_{x+y}(i) + \varepsilon) \quad (2.38)$$

Variância da diferença ( $f_{10}$ ):

$$f_{10} = \text{Variância de } (P_{x-y}) \quad (2.39)$$

Entropia da diferença ( $f_{11}$ ):

$$f_{11} = - \sum_{i=0}^{N_{NC}-1} P_{x-y}(i) \log(P_{x-y}(i) + \varepsilon) \quad (2.40)$$

Média de correlação 1 ( $f_{12}$ ):

$$f_{12} = \frac{f_9 - HXY1}{MAX(HX, HY)} \quad (2.41)$$

sendo  $HX$  e  $HY$  as entropias de  $P_x$ ,  $P_y$  e.

Média de correlação 2 ( $f_{13}$ ):

$$f_{13} = (1 - e^{-2(HXY2 - f_9)})^{1/2} \quad (2.42)$$

Máximo coeficiente de correlação ( $f_{14}$ ):

$$f_{14} = (\text{segundo maior autovalor de } Q)^{1/2} \quad (2.43)$$

#### 2.4.4 DESCRITORES COR

Os descritores de cor utilizados neste trabalho estão relacionados a uma região de uma imagem, baseados nas propriedades de intensidade de *pixels* em uma vizinhança. A média (*mean*) e o desvio padrão (*stddev*) são duas propriedades muito usadas, devido a sua relevância para a aparência da imagem. A média relaciona-se com a medida de brilho, já o desvio padrão está relacionado ao contraste. Estas propriedades podem ser extraídas usando um ou mais canais de cores e podem ser calculados pelas Equações 2.44 a 2.46.



$$N = \sum_{I, \text{mask}(I) \neq 0} 1 \quad (2.44)$$

$$\text{mean}_c = \frac{\sum_{I: \text{mask}(I) \neq 0} \text{src}(I)_c}{N} \quad (2.45)$$

$$\text{stddev}_c = \sqrt{\frac{\sum_{I: \text{mask}(I) \neq 0} (\text{src}(I)_c - \text{mean}_c)^2}{N}} \quad (2.46)$$

onde  $I$  é a intensidade dos *pixels*, *mask* está relacionado ao canal ou aos canais de cores selecionados e *src* é a imagem de entrada ou a região da imagem de entrada.

## 2.5 RECONHECIMENTO DE PADRÕES

O reconhecimento de padrões é o campo da ciência da computação que tem por objetivo a classificação de objetos em um determinado número de categorias ou classes a partir da observação de suas características (SANTOS, 2009). O reconhecimento de padrões visa construir uma representação mais simples de um conjunto de dados através de suas características mais relevantes, possibilitando sua partição em classes (DUDA; HART; STOK, 2000).

Um padrão é um tipo de tema de eventos recorrentes de objetos, algumas vezes referidos como elementos de um conjunto. Estes elementos de um modo previsível são modelos que podem ser usados para gerarem objetos ou partes de um objeto, especialmente se os objetos que são criados têm em comum o suficiente para o padrão subjacente ser inferido. Exemplo: reconhecimento da impressão digital, reconhecimento de voz e etc.

Há duas maneiras de reconhecer e/ou classificar o padrão (CONNELL; JAIN, 2001):

- **Classificação supervisionada:** o padrão de entrada é identificado como um membro de uma classe pré-definida, ou seja, a classe é definida pelo projetista do sistema;
- **Classificação não supervisionada:** o padrão é determinado por uma “fronteira” de classe desconhecida.

Assim, o reconhecimento de padrão é uma tarefa de classificação ou categorização, onde as classes são definidas pelo projetista (classificação supervisionada) ou são “aprendidas” de acordo com a similaridade dos padrões (classificação não supervisionada).

Entende-se por padrão as propriedades que possibilitam o agrupamento de objetos semelhantes dentro de uma determinada classe ou categoria, mediante a interpretação de dados de entrada, que permitam a extração das características relevantes desses objetos

(ANTONIO *et al.*, 2002). Entende-se por classe de um padrão, um conjunto de atributos comuns aos objetos de estudo.

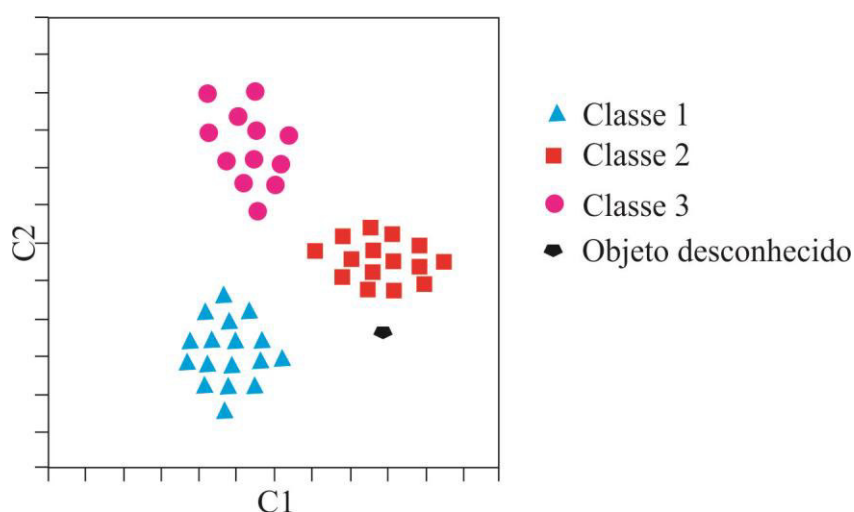
Assim, reconhecimentos de padrões podem ser definidos como sendo um procedimento em que se busca a identificação de certas estruturas nos dados de entrada, em comparação a estruturas conhecidas e sua posterior classificação dentro de categorias, de modo que o grau de associação seja maior entre estruturas de mesma categoria e menor entre as categorias de estruturas diferentes.

### 2.5.1 CLASSIFICAÇÃO SUPERVISIONADA

Na classificação supervisionada, as classes são definidas a partir de uma base de conhecimento, formada por exemplos dos padrões conhecidos. São fornecidos os exemplos e suas respectivas classificações com as quais o sistema é treinado.

Na Figura 2.10, pode-se observar um conjunto de objetos representados em um espaço de características bidimensional. Estes objetos são conhecidos e foram previamente rotulados (classe 1, classe 2, classe 3). Assim, a partir desta base de conhecimento, pode ser atribuída uma classe a um objeto desconhecido através de um procedimento de classificação supervisionada. No exemplo, o objeto desconhecido é classificado como pertencente à classe 2. O critério utilizado é a proximidade no espaço de características, aos objetos desta classe.

Figura 2.10 Classificação supervisionada



Fonte: Adaptada de Gomes (2007).

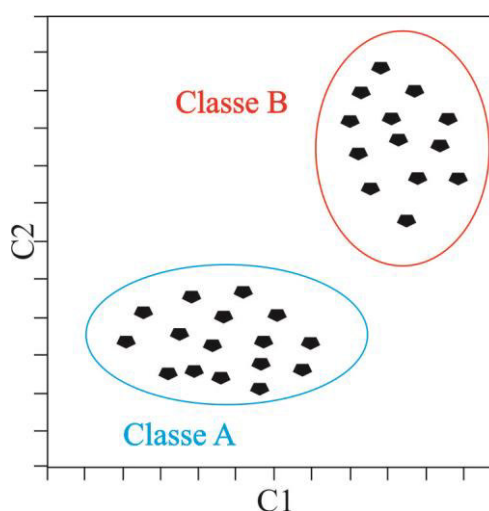
### 2.5.2 CLASSIFICAÇÃO NÃO SUPERVISIONADA

Na classificação não supervisionada não há nenhuma informação prévia sobre as classes às quais os padrões na amostra pertencem. Dado um conjunto de padrões, métodos

não supervisionados os agrupam em função de sua disposição no espaço de características.

A Figura 2.11 mostra a classificação não supervisionada. Nela, outro conjunto de objetos está representado no espaço de características bidimensional, contudo agora os padrões dos objetos não são conhecidos. Um procedimento de classificação não supervisionada procura por objetos similares e os agrupa em classes. No exemplo em questão, os objetos foram divididos entre duas classes distintas (classe A e classe B).

Figura 2.11 Classificação não supervisionada



Fonte: Adaptada de Gomes (2007).

## 2.6 DESBALANCEAMENTO DE DADOS

Muitos aspectos podem influenciar no desempenho de um classificador, um desses aspectos está relacionado com a diferença de números de exemplos pertencentes a cada uma das classes. Quando essa diferença é grande, os sistemas de aprendizado podem encontrar dificuldades em aprender características relevantes às classes minoritárias. Nessas condições, os classificadores tornam-se tendenciosos a terem um bom desempenho para as classes majoritárias e um péssimo desempenho para classes minoritárias, o que leva este classificador a tornar-se um modelo de classificação trivial.

Vários pesquisadores têm analisado o problema de aprender a partir de conjuntos de dados com classes desbalanceadas, além disso, ocorreram dois *workshops* internacionais relacionados a esse tema, incluindo revistas relacionadas ao grupo de estudo de mineração de dados da ACM, SIGKDD Explorations, citado em Prati (2006).

Duas abordagens principais têm sido utilizadas com maior frequência nesse tema. Uma delas consiste em atribuir custos diferenciados às classes no momento de induzir o

modelo (ZADROZNY; LANGFORD; ABE, 2003) (PRATI, 2006). A outra é anterior à fase de treinamento e baseia-se em reamostragem de dados.

Na abordagem de reamostragem, a ideia é usar um conjunto mais balanceado do que o conjunto original para construir o classificador. Os métodos de tratamento baseados em reamostragem dividem-se em duas categorias, conhecidas como subamostragem (em inglês, *undersampling*) e sobreamostragem (em inglês, *oversampling*).

Dentre as abordagens de reamostragem, foram selecionadas duas abordagens bastante comuns em aprendizado de máquina. São elas: remoção de exemplos das classes majoritárias e inclusão de exemplos das classes minoritárias (PRATI, 2006).

A primeira abordagem é geralmente chamada de *undersampling*. Uma forma bastante direta de solucionar o problema de classes desbalanceadas é balancear artificialmente a distribuição das classes no conjunto de exemplos. Os métodos de *undersampling* visam balancear o conjunto de dados por meio da remoção de exemplos das classes majoritária, procurando aproximar a classe minoritária.

A segunda abordagem têm métodos que são geralmente conhecidos como métodos de *oversampling*. Os métodos de *oversampling* são similares aos métodos de *undersampling*. Porém, esses métodos visam balancear a distribuição das classes por meio da replicação de exemplos da classe minoritária até se aproximar a classe majoritária. (PRATI, 2006).

### 2.6.1 PROBLEMAS COM OS MÉTODOS DE BALANCEAMENTO

Os métodos de balanceamento direto, *oversampling* e *undersampling*, geralmente trabalham com inclusão ou remoção de dados aleatórios. Por esse motivo, estes métodos têm algumas limitações a serem considerados:

Com os métodos de *undersampling*, há possibilidade de eliminar dados úteis, enquanto o *oversampling* aleatório pode aumentar as chances de ocorrer um superajustamento aos dados, pois as cópias exatas dos exemplos pertencentes à classe minoritária serão duplicadas.

Trabalhos recentes têm tentado superar as limitações existentes desses métodos. Uma das formas de minimização é utilizar heurísticas para selecionar os exemplos a serem acrescentados/removidos, cujo principal objetivo é minimizar a quantidade de dados potencialmente úteis descartados. (PRATI, 2006).

## 2.6.2 OS MÉTODOS SELECIONADOS.

A técnica chamada SMOTE (CHAWLA *et al.*, 2002) define um método de *oversampling* que não replica os exemplos de treinamento. Ao invés disso cria novos exemplos sintéticos com base nas características espaciais entre exemplos da classe minoritária. Especificamente, para um subconjunto  $S_{min}$  (conjunto da classe minoritária), considere os  $k$  vizinhos mais próximos para cada exemplo que pertençam a  $S_{min}$  para algum valor inteiro  $k$ . Dependendo da quantidade de *oversampling* escolhida os  $k$  vizinhos mais próximos são escolhidos aleatoriamente. Amostras sintéticas são geradas da seguinte maneira: calcule a distância euclidiana entre o vetor de pontos (amostras) em questão e seu vizinho mais próximo. Multiplique essa distância por um número aleatório entre zero e um e o adicione ao vetor de pontos em consideração. Isso causa a seleção de um ponto ao longo de uma linha entre os dois pontos escolhidos. Esta aproximação eficaz faz com que a região da classe minoritária fique com mais força para tornar-se mais geral. Esse método é chamado de SMOTE (*Synthetic Minority Oversampling Technique*).

*Spread Subsample* (HALL *et al.*, 2009) é um método de *undersampling*, que elimina amostras aleatórias nos exemplos da classe majoritária. Para realizar o balanceamento, é configurada a razão entre as classes majoritária e minoritária.

## 2.7 REDES NEURAIAS ARTIFICIAIS

A Rede Neural Artificial (RNA) é um dos ramos da Inteligência Artificial, que utiliza abordagem conexionista para a solução de problemas computacionais. São caracterizadas por sistemas que, em algum nível, relembram a estrutura do cérebro humano. Além disso, como a computação neural não é baseada em regras, então ela constitui-se uma alternativa à computação algorítmica convencional.

A capacidade mais interessante das RNAs é, assim como o cérebro humano, aprender através de exemplos e representar essas experiências na forma de pesos sinápticos. O “aprendizado” pode ser compreendido como a aquisição de uma relação a partir dos dados de entrada e de saída. Tal característica está presente na abordagem chamada por alguns autores de conexionista e existem atualmente diversas técnicas que a implementem entre elas: Redes Neurais Artificiais, Algoritmo Genético e Aprendizagem por Reforço.

As RNAs têm a capacidade de calcular, armazenar e distribuir as informações pela rede. São sistemas distribuídos compostos por unidades de processamento simples (neurônios

artificiais) que calculam determinadas funções matemáticas (normalmente não lineares). Tais unidades são dispostas em uma ou mais camadas e fortemente interligadas entre si, geralmente unidirecional (BRAGA; DE CARVALHO; LUDEMIR, 2007), mas, há casos de conexões de realimentação. Na maioria dos modelos, essas conexões estão associadas a valores numéricos chamados de pesos, que têm a função de armazenar o conhecimento adquirido pelo modelo e servem para ponderar cada entrada recebida por um neurônio da rede. Em Haykin (1999), é dito que uma rede neural é um processador paralelamente distribuído e constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso.

As etapas de uma RNA para solucionar um conjunto de problemas têm sua fase inicial no aprendizado, onde um conjunto de exemplos é apresentado à rede para extrair as características do problema. Com base nessas características, é possível nas próximas etapas do processo reutilizá-las, a fim de encontrar solução para outros casos do mesmo problema.

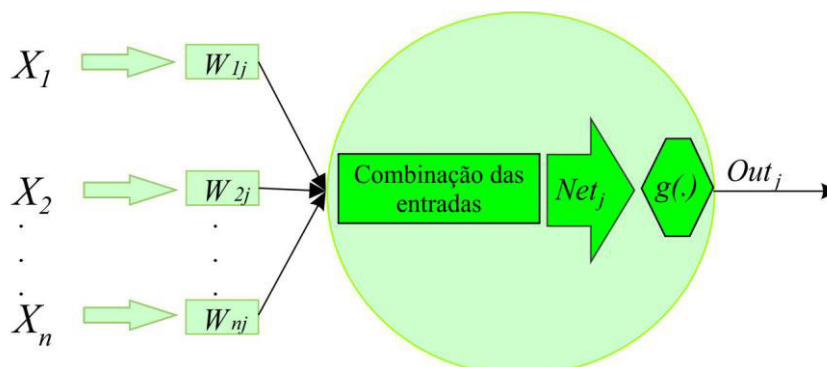
A capacidade de aprender por meio de exemplos e de generalizar a informação aprendida é, sem dúvida, o atrativo principal da solução de problemas por meio das RNAs.

O neurônio artificial de forma individual possui capacidade computacional limitada, porém quando existe um conjunto de neurônios artificiais conectados formando uma rede, essa capacidade eleva-se bastante, sendo capaz de resolver problemas de alta complexidade.

O neurônio artificial é a unidade elementar de processamento da informação de uma RNA. Eles foram desenvolvidos através da simplificação do que se conhecia a respeito dos neurônios biológicos. Esse modelo é chamado MCP, representando as iniciais dos nomes de seus criadores, McCulloch e Pitts.

O neurônio artificial MCP consiste em dois elementos importantes: um módulo de combinação de entradas e uma função de ativação, apresentados na Figura 2.12.

Figura 2.12 Modelo de um neurônio MCP



Fonte: Adaptada de Braga; De Carvalho; Ludemir (2007).

Baseado na Figura 2.12, as entradas são representadas por  $x$  e os pesos são representadas por  $w$ . A combinação das entradas é o módulo responsável pela tarefa de coletar e combinar as informações vindas de outros neurônios. Esta combinação é representada por  $net$  e geralmente é um somatório básico das entradas ponderadas do neurônio. Este valor  $net$  é dado por:

$$net_j = \sum_{i=1}^n w_{ij}x_j \quad (2.47)$$

A função de ativação  $g(.)$  define a saída do neurônio em termos do nível do sinal aplicado à sua entrada, neste caso esta função é responsável por gerar a saída  $out$  a partir dos valores dos vetores de pesos  $w = (w_{1j}, w_{2j}, w_{nj})$  e de entrada  $x = (x_1, x_2, \dots, x_n)$ , ou seja:

$$out_j = g(net_i) \quad (2.48)$$

Podem-se identificar três tipos básicos de funções de ativação, cujas equações estão dispostas na Tabela 2.7:

Tabela 2.7 Equações das funções de ativação mais comuns

Função de ativação	Equação
Linear	$out = k.net$ , onde $k$ é uma constante
Sigmoide	$out = (1 + e^{-net})^{-1}$
Tangente hiperbólica	$out = (1 - e^{-net})(1 + e^{-net})^{-1}$

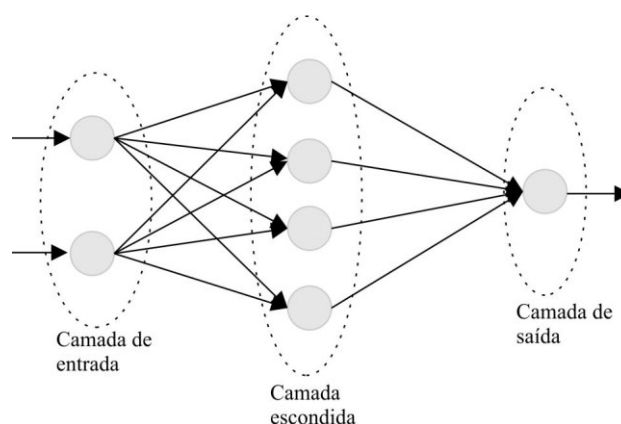
### 2.7.1 ARQUITETURAS

A maioria das arquiteturas neurais é tipicamente organizada em camadas, como mostra a Figura 2.13, onde suas unidades podem estar conectadas tanto às unidades das camadas posteriores, quanto às camadas anteriores. Nessa figura, os círculos representam os neurônios, as colunas verticais de neurônios representam as camadas e as setas que ligam os neurônios são chamadas de pesos. Normalmente as camadas são classificadas em três grupos:

- **Camada de Entrada:** a única função dessa camada é receber os padrões de entrada e repassá-los à camada seguinte;
- **Camadas Escondidas:** é responsável pela maior parte do processamento. Elas também são chamadas extratoras de características;
- **Camada de Saída:** onde o resultado final é concluído e apresentado.

As conexões entre as camadas podem gerar várias estruturas diferentes (arquiteturas).

Figura 2.13 Arquitetura de uma rede neural artificial organizada em camadas



Fonte: Adaptada de de Almeida Neto (2003)

Na arquitetura apresentada na Figura 2.13, a coluna à esquerda representa os neurônios da camada de entrada, a qual simplesmente recebe os dados para serem processados pela RNA nas camadas seguintes. Desta forma, para alguns autores, a camada de entrada não é considerada uma camada de neurônios de fato, pois não processa nenhuma informação. A quantidade de neurônios na camada de entrada é definida pela quantidade de informação a ser processada pela rede, assim há um neurônio para cada informação de entrada da rede.

A coluna central é conhecida como camada escondida, oculta ou intermediária, nomenclatura que varia entre alguns autores. Ela é responsável por realizar o primeiro processamento. Existe também a possibilidade de uso de mais de uma camada escondida, em especial o caso conhecido como *deep learning*, onde uma rede possui muitas camadas desse tipo.

A última coluna representa a camada de saída, que realiza o processamento final da rede, de tal modo que a RNA apresente seus resultados ao meio externo. Assim como na camada de entrada, é obrigatório que exista um neurônio para cada saída da rede.

Os pesos na rede artificial representam as sinapses de um neurônio biológico. Eles são valores numéricos a serem definidos com o objetivo de acumular conhecimento.

A quantidade de camadas e neurônios é uma especificação definida pelo projetista da RNA e costuma ser relacionada com o problema. Assim, a definição correta dessas informações é fundamental para uma boa solução, então a modelagem da arquitetura da RNA passa a ser uma tarefa de grande importância.

À forma como as conexões podem estar arranjadas, dá-se o nome de topologia, que podem ser classificadas como: *feedforward* e *feedbackward*. A topologia *feedforward*



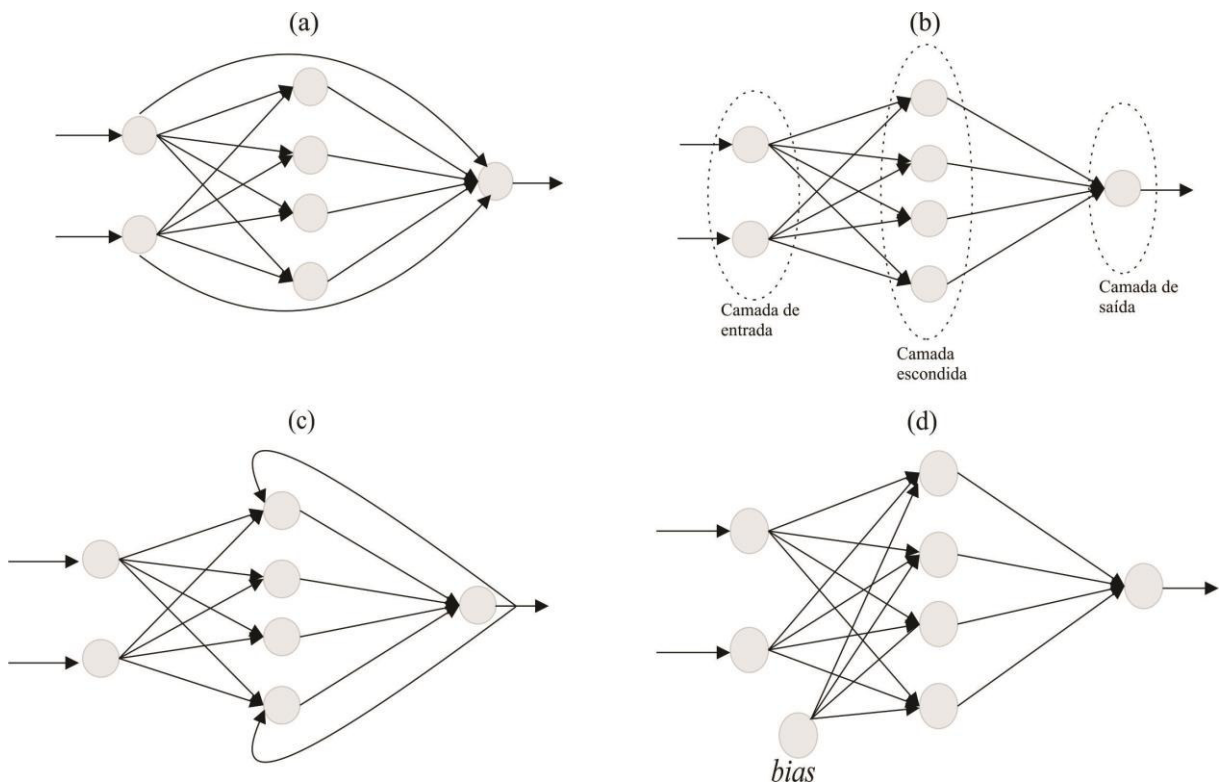
apresenta fluxo de dados em apenas um sentido, partindo de um neurônio de uma camada e terminando em outro neurônio de uma camada posterior, situado na Figura 2.14 (a).

Um caso especial da topologia *feedforward* é chamado estritamente *feedforward*, que apresenta fluxo de dados em apenas um sentido, porém, o neurônio de uma camada somente pode conectar-se ao neurônio da camada imediatamente seguinte, presente na Figura 2.14 (b).

Na topologia *feedbackward*, o fluxo de dados pode ocorrer em ambos os sentidos, como pode ser visto na Figura 2.14 (c).

Vale lembrar que independente da topologia da rede adotada, pode-se adicionar um elemento semelhante a um neurônio, conectando-o à camada intermediária ou à camada de saída. Esse elemento é conhecido como *bias* (Figura 2.14 (d)) e sua saída é sempre o valor um, porém, as conexões entre esse elemento e os neurônios também são pesos que influenciam o aprendizado da rede.

Figura 2.14 Topologias de RNA



Fonte: Adaptada de de Almeida Neto (2003).

A definição da estrutura de uma RNA para resolução de um determinado problema depende de vários fatores, entre eles (BRAGA; DE CARVALHO; LUDEMIR, 2007):

- complexidade do problema;

- dimensionalidade do espaço de entrada;
- características dinâmicas ou estáticas;
- conhecimento *a priori* sobre o problema;
- representatividade dos dados.

Esses fatores vão influenciar diretamente a arquitetura ou topologia de um RNA.

### 2.7.2 PROCESSOS DE APRENDIZAGEM

Segundo Haykin (1999), o aprendizado é o processo pelo qual os parâmetros livres, ou pesos de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida.

O referido estímulo consiste nos dados de entrada da rede, que depois de recebidos dão início a um procedimento de modificação interna da RNA, necessário para que ocorra o aprendizado. Este procedimento, chamado de treinamento, de fato ajusta os pesos até que a saída da rede assemelhe-se a um valor desejado para cada entrada.

As abordagens de aprendizado de máquina são classificadas como aprendizado supervisionado, não supervisionado e semi-supervisionado. A aprendizagem supervisionada é marcada pela necessidade de um supervisor para fazer a rotulação dos dados que serão passados para rede. Na aprendizagem não supervisionada, os dados não são rotulados, e a rede é capaz de aprender as características destes dados e apresentar como saída agrupamento de dados através das características compartilhadas entre eles. No caso da aprendizagem semi-supervisionada é indicado para treinamentos de classificadores, quando uma grande quantidade de exemplos de dados não rotulados estão disponíveis, junto a um pequeno conjunto de exemplos rotulados. Os benefícios deste tipo de aprendizado estão em aprender características dos dados rotulados e agrupar os dados não rotulados levando em conta características comuns aprendidas.

### 2.7.3 PERCEPTRON DE MÚLTIPLAS CAMADAS

Dentre a variedade de redes neurais publicadas em revistas especializadas, os modelos bastante utilizados são: Multilayer Perceptron (Perceptron de Múltiplas Camadas - MLP), rede de funções de base radial (RBF) e rede de Kohonen. Os principais aspectos que

diferem os diversos tipos de redes neurais estão relacionados ao tipo de conexão entre os neurônios, o número de camadas da rede e o algoritmo de treinamento utilizado.

A MLP é a rede neural de estudo neste trabalho, por isso a seguir, serão apresentados os principais aspectos deste modelo de RNA. A rede MLP é formada por neurônios derivados do modelo de McCulloch e Pitts. Para o treinamento dessas redes, utiliza-se frequentemente o algoritmo *backpropagation*, desenvolvido por Rumelhart *et al.*, (1986). Durante o treinamento com o algoritmo *backpropagation*, a rede opera alternativamente nas fases: *feedforward* e *feedback*. Na primeira fase, um padrão é apresentado à camada de entrada da rede, que segue aos demais neurônios das camadas seguintes (*forward*), camada por camada, até que a resposta seja produzida pela camada de saída, com o objetivo de calcular o erro da rede. Nesta primeira fase, os pesos das conexões não se alteram.

Na segunda fase, a saída obtida é comparada à saída desejada para o padrão particular da entrada. Se esta não estiver correta, ocorrerá um erro, que é calculado para servir de base para a modificação dos pesos, por isso ele é retropropagado (*feedback*) a partir da camada de saída até a primeira camada escondida (logo posterior à camada de entrada). Os pesos das conexões entre as unidades das camadas internas vão sendo modificados conforme o erro é retropropagado.

Deste modo, os pesos são ajustados de maneira que o erro diminua a cada iteração por meio de uma variação no valor atual de cada peso. O cálculo da variação dos pesos entre as camadas escondida e de saída é mostrado com a seguinte equação:

$$\Delta W_{kj} = \eta \cdot e_k \cdot f'_k \cdot out_j \quad (2.49)$$

onde  $\Delta W_{kj}$  representa a variação do peso entre o neurônio  $k$  da camada de saída e o neurônio  $j$  da camada escondida;  $\eta$  é a taxa de aprendizado, definida pelo projetista;  $e_k$  é o erro de saída do neurônio  $k$ ;  $f'_k$  é a derivada da função de ativação do neurônio  $k$  com relação à sua entrada  $net$  e  $out_j$  é o valor de saída do neurônio  $j$  da camada escondida.

O parâmetro  $\eta$  é fundamental neste processo de atualização dos pesos, tendo em vista que ele regula o aprendizado. Se este valor for alto, o treinamento poderá ser mais rápido, mas, poderá também ocorrer oscilações no processo de aprendizagem de modo a dificultar o encontro de um mínimo adequado. Caso  $\eta$  apresente um valor pequeno, o treinamento tende a ter menos oscilações, porém, será mais lento.

Dependendo dos valores de inicialização dos pesos, o processo de aprendizagem poderá ficar preso em um mínimo local, diminuindo as chances de sucesso no treinamento.

Cabe ao projetista ajustar os parâmetros, como a taxa de aprendizado, de forma a conduzir o processo de aprendizagem em um tempo curto e com o menor erro possível (OLIVEIRA, 2013).

O número de vezes que o conjunto de treinamento completo é apresentado à rede é chamado de épocas do treinamento, ou seja, cada vez que o conjunto de treinamento completo é apresentado, conta-se uma época.

O aprendizado da rede neural é, de modo geral, avaliado com base no erro de saída da rede durante o treinamento. Este erro é toda iteração calculada para verificação se a saída esperada da rede é obtida, ou seja, se está em um valor aceitável pelo projetista.

Completado o treinamento da rede, ou seja, quando o aprendizado estiver em um nível satisfatório, ela poderá ser utilizada como uma ferramenta para classificação de novos dados. Para isto, a rede deverá ser utilizada apenas no modo progressivo *feedforward*, ou seja, novas entradas são apresentadas à camada de entrada, são processadas nas camadas intermediárias e os resultados são apresentados na camada de saída, como no treinamento, mais sem a retropropagação do erro. A saída corresponde à interpretação da rede para a nova entrada apresentada.

Para avaliar o aprendizado da rede é preciso fazer a validação. A validação da rede é realizada com testes de generalização, onde dados não usados no treinamento são apresentados à rede para verificar se a saída está correta, como pretendido. Neste teste, utilizou-se a rede no modo progressivo. Caso a rede consiga apresentar uma saída correta para esses dados, é sinal de que a rede conseguiu aprender a solução do problema, independente dos dados de entrada.

## 2.8 ANÁLISE ROC

Análise ROC, do inglês *Receiver Operating Characteristic*, é um método gráfico para avaliação que permite visualizar, avaliar, organizar e selecionar sistemas de diagnósticos e/ou predição. Para realizar estas análises, gráficos ROC podem mostrar o limiar entre taxa de verdadeiros positivos (sensibilidade) e taxas de falsos positivos (especificidade) dos classificadores. Originalmente foram utilizados em detecção de sinais, para avaliar a quantidade de transmissão de um sinal em um canal com ruído (EGAN, 1975).

Um dos primeiros usos dos gráficos ROC em aprendizado por máquina foi desenvolvido por Spackman (1989), que demonstrou a utilidade das curvas ROC na avaliação

e comparação entre algoritmos.

Para construir o gráfico ROC, precisa-se de métricas capazes de avaliar o desempenho dos classificadores, como sensibilidade e especificidade, que são extraídas da matriz de confusão, que será explicada na próxima seção.

### 2.8.1 DESEMPENHO DOS CLASSIFICADORES

Alguns classificadores geram como saída valores contínuos (em geral, um classificador que gera a probabilidade de uma instância pertencente a uma dada classe), para os quais diferentes limiares podem ser aplicados para gerar diferentes conjuntos de saída (em um classificador contínuo pode-se aplicar um limiar para binarizá-lo, e gerar duas classes). Outros geram resultados discretos indicando somente a classe.

Considera-se um conjunto de amostras, onde uma instância  $I$  (instância representa um elemento que não se sabe a qual classe pertence e deseja-se classificá-lo), pode assumir valores no conjunto positivo e negativo, respectivamente. Tendo-se um classificador e uma instância podem-se ter quatro situações.

- se a instância é positiva e é classificada como positiva, conta-se como *true positive* ( $a$ ), verdadeiro positivo (TP);
- se a instância é positiva e é classificada como negativa, conta-se como *false negative* ( $b$ ), falso negativo (FN);
- se a instância é negativa e é classificada como tal, conta-se como *true negative* ( $d$ ), verdadeiro negativo (TN);
- se a instância é negativa e é classificada como positiva, conta-se como *false positive* ( $c$ ), falso positivo (FP).

Portanto, tendo um classificador e um conjunto de instâncias, pode-se construir uma matriz de confusão de 2 por 2, no caso de duas classes. Esta matriz serve como base para muitas métricas que podem ser aplicadas à classificação. A Tabela 2.8 mostra um exemplo de matriz de confusão.

Tabela 2.8 Matriz de confusão para duas classes.

		Classificação	
		Positiva	Negativa
Real	Positivo	a: Verdadeiro Positivo	b: Falso Negativo
	Negativo	c: Falso Positivo	d: Verdadeiro Negativo

Com a matriz de confusão, podem-se obter algumas métricas relacionadas à validação de classificadores, como:

- acurácia (AC): acurácia é a proporção do número total de classificações corretas.

$$AC = \frac{a + d}{a + b + c + d} \quad (2.50)$$

- chamada ou sensibilidade (SEN) ou taxa de verdadeiros positivos (TPR) é a proporção de verdadeiros positivos, ou seja, a capacidade do sistema em prever corretamente a condição para casos que realmente a têm.

$$SEN = \frac{a}{a + b} \quad (2.51)$$

- taxa de verdadeiro negativo ou especificidade (ESP): É a proporção de verdadeiros negativos, ou seja, a capacidade do sistema em prever corretamente a ausência da condição para casos que realmente não a têm.

$$ESP = \frac{d}{d + c} \quad (2.52)$$

- precisão (P): é a proporção de casos positivos que foram classificados corretamente.

$$P = \frac{a}{a + c} \quad (2.53)$$

### 2.8.2 ESPAÇO ROC

Os gráficos ROC são bidimensionais, onde, para construir o gráfico desenham-se no eixo das ordenadas os valores da TPR (*True Positive Rate*, ou Taxa de Verdadeiros Positivos) e no eixo das abscissas os valores da FPR (*False Positive Rate* ou Taxa de Falsos Positivos). A Figura 2.15 mostra somente pontos relativos a classificadores discretos.

Classificadores discretos são aqueles que geram como saída somente uma classe. Estes classificadores fornecem um par (Taxa de Falsos Positivos, Taxa de Verdadeiros Positivos) correspondendo a um ponto no espaço ROC.

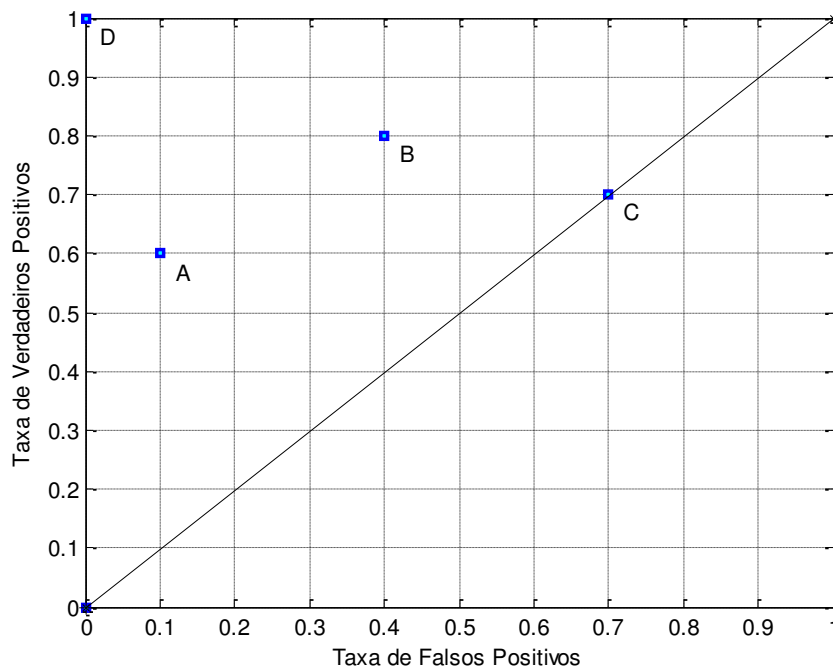
Muitos pontos são importantes no espaço ROC. O ponto inferior esquerdo (0,0) representa uma estratégia que nunca gera uma classificação positiva, como um classificador que não comete erros em classificar os **falsos positivos**, mas, também não classifica nenhum **verdadeiro positivo**. A estratégia oposta, gerar verdadeiros positivos incondicionalmente, é

representada pelo ponto superior direito (1,1).

O ponto (0,1) representa uma classificação perfeita. Este caso pode ser representado pelo classificador relacionado ao ponto (D). De maneira geral, um ponto, no espaço ROC, é melhor que outro, se ele está mais próximo do canto superior do lado esquerdo do gráfico. Classificadores no lado esquerdo do gráfico ROC (perto do eixo Y) são ditos “conservadores”, pois fazem classificações positivas somente com uma evidência forte, portanto, cometem poucos erros em relação aos falsos positivos. Classificadores no lado direito são ditos “liberais”, pois fazem classificações positivas com pouca evidência, mas cometem muitos erros em relação aos falsos positivos. Na Figura 2.15, o ponto (A) é mais conservador que o ponto (B).

Qualquer classificador que estiver abaixo da linha diagonal, que vai do ponto (0,0) ao ponto (1,1), é dito pior do que um classificador aleatório. Por isso, o triângulo inferior de um gráfico ROC está geralmente vazio.

Figura 2.15 Gráfico ROC



Fonte: Elaborado pelo autor.

### 2.8.3 CURVA NO ESPAÇO ROC

Alguns classificadores, tais como *Naive Bayes* ou Redes Neurais, naturalmente produzem uma probabilidade ou valor para cada instância que representa o grau de representatividade da classe sobre aquela instância. A partir destes classificadores, podem-se

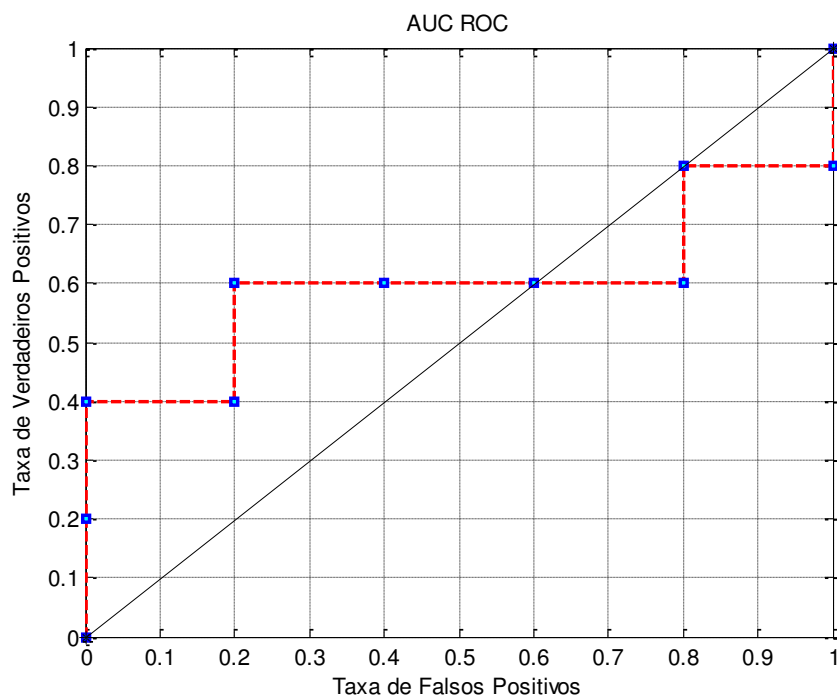
gerar classificadores discretos com uma simples aplicação de um limiar. Cada limiar produzirá um conjunto distinto de pontos no espaço ROC.

A Figura 2.16 mostra um exemplo de uma curva ROC produzida a partir de 1 experimentos com 10 etapas, descritos na Tabela 2.9. Nesta tabela, T representa o número do experimento realizado, TP, FP, TN, FN, FPR e TPR representam as métricas obtidas a partir da matriz de confusão, descritas na Seção 2.8.2. Para cada etapa, foram obtidos diferentes valores para a matriz de confusão. Os valores obtidos no experimento são apresentados na Figura 2.16, através de um gráfico que relaciona os valores de FPR e TPR, este gráfico é utilizado para avaliar o desempenho do classificador.

Tabela 2.9 Tabela de limiares ROC

T	1	2	3	4	5	6	7	8	9	10
TP	5	4	4	3	3	3	3	2	2	1
FP	5	5	4	4	3	2	1	1	0	0
TN	0	0	1	1	2	3	4	4	5	5
FN	0	1	1	2	2	2	2	3	3	4
TPR	1,0	0,8	0,8	0,60	0,60	0,6	0,6	0,4	0,4	0,2
FPR	1,0	1,0	0,8	0,80	0,60	0,4	0,2	0,2	0,0	0,0

Figura 2.16 Uma curva ROC criada pela limiarização



Fonte: Elaborado pelo autor.

Qualquer curva ROC gerada por um número finito de instância produz uma “função degrau” no espaço ROC, então quanto maior o número de instâncias, mais contínua será a curva.



É importante ressaltar que cada ponto criado no tempo  $t$  do processo de geração da curva ROC depende do ponto  $t-1$ . Caso isso não seja considerado, a curva gerará uma nuvem de pontos e não pontos de uma curva.

#### 2.8.4 ÁREA SOB A CURVA ROC

Uma curva ROC é uma demonstração bidimensional do desempenho de um classificador. Para comparar classificadores é preciso reduzir a curva ROC a um valor escalar. Um método comum para realizar esta redução é calcular a área abaixo da curva ROC (AUC – *Area Under Curve*).

Como a qualidade de um classificador depende da área sob a curva ROC, pode-se dividir essa qualidade conforme apresentado na Tabela 2.8.

Para calcular a área sob a curva ROC, dois métodos são comumente utilizados:

- Método paramétrico usando um estimador de máxima verossimilhança para aproximar uma curva suave para os valores dos pares ordenados (1–ESP, SEN);
- Método não paramétrico baseado na construção de trapézios sob a curva.

Tabela 2.10 Qualidade do classificador em relação à área da curva ROC

Área (AUC)	Qualidade do classificador
0.9 a 1.0	Excelente
0.8 a 0.9	Bom
0.7 a 0.8	Regular
0.6 a 0.7	Ruim
0.5 a 0.6	Insignificante

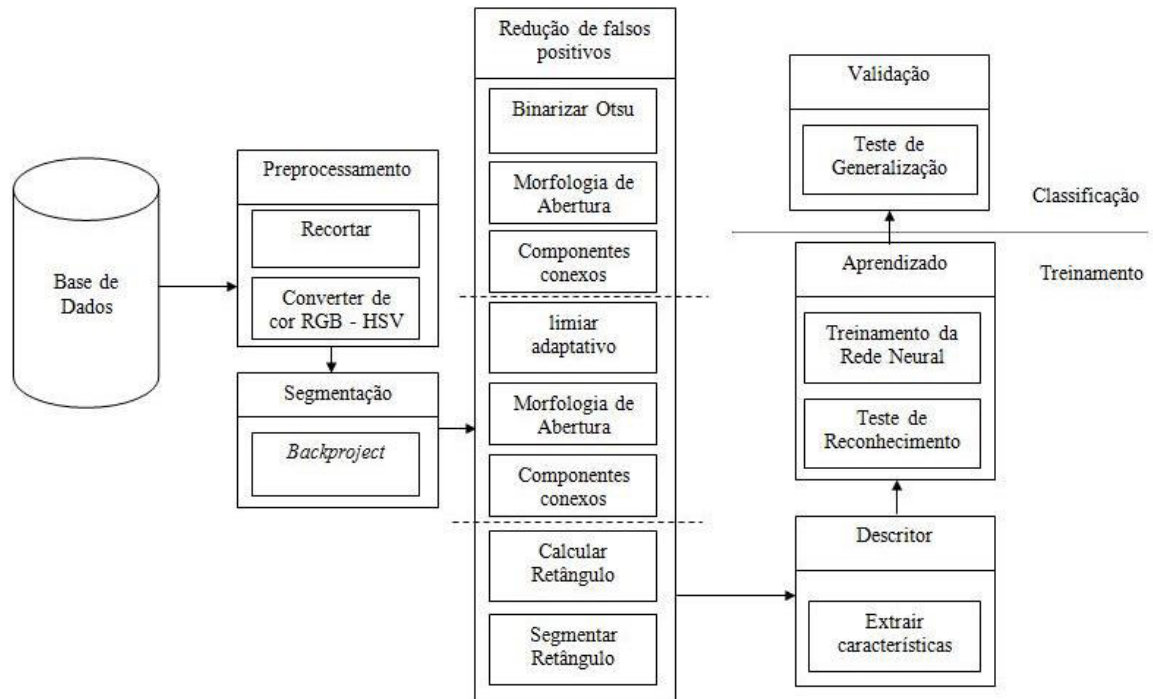
#### 2.8.5 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentadas as teorias sobre as técnicas utilizadas para desenvolver a metodologia do presente trabalho. Assim, este texto utilizou etapas de um projeto de reconhecimento de padrão, em uma metodologia bem específica. No próximo capítulo é descrita essa metodologia, onde estes conceitos são apresentados nas etapas correspondentes, compreendendo: pré-processamento, segmentação, extração de características, aprendizado do classificador e a validação do referido aprendizado.

### 3 METODOLOGIA PROPOSTA

Este trabalho consiste no desenvolvimento de uma metodologia para o reconhecimento de semáforos e a respectiva classificação em verde ligado, vermelho ligado ou não semáforo. Para este propósito, os seguintes módulos foram construídos:

Figura 3.1 Etapas da metodologia



Fonte: Elaborado pelo autor.

Nas subseções seguintes, cada módulo desenvolvido é apresentado, tendo a implementação realizada com o auxílio da biblioteca OpenCV e as linguagens de programação C/C++.

#### 3.1 BASE DE DADOS

Adotou-se neste trabalho a base de dados LaRA (2013), na qual o grupo focal dos semáforos é retangular com cor predominante preta e a sinalização luminosa (foco) é circular de três cores: verde, amarela e vermelha. O posicionamento dos grupos focais dos semáforos é exclusivamente na vertical.

A base de dados para reconhecimento de semáforos em cenas urbanas é de domínio público, sendo formada por imagens e vídeos contendo semáforos. Esta base contém 9.168 imagens, que foram extraídas de um vídeo de 8 min e 49 s, de dimensões 640 x 480 *pixels* (RGB, oito bits), da cidade de Paris, França. Este vídeo foi capturado através de um veículo

C3, equipado com uma câmera sensor Marling F-046C com lentes 12 mm montada no interior do carro sobre o espelho retrovisor, viajando com uma velocidade menor que 50 km/h.

Os semáforos contidos nesta base estão divididos da seguinte forma, de acordo com a luz acionada: 3.381 casos de verde, 58 de amarelo, 5.280 de vermelho e 449 casos ambíguos. Casos ambíguos, no momento da rotulação manual, são regiões que apresentaram problemas na identificação do semáforo real. Estas regiões, não foram consideradas neste estudo.

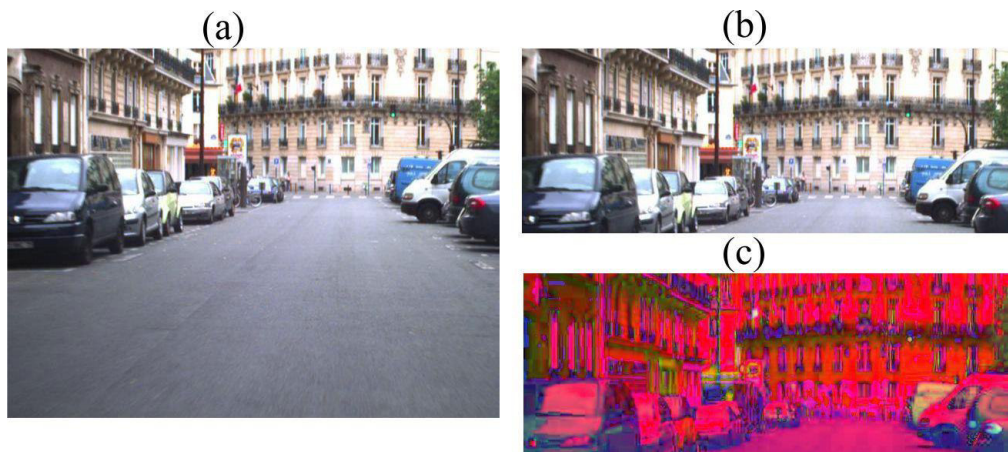
Adicionalmente, alguns semáforos (655) estão parcialmente visíveis na imagem e outros (863) possuem área inferior a cinco *pixels*, por isso também foram ignorados no reconhecimento. Assim, um total de 7.201 casos foram utilizados para a avaliação, sendo 2.594 com verde acionado, 4.549 com vermelho e 58 com a luz amarela acionada.

### 3.2 PREPROCESSAMENTO

Antes de processar a imagem, precisaram-se fazer algumas melhorias nestas e redução de informações desnecessárias. Assim, iniciou-se esta etapa fazendo-se um recorte na imagem original de dimensões 640 x 480 pixels, para uma nova imagem de dimensões 640 x 240 pixels. Este recorte foi possível, pois, os semáforos só aparecem na parte superior da imagem, então a metade inferior das imagens torna-se desnecessária, ver Figura 3.2 (a) e (b).

A etapa seguinte foi a conversão de cor, pois é preciso um espaço de cor que seja pouco influenciado com as variações de luminosidade, o que não ocorre no espaço de cor RGB (original destas imagens). Desta maneira, durante o desenvolvimento, foram testados vários espaços de cor, como: HSV, HSL, YUV, CIELab, RGB Normalizado e YCbCr. O espaço de cor HSV obteve melhores resultados para esta aplicação.

Figura 3.2 Exemplos de imagens: (a) imagem original (b) imagem recortada (c) imagem em HSV



Fonte: Elaborado pelo autor.

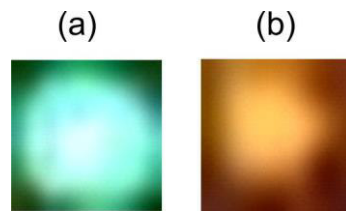
### 3.3 SEGMENTAÇÃO

Esta etapa iniciou-se com a aplicação da técnica de histograma retroprojetado modificado, cujo objetivo foi a segmentação das luzes dos semáforos, descrita na Seção 2.3.3. Para a utilização da técnica, foram selecionados como referência dois objetos ( $O$ ): a imagem do foco de luz do semáforo nas cores verde e vermelha (Figura 3.3). Um objeto por vez é utilizado para fazer a segmentação, iniciando-se pelo objeto verde e depois o objeto vermelho.

Com os objetos ( $O$ ) e a imagem de entrada ( $M$ ), aplicou-se a técnica de histograma retroprojetado, apresentada na Seção 2.3.3, porém, a quantidade de canais utilizados na metodologia foram três (HSV).

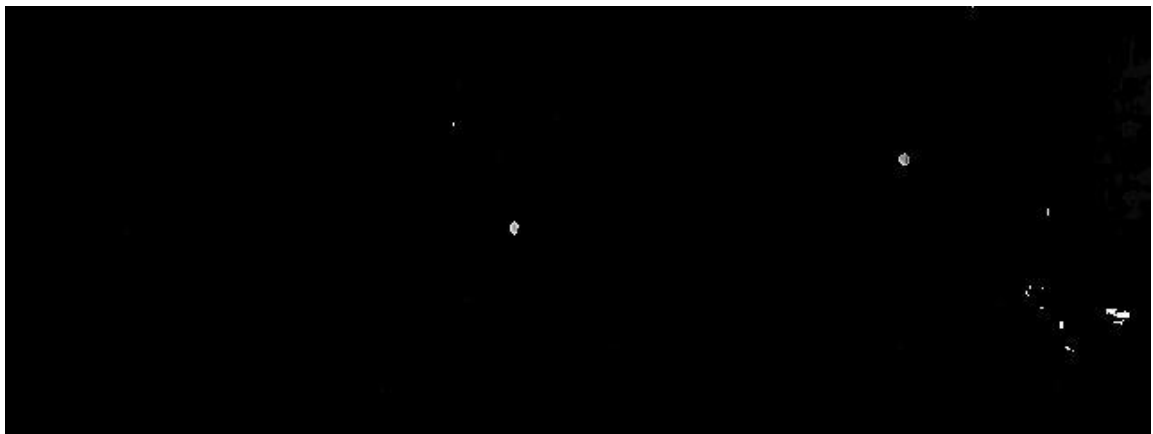
O resultado do algoritmo é armazenado em uma matriz e com ela, pode-se encontrar a localização dos possíveis focos de luzes circulares dos semáforos.

Figura 3.3 luzes do semáforo, (a) luz verde e (b) luz vermelha



Fonte: Elaborado pelo autor.

Figura 3.4 Imagem segmentada



Fonte: Elaborado pelo autor.

Com o resultado da segmentação, o próximo passo foi a aplicação dos filtros morfológicos, utilizando as propriedades de área mínima e circularidade mínima, com o propósito de redução de falsos positivos.

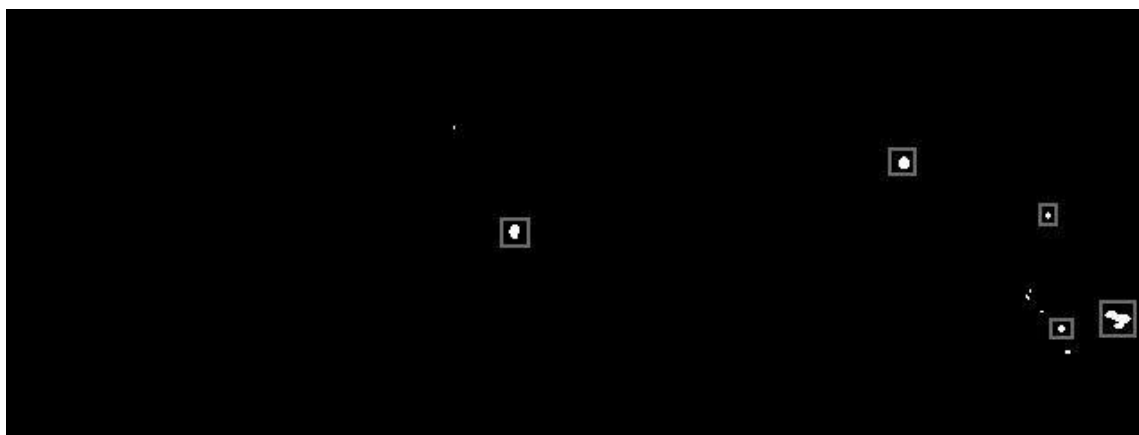
### 3.4 REDUÇÃO DE FALSOS POSITIVOS

Antes da aplicação dos filtros, a matriz resultante da segmentação passou por três operações: A primeira foi a binarização dos valores, através da técnica de Otsu; a segunda foi a aplicação da operação morfológica de abertura, com elemento estruturante elíptico para ressaltar as regiões circulares. Depois foi aplicada a técnica de componentes conexos para encontrar os objetos contidos na imagem.

Através dos componentes conexos, foram extraídas informações dos objetos relacionadas à área e à circularidade.

Com base no exemplo apresentado na Figura 3.4, a quantidade de objetos encontrados foram 10, mas, após aplicação dos filtros, resultaram apenas cinco, que são os objetos delimitados pelos retângulos na cor cinza nessa Figura 3.4. Estes resultados passaram por mais uma etapa de aplicação de filtros morfológicos, com a finalidade de fazer uma segunda redução de falsos positivos, apresentada na próxima seção.

Figura 3.5 Imagem resultante dos filtros morfológicos



Fonte: Elaborado pelo autor.

Com base nos objetos resultantes da primeira redução de falsos positivos, foram utilizadas as informações das localizações espaciais das regiões que delimitam cada objeto. Com essas informações foi possível extrair essas regiões de interesse (ROI) na imagem original. Nesta extração utilizou-se a imagem recortada, apresentada na etapa de pré-processamento, como imagem de entrada.

Com as ROIs extraídas, o objetivo era encontrar as melhores características dos objetos circulares contidos nas ROIs, utilizando-se novamente a técnica de componentes conexos. Antes disso, é necessário preparar as ROIs, para então aplicar um limiar adaptativo local, discutido na Seção 2.3.3, que resultou em ROI binarizada, depois foi aplicada a

operação morfológica de abertura, utilizando elemento estruturante elíptico. Depois destes processos, aplicou-se a técnica de componentes conexos, além de utilizar a filtragem morfológica de área mínima e circularidade mínima, para fazer a segunda redução de falsos positivos. Após a aplicação da primeira redução de falsos positivos, a segunda redução de falsos positivos foi realizada, resultando apenas dois objetos, que podem ser visualizados na Figura 3.6 (a) com destaque em (c) e (e).

Com base neste último resultado, duas características principais relacionadas aos objetos circulares foram extraídas: o raio e o centro do círculo do objeto  $c(x, y)$ . Através destas duas características, pode-se gerar um retângulo  $(x, y, w, h)$  que determinou a área a ser segmentada. Para isto, precisou-se determinar os valores para as seguintes variáveis:  $x$  e  $y$ , que são as coordenadas do ponto inicial do retângulo;  $k$  que representa uma constante de multiplicação em relação ao raio e que serve para calcular os valores de  $x$  e  $y$  através das Equações 3.1 e 3.2;  $w$  e  $h$  que representa a largura e a altura do retângulo, respectivamente;  $kw$  e  $kh$  que representam uma constante de multiplicação para determinar a largura e a altura, respectivamente. Estas constantes são utilizadas nas Equações 3.3 e 3.4.

$$x = c(x) - k * r \quad (3.1)$$

$$y = c(y) - k * r \quad (3.2)$$

$$w = kw * r \quad (3.3)$$

$$h = kh * r \quad (3.4)$$

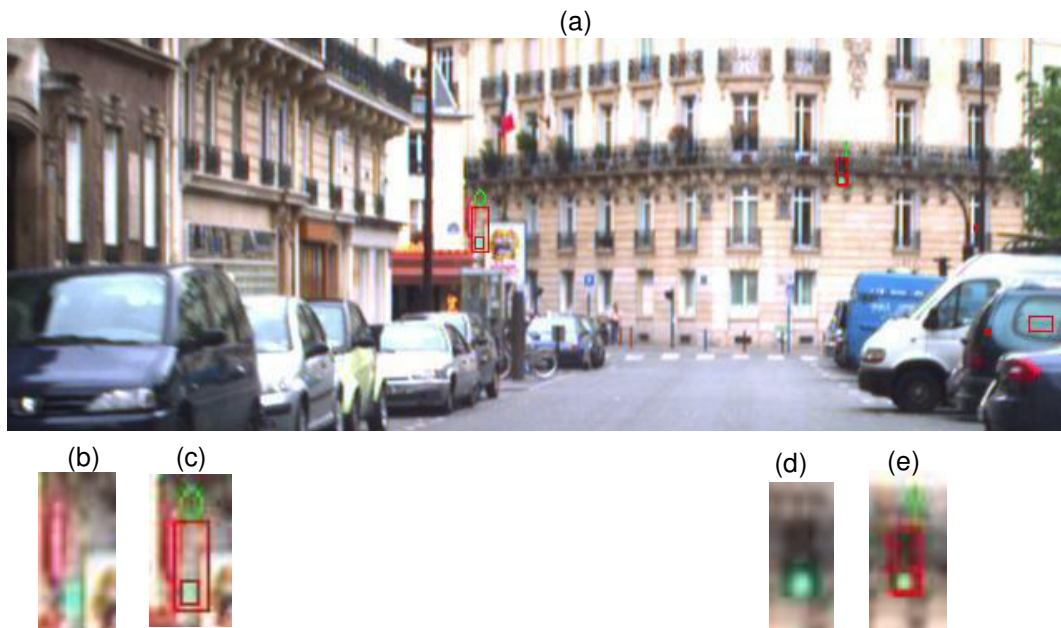
Estas equações, no modo como se apresentam estão prontas para gerar os retângulos para os semáforos com vermelho acionado. No caso da obtenção dos retângulos para os semáforos com luz verde acesa, é necessário fazer uma modificação na Equação 3.2 para obter-se o valor de  $y$ :

$$y = (c(y) + r) - h \quad (3.5)$$

No final desta etapa, vários objetos retangulares foram obtidos, entre estes, os semáforos. Um exemplo pode ser visto na Figura 3.6, onde são apresentados os retângulos que determinam a área correspondente ao corpo do semáforo.

Na Figura 3.6 (a), foram detectados dois objetos tipo semáforo. Na Figura 3.6 (d), extraída da Figura 3.6 (a), é apresentado o semáforo e a Figura 3.6 (e) apresenta o retângulo na cor vermelha delimitando o corpo do semáforo. Enquanto na Figura 3.6 (b) apresenta um falso positivo e na Figura 3.6 (c) o retângulo delimitando esse falso positivo na cor vermelha.

Figura 3.6 Detectando semáforo



Fonte: Elaborado pelo autor.

### 3.5 EXTRAÇÃO DE CARACTERÍSTICAS

Após a segmentação do retângulo, este passou pelo processo de extração de características, com objetivo de criar um vetor de informações para uso na fase de classificação.

Os métodos selecionados neste trabalho estão relacionados às características de cor, forma e textura, apresentadas na Seção 2.4.

Para extrair as características de textura, foi escolhido o método dos descritores de Haralick (1979). Esta técnica obteve 14 características, porém, apenas 13 foram utilizadas, pois a 14ª característica tem um alto custo computacional. Como essas características são extraídas de imagens em escala de cinza, optou-se por imagens no modelo de cor HSV, utilizando apenas o canal H (*Hue* ou matiz).

Junto a essas 13 características de Haralick, foram adicionadas seis características. Embora estas seis características não tenham sido testadas quanto ao custo computacional, acredita-se que elas não têm forte impacto no desempenho do classificador. Estas características estão divididas em grupos chamados de extensão 1, 2 e 3:

- Extensão 1: está relacionada à cor do *pixel* do foco de luz do semáforo e calcula a média ( $meanH$ ) e o desvio padrão ( $stddevH$ ) dos *pixels* relacionados, utilizando o canal H do modelo de cor HSV, que podem ser

calculados pelas Equações 2.44 e 2.46;

- Extensão 2: está relacionado às características de forma do círculo, como o raio ( $r$ ) do círculo, área ( $A$ ) do círculo e a circularidade ( $c$ );
- Extensão 3: está relacionada ao posicionamento do círculo dentro da região do corpo do semáforo em um plano bidimensional, que é calculada através das variáveis  $yc$  e  $rows$ , onde  $yc$  é a variável que guarda a coordenada  $y$  do centro do círculo correspondente ao foco de luz, enquanto a variável  $rows$  está relacionada à quantidade de linhas que as bordas da região do semáforo ocupam em um plano bidimensional.

Após serem calculadas as características de cada objeto, elas são armazenadas no vetor de características e depois é verificada se a base está balanceada, caso não esteja, será aplicada um método para balanceamento de dados.

### 3.6 BALANCEAMENTO DE DADOS

Para fazer o balanceamento da base de dados adotada para este trabalho, foi utilizado o *software* Weka versão 3.7.12 (HALL *et al.*, 2009), de uso sob licença GNU (General Public License – Licença Pública Geral). Este *software* é bastante utilizado por pessoal da área de inteligência artificial com o objetivo de estudos de aprendizado de máquina.

Para verificar e solucionar problemas de desbalanceamento de dados, foram aplicadas duas técnicas, uma relacionada a abordagens *undersampling* (*Spread subsample*) e outra relacionada a abordagens de *oversample* (SMOTE).

Os objetivos para realização destes testes estão selecionados, primeiramente em realizar o balanceamento da base de dados e segundo, verificar qual algoritmo apresenta melhor desempenho, com o classificador escolhido para este trabalho. Assim para esses testes foram utilizadas as seguintes métricas: taxa de classificações corretas, taxa de classificações incorretas, precisão, recall e área sobre a curva ROC (AUC ROC).

Para as realizações dos testes foram definidas uma taxa de confiança de 95% e a validação cruzada com 10 repetições, como método de validação.

Depois do balanceamento da base, as características presentes neste foram apresentadas a RNA para classificação do estado do semáforo.



### 3.7 REDES NEURAIAS ARTIFICIAIS

Para realizar o reconhecimento dos padrões contidos no vetor de características, foi criada uma rede neural artificial (RNA) do tipo MLP (Perceptron de Múltiplas Camadas), cujas características encontram-se abaixo:

- Arquitetura baseada em três camadas, sendo uma camada de entrada, uma camada escondida e uma camada de saída;
- A quantidade de neurônios na camada de entrada foi definida pelo tamanho do vetor característico, sendo neste caso 20 unidades;
- A quantidade de neurônios na camada escondida foi definida em 45 unidades, por apresentar o melhor resultado, após experimentos realizados variando-se esta quantidade de três até 150;
- A quantidade de neurônios na camada de saída foi alinhada à quantidade de classes definidas para este projeto, no caso três classes: classe 0 (vermelho acionado), classe 1 (verde acionado) e classe 2 (não semáforos). A classe 2, compreende todo objeto que não é classificado como classe 0 e nem classe 1;
- A função de ativação dos neurônios das camadas escondida e de saída da rede foi a sigmoide.

O algoritmo de aprendizagem selecionado para este projeto foi *backpropagation*, por ser um algoritmo frequentemente utilizado pelos projetistas de rede. E como critério de parada do treinamento, definiu-se em 1000 a quantidade máxima de iterações ou um erro quadrático médio de 0,005.

### 3.8 CONSIDERAÇÕES FINAIS

Durante a fase de processamento de imagens, as dificuldades encontradas estavam relacionadas à segmentação dos objetos contidos na imagem. Os algoritmos testados, nesta fase, necessitavam de limiares adaptativos que mudavam a cada nova imagem de entrada, devido à complexidade das imagens por possuírem um número razoável de objetos e por causa dos seus ambientes dinâmicos.

Na literatura é comum encontrar citações sobre a segmentação probabilística, para resolver problemas de segmentação de objetos em imagens com ambientes complexos. Através de testes feitos com algoritmos que implementavam segmentação probabilística, foi possível chegar ao algoritmo de histograma retroprojetado, que proporcionou resultados

interessantes.

Com a adoção deste algoritmo, surgiram outras dificuldades, entre elas encontrar o modelo de cor como mencionado na metodologia, foram testados alguns modelos até chegar-se ao modelo HSV, que juntamente com o algoritmo de histograma retroprojetado, obtiveram-se os melhores resultados.

A outra dificuldade encontrada estava relacionada à seleção da quantidade de canais do modelo de cor escolhido. Para chegar a esta decisão, foram realizados testes com o algoritmo básico de histograma retroprojetado e a cada teste realizado, foram surgindo necessidades de modificações no algoritmo. Com os resultados promissores destas modificações, foi-se desenvolvendo o algoritmo de histograma retroprojetado modificado. Uma diferença básica entre esses algoritmos, o tradicional e o modificado, consiste no fato de o modificado ter como entrada a imagem, que é comum aos dois algoritmos, mais um objeto de referência representativo das luzes do semáforo, que serão segmentadas da imagem de entrada.

Ao sanar estas dificuldades, observou-se que o algoritmo necessitava de configurações relacionadas aos limiares do canal de cor, então outras baterias de testes foram realizadas com o propósito de buscar o limiar que pudesse proporcionar um maior número de objetos segmentados. Vários limiares foram testados, porém, a escolha do limiar usado foi baseada em resultados que pudessem manter um equilíbrio entre o maior número de objetos verdadeiros positivos e o menor número de objetos falsos positivos, pois alguns limiares proporcionavam resultados com um grande número de objetos segmentados, porém, o número de falsos positivos era demasiadamente grande, que conseqüentemente provocava elevação no tempo de processamento.

Com a escolha do limiar para os canais de cores, ainda era significativa a quantidade de falsos positivos, o que se fez necessária à aplicação de redução de falsos positivos. Para esta tarefa, foram utilizadas duas técnicas de segmentação de imagem: Otsu e *Adaptive Threshold*. Estas técnicas, juntamente com filtros morfológicos e componentes conexos, serviram para filtrar objetos baseados em formas circulares.

A segmentação iniciou-se através do algoritmo de histograma retroprojetado modificado, chegando até a redução de falsos positivos. Assim, conseguiu-se chegar até os possíveis focos de luzes dos semáforos, porém a segmentação só finalizou quando se conseguiu segmentar o corpo do semáforo. Neste momento, surgiram as dificuldades relacionadas ao corpo dos semáforos, que se apresentaram em tons escuros e muitas vezes

eram facilmente confundidos com objetos semelhantes. A busca para encontrar a técnica que pudesse superar estas dificuldades foi cessada através de citações de trabalhos na literatura, que utilizavam segmentação baseada em características de proporcionalidade. Então foram testadas equações que pudessem encontrar a delimitação do corpo do semáforo, baseando-se na posição e no raio da luz acionada. A equação desenvolvida sofreu modificações para se adaptar aos dois estados do semáforo utilizados neste trabalho, sendo o primeiro estado quando a luz vermelha está acionada (a luz encontra-se na parte superior do semáforo) e o segundo estado é quando a luz verde está acionada (a luz se encontra na parte inferior do semáforo).

Após a segmentação do corpo do semáforo, a etapa seguinte foi a extração de características. Nesta etapa, buscou-se selecionar características que pudessem ser utilizadas pela rede neural, para obterem-se resultados semelhantes ou superiores aos citados na literatura.

As dificuldades encontradas nesta seleção estavam ligeiramente condicionadas aos padrões de cor e forma, então se chegou aos descritores de Haralick, porém os resultados inicialmente se apresentavam abaixo do esperado. Para contornar esta nova dificuldade, buscaram-se novas alternativas na literatura, até o momento em que foram encontrados trabalhos de autores que mencionavam modificação dos descritores de Haralick. Dentre as modificações citadas, uma delas mencionava a inclusão ou exclusão de alguns descritores, através destas ideias foi possível construir um vetor de características para este trabalho, que conseguiu atender aos resultados esperados.

Com o vetor de característica pronto, ainda foi possível notar no trabalho uma base de dados desbalanceada, devido à quantidade de falsos positivos serem bastante superior aos casos de verdadeiros positivos. Este tipo de dificuldade já vem sendo estudada, e foi possível encontrar algumas técnicas para solucionar este tipo de problema. Estas técnicas mencionadas na metodologia foram encontradas já implementadas em *software*. Neste trabalho, foi interessante utilizar estas implementações existentes e fazer apenas as comparações para descobrir quais das técnicas teriam o melhor resultado. Com esta postura, evitaram-se problemas que poderiam ser ocasionados pela tarefa de uma nova implementação destas técnicas.

Inicialmente, a rede neural foi implementada em três *softwares*: OpenCV, Weka e MatLab. A rede confeccionada em OpenCV muitas vezes apresentava erros de implementação e os resultados de classificação distanciavam-se dos resultados apresentados

pelos outros *softwares*, por este motivo, optou-se por abandonar a implementação da rede em OpenCV.

As dificuldades relacionadas à escolha da arquitetura da rede neural foram resolvidas com testes exaustivos, que podem ser visualizados na metodologia deste trabalho. Dentre os problemas encontrados na escolha das métricas para validação, o maior foi construir a curva ROC, pois a curva construída neste trabalho relacionava-se a três classes distintas, o que não é comum na literatura.

## 4 EXPERIMENTOS

Após o desenvolvimento da metodologia, foram executados procedimentos com a base de imagens, para estabelecer resultados consistentes a respeito desta metodologia.

As 7.201 imagens selecionadas na base adotada foram agrupadas em três classes: verde, vermelha e não semáforo. A classe verde (2.594 casos) representa o conjunto de semáforos em estado verde, ou seja, “Siga”. A classe vermelha (4.549 casos) é o conjunto de semáforos em estado vermelho, ou seja, “Pare”. A terceira classe representa semáforos com estado amarelo acionado e outros casos não relacionados às duas primeiras classes.

### 4.1 DETECÇÃO DOS SEMÁFOROS.

O experimento com a base de dados iniciou-se pela fase de detecção (Tabela 4.1). Nesta, são apresentados a quantidade de padrões verdadeiros detectados, seu valor percentual e a média de detecção de objetos por *frame*, ressaltando que há no máximo dois semáforos verdadeiros por *frame*. A última coluna a direita informa o número médio de objetos encontrados em um *frame*. Para esta detecção, foi utilizado o algoritmo de histograma retroprojetado modificado, mencionado no Capítulo 3.

Tabela 4.1 Estimativas de detecção.

Cor	Semáforos	TPR	Acurácia	Falsos Positivos Médios por Imagem
Verde	2594	2579	99,57%	8
Vermelho	4549	4545	99,91%	10

Na Tabela 4.1, a coluna “Cor” representa a cor da luz acesa do semáforo nas imagens. A coluna “Semáforos” corresponde ao número de semáforos existentes na base em cada caso. A coluna “TPR” representa os semáforos detectados pelo algoritmo. A coluna “Acurácia” descreve a razão em percentual entre o número de semáforos da base e a quantidade detectada. Por fim, a coluna da direita indica a quantidade média de objetos detectados em cada imagem de entrada submetida ao algoritmo de detecção. Este algoritmo apresentou bons resultados de acurácia (99 %). Depois, foi feita a extração de características, que gerou uma base de dados, separada por três classes mencionadas na metodologia.

### 4.2 PREPARAÇÃO DOS DADOS PARA APRENDIZADO

Depois da detecção, foi realizada a preparação da base de dados para a execução da

fase de aprendizado. A primeira parte da preparação consistiu na separação dos dados para o treinamento e para a validação. A divisão é mostrada na Tabela 4.2. Vale ressaltar que os valores referentes à divisão da base são valores aproximados, pois não é possível usar uma quantidade fracionária de imagens.

Tabela 4.2 Divisão da base de dados

Cor	Base	Aprendizado (80%)	Validação (20%)
Verde	2579	2060	519
Vermelho	4545	3630	915
Não semáforos	56330	45060	11270

A Tabela 4.2 mostra um exemplo claro de uma base desbalanceada, pois a quantidade do caso “Não semáforo” é muito maior que os demais casos. Assim, primeiro o balanceamento de dados foi providenciado aplicando duas técnicas: *spread* e SMOTE. Nesta fase, foi feito o balanceamento apenas na parte relacionada ao aprendizado, onde na Tabela 4.4 são apresentados os valores destes testes. As colunas SMOTE e *Spread* apresentam os números das classes após a aplicação das referidas técnicas.

Tabela 4.3 Balanceamento por SMOTE e *Spread*

Cor	Aprendizado (80%)	SMOTE	Spread
Verde	2060	45060	2060
Vermelho	3630	45060	2060
Falsos Positivos	45060	45060	2060

Os resultados das métricas usadas nos experimentos são apresentados na Tabela 4.4 a seguir. Com base nos resultados nessa tabela, as métricas precisão, *recall*, e AUC ROC apresentaram valores idênticos. Já em relação às classificações corretas e incorretas, os resultados têm diferenças bastante pequenas, apontando os melhores resultados para a técnica de SMOTE, o que resultou na adoção desta técnica.

Tabela 4.4 Desempenho do balanceamento

Balanceamento	Classificações Corretas(%)	Classificações Incorretas(%)	Precisão	<i>Recall</i>	AUC ROC
SMOTE	98,30	1,70	0,99	0,99	1,00
Spread	97,63	2,37	0,99	0,99	1,00

### 4.3 INCLUSÃO DE ATRIBUTOS

A inclusão de atributos foi realizada com o intuito de melhorar o percentual de acertos de classificação para a referida base, para tanto foram acrescentados novos atributos no vetor de características, conforme está descrito na Seção 3.5, que estão relacionadas à cor e

à forma e são nomeadas como extensões 1, 2 e 3.

O experimento foi realizado através do *software* Weka (HALL *et al.*, 2009), conforme as configurações abaixo:

- base balanceada através da técnica de SMOTE;
- foi escolhido o *split* 70 para dividir a parte referente ao aprendizado em treino e teste;
- a coluna atributos está relacionada aos atributos de Haralick, bem como suas extensões descritas na Seção 3.5;
- 13 atributos somente Haralick; 15 atributos juntando Haralick e a extensão 1; 18 atributos juntando Haralick e as extensões 1 e 2; e 20 atributos juntando Haralick e as três extensões;
- cada valor de atributo representa uma base de dados que foi testada;
- as métricas utilizadas para este experimento foram TPR (taxa de verdadeiros positivos), FPR (Taxa de falsos positivos), precisão, *recall* e AUC ROC (área sob a curva ROC);
- são considerados os valores médios para cada métrica;
- cada teste foi executado três vezes em cada base;
- são guardados somente os melhores resultados.

De acordo com a Tabela 4.5, podem ser feitas algumas deduções:

- o maior valor médio para TPR foi obtido com 20 atributos;
- o menor valor médio para FPR foi obtido com 20 atributos;
- o maior valor médio de precisão foi obtido com 20 atributos;
- o maior valor médio para AUC ROC são com 18 e 20 atributos.

Baseando-se nestas inferências, pode-se concluir que a inclusão dos atributos melhorou gradativamente a qualidade do classificador. Os melhores resultados apontam para a inclusão de todos os atributos, por isso o vetor característico foi dotado de 20 atributos.

#### 4.4 TREINAMENTO DA REDE NEURAL

A parte da base referente ao aprendizado é submetida às novas subdivisões que foram utilizadas no treinamento da rede. As subdivisões são treino e validação. Foi adotada a metodologia de *splits*, que consiste em dividir uma base de dados em duas partes, uma parte

dos dados é utilizada para o treinamento e o restante da base é utilizado para testar o desempenho do treinamento. O *split* é dado em porcentagem, então quando é mencionado *split 70*, significa que 70% da base são utilizadas para treino e os 30% restantes são utilizadas para teste de desempenho do treino. Na Tabela 4.6, são apresentados os *splits* escolhidos para realizar treinamento da rede.

Tabela 4.5 Inclusão de atributos

Classe	Atributos	TPR (%)	FPR (%)	Precisão (%)	Recall (%)	AUC ROC (%)
Vermelho	13	92,7	1,9	96,1	92,7	98,6
Verde	13	94,4	6,0	88,7	94,4	97,5
Falsos Positivos	13	91,4	2,9	94,1	91,4	97,6
Média		92,8	3,6	93,0	92,8	97,9
Vermelho	15	98,6	0,3	99,4	98,6	99,8
Verde	15	98,3	2,0	96,1	98,3	99,5
Falsos Positivos	15	95,9	1,4	97,2	95,9	99,2
Média		97,6	1,2	97,6	97,6	99,5
Vermelho	18	98,8	0,2	99,7	98,8	99,8
Verde	18	97,6	1,0	98,1	97,6	99,8
Falsos Positivos	18	98,0	1,7	96,7	98,0	99,7
Média		98,1	0,9	98,1	98,1	<b>99,8</b>
Vermelho	20	98,7	0,2	99,7	98,7	99,8
Verde	20	97,7	0,8	98,4	97,7	99,8
Falsos Positivos	20	98,3	1,7	96,7	98,3	99,8
Média		<b>98,3</b>	<b>0,9</b>	<b>98,3</b>	<b>98,3</b>	<b>99,8</b>

Tabela 4.6 Divisão por *splits*

SMOTE	20%	30%	40%	50%	60%	70%	80%
Treino	9.012	13.518	18.024	22.530	27.036	31.542	36.048
Teste	36.048	31.542	27.036	22.530	18.024	13.518	9.012
Total	45.060	45.060	45.060	45.060	45.060	45.060	45.060

Para encontrar os melhores resultados na fase de treinamento, a rede neural sofreu diversas alterações em sua arquitetura, a fim de encontrar uma configuração que apresente o melhor desempenho. Para isso, foram feitos muitos testes, variando a quantidade de neurônios na camada escondida, a taxa de aprendizagem e a quantidade de épocas.

A Tabela 4.7 mostra um exemplo da matriz de confusão na fase de treino, utilizando o *split 70*, com uma rede neural com 45 neurônios na camada escondida. A Tabela 4.8 apresenta alguns resultados selecionados, para mostrar o desempenho da rede alterando a quantidade de neurônios na camada escondida (*Hidden*). Para estes testes, foram levados em conta os percentuais relacionados às métricas: sensibilidade (SEN) e especificidade (ESP) da matriz de confusão.



Tabela 4.7 Matriz de confusão - treino

Real	Classificação			Totais	SEN
	Vermelho	Verde	Não Semáforos		
Vermelho	13.432	0	62	13.494	0,995
Verde	31	13.369	200	13.600	0,983
Não Semáforos	55	149	13.256	13.460	0,985
Totais	13.518	13.518	13.518	Média SEN	<b>0,988</b>
ESP	0,994	0,989	0,981	Média ESP	<b>0,988</b>

Tabela 4.8 Teste com a camada escondida

Hidden	20		30		45		50		100	
Classes	SEN	ESP	SEN	ESP	SEN	ESP	SEN	ESP	SEN	ESP
Verde	0,994	0,991	0,996	0,992	0,995	0,994	0,994	0,991	0,995	0,992
Vermelho	0,978	0,985	0,982	0,988	0,983	0,989	0,976	0,984	0,982	0,987
Falsos Positivos	0,979	0,975	0,982	0,980	0,985	0,981	0,979	0,973	0,982	0,980
Média	0,984	0,984	0,987	0,987	<b>0,988</b>	<b>0,988</b>	0,983	0,983	0,987	0,987

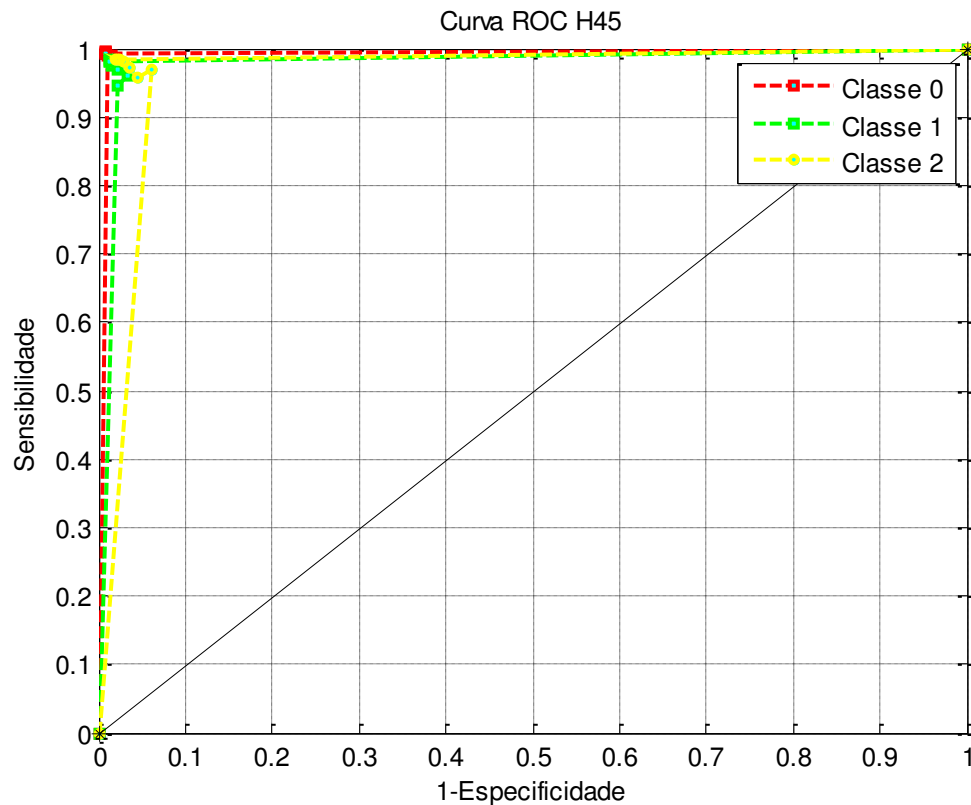
Com base nos resultados da Tabela 4.8, é verificado que o melhor resultado médio é obtido com a rede com 45 neurônios na camada escondida. Estes testes foram feitos de forma incremental, com a quantidade de neurônios na camada escondida sendo 7, 8, 9, 10, 14, 15, 16, 20, 30, 40, 45, 50, 100. Utilizou-se para tanto o *software* Matlab R2013a com o *toolbox* *nnprtool* (BEALE; HAGAN; DEMUTH, 2015).

Com base na melhor arquitetura escolhida para treinamento, ver Tabela 4.8, foram realizados os treinamentos utilizando os *splits* da Tabela 4.6 para construir a curva ROC, apresentada na Figura 4.1. Os dados utilizados são apresentados na Tabela 4.9, onde o eixo Y representa a sensibilidade, enquanto o eixo X representa 1-especificidade.

Tabela 4.9 Tabela de dados da curva ROC

Classe	Vermelho		Verde		Falsos Positivos	
Splits	SEN	ESP	SEN	ESP	SEN	ESP
	0,000	1,000	0,000	1,000	0,000	1,000
20	0,991	0,989	0,945	0,978	0,970	0,940
30	0,990	0,986	0,960	0,967	0,958	0,954
40	0,994	0,992	0,976	0,985	0,980	0,972
50	0,995	0,992	0,981	0,988	0,983	0,979
60	0,992	0,991	0,970	0,979	0,972	0,965
70	0,995	0,992	0,983	0,988	0,984	0,980
80	0,994	0,993	0,980	0,988	0,983	0,976
	1,000	0,000	1,000	0,000	1,000	0,000
AUC	0,9915		0,9786		0,9786	

Figura 4.1 Curva ROC



Fonte: Elaborado pelo autor.

De acordo com a Figura 4.1, classe 0 refere-se a classe dos semáforos vermelhos, a classe 1 refere-se aos semáforos verdes e a classe 3 refere-se aos não semáforos.

#### 4.5 TESTE DE VALIDAÇÃO

Foi aplicado balanceamento na parte da base referente à validação a Tabela 4.10, mostra o resultado do balanceamento, utilizando SMOTE.

Tabela 4.10 Balanceamento SMOTE - Validação

Cor	Validação (20%)	SMOTE
Verde	519	11.270
Vermelho	915	11.270
Falsos Positivos	11.270	11.270

A Tabela 4.11 apresenta os resultados referentes ao teste de validação usando a rede com 45 neurônios na camada escondida e split 70. Esta tabela também apresenta os resultados referentes à sensibilidade (SEN), à especificidade (ESP), à precisão (P) e ao recall (R).

Tabela 4.11 Matriz de confusão – Validação

Real	Classificação			Totais	SEN	Pecision	Recall
	Vermelho	Verde	Não semáforos				
Vermelho	11.169	6	51	11.226	0,995	0,991	0,995
Verde	73	11.102	159	11.334	0,980	0,985	0,980
Falsos Positivos	28	162	11.060	11.250	0,983	0,981	0,983
Totais	11.270	11.270	11.270	Média	<b>0,986</b>	<b>0,986</b>	<b>0,986</b>
ESP	0,991	0,985	0,981	Média ESP	<b>0,986</b>		

#### 4.6 DISCUSSÃO DOS RESULTADOS

Este trabalho foi dividido em quatro partes bem definidas: detecção, extração de características, treinamento e teste de generalização. A fase de detecção é marcada pelos algoritmos de processamento de imagens, e tem como objetivo detectar e segmentar os semáforos ou objetos semelhantes. Apesar de existirem três estados para os semáforos, a metodologia desenvolvida só trabalha com dois estados: o verde e o vermelho. O amarelo, neste trabalho é tratado com não semáforo. O resultado desta fase são três classes de dados com características distintas e com quantidades de objetos por classe bastante diferenciada, ver Tabela 4.1.

Nesta fase, um dos fatores que contribuíram para o desbalanceamento foi a circularidade do objeto. Os trabalhos, vistos na literatura, apontam para circularidades bem próximas de um. Neste trabalho, foi adotada uma circularidade bem pequena, próxima de zero, para conseguir maximizar as detecções dos semáforos, o que culminou nesse alto número de falsos positivos. As imagens submetidas na detecção possuem um ou dois semáforos. Os resultados do processo de detecção geraram quantidade média de regiões de interesse em torno de 8 a 10 regiões por imagem.

Na fase de extração de características, inicialmente foram selecionados 13 dos 14 atributos de Haralick, porém, em resultados preliminares em relação à precisão e ao *recall*, obteve-se uma média de 93% e 92,8%, respectivamente. Para melhorar estes resultados, foram incluídos novos atributos, o que proporcionou a elevação da precisão e do *recall* para uma média de 98,3%, que pode ser visualizado na Tabela 4.5.

De acordo com os resultados obtidos na fase de detecção, surge o problema da base de dados desbalanceada. Para solucionar, foram utilizadas duas técnicas de balanceamento: SMOTE e *Spread*. Foram comparados os dois resultados (Tabela 4.4), onde se verificou que a diferença entre as duas técnicas em percentuais é pequena. A diferença das classificações corretas entre SMOTE e *Spread* é 0,67% e a diferença entre as classificações incorretas

também chega a 0,67%. Já para as demais métricas como precisão, *recall* e AUC ROC, não houve diferença. Mesmo a diferença percentual sendo pequena, o resultado da técnica adotada (SMOTE) para o balanceamento é interessante, visto que os resultados foram elevados de 97% de classificação corretas para 98%, ver Tabela 4.4.

A fase de inclusão de atributos e a fase de balanceamento complementam-se, visto que os testes feitos na fase de inclusão de atributos foram realizados com a base já balanceada.

Na fase de treinamento da rede neural, primeiro foi preciso descobrir uma arquitetura eficiente para a rede neural, a fim de que ela possa obter o melhor resultado no teste final. Para chegar-se à arquitetura apropriada, foram executados vários testes, alterando a quantidade de neurônios da camada escondida, a taxa de aprendizagem e aumentando a quantidade de épocas. Porém, os dados apresentados na Seção 4.4 referem-se apenas aos resultados relacionados à quantidade de neurônios na camada escondida, pois uma mudança nessa quantidade impactava mais significativamente os resultados do que os outros e conseqüentemente afetava a escolha da arquitetura. A melhor arquitetura encontrada continha 45 neurônios na camada escondida (H45), pois se chega a um ponto em que mesmo aumentando a quantidade de neurônios não é possível ver a elevação das métricas mostradas na Tabela 4.8. Cada teste realizado foi repetido três vezes, para haver mais confiança nos resultados.

Com a curva ROC apresentada na Figura 4.1, verificou-se que a classe vermelha obteve o melhor valor de AUC com 0,9915, enquanto que as demais obtiveram AUC de 0,9786. Com esses valores, algumas afirmações podem ser feitas, de acordo com a Tabela 2.8, pode-se afirmar que a qualidade deste classificador é excelente. Os valores mais altos estão relacionados à cor vermelha, tanto no treino como na validação.

No teste de validação, onde se testou quanto a rede aprendeu e quanto ela pôde generalizar, quando são apresentados casos que ainda não foram passados pela classificação, a rede neural apresentou bons resultados, com taxas de reconhecimento em torno dos 98%.

A metodologia deste trabalho foi desenvolvida com o uso dos seguintes *softwares*: biblioteca OpenCV e compiladores C/C++, para o processamento de imagens; Weka, para os testes de inclusão de atributos e balanceamento de dados; e Matlab 2013a (BEALE; HAGAN; DEMUTH, 2015), para os testes na fase de treinamento e validação com a rede neural.

#### 4.7 COMPARAÇÃO COM TRABALHOS RELACIONADOS

Dentre os trabalhos relacionados encontrados na literatura e descritos na Seção 1.2, foram selecionados seis trabalhos para realizar uma comparação de resultados. A Tabela 4.12 mostra a comparação dos resultados, utilizando as métricas precisão e *recall*, apesar de existirem mais métricas para avaliação de desempenho dos classificadores, a maioria dos trabalhos selecionados apresentam somente essas duas.

Sobre a tabela abaixo, pode-se verificar que:

- Os melhores resultados para as duas métricas foram obtidos por Chen, Shi e Zou (2014);
- O trabalho proposto fica entre os melhores em relação à precisão e *recall*.
- Combinando as duas métricas, é possível verificar que o trabalho aqui apresentado fica apenas atrás do trabalho publicado por Chen, Shi, e Zou (2014);

Tabela 4.12 Comparação dos Resultados

<b>Algoritmos</b>	<b>Autores</b>	<b>Precisão</b>	<b>Recall</b>
1.	de Charette e Nashshibi (2009)	95,38%	98,41%
2.	Li, <i>et al.</i> (2011)	97,00%	97,00%
3.	Wang, <i>et al.</i> (2011)	97,20%	96,50%
4.	Diaz-Cabrera, Cerri e Medici (2015)	98,88%	97,15%
5.	Chen, Shi e Zou (2014)	99,18%	98,96%
6.	Sathiya, Balasubramanian e Priya (2014)	97,00%	96,00%
7.	Trabalho proposto	98,60%	98,60%

Embora o trabalho de Chen, Shi e Zou (2014) apresente resultados superiores, vale ressaltar que a base de dados usada foi diferente da base deste trabalho, assim merece uma investigação para verificar se o trabalho proposto teria uma precisão maior ou menor se fosse utilizada a outra base.

## 5 CONCLUSÃO

Este trabalho realizou a detecção de semáforos utilizando técnicas de processamento de imagens e redes neurais artificiais, juntamente com descritores de cor, forma e textura, fizeram o reconhecimento de semáforos em ambientes urbanos, especificamente para os estados verde e vermelho.

A metodologia apresentada neste trabalho está dividida em duas etapas principais: detecção e reconhecimento. A etapa de detecção é marcada pelo uso de técnicas de processamento de imagem para segmentar os pontos de luzes dos semáforos, onde se destacou a técnica de segmentação que utiliza o histograma retroprojetado modificado. Esta técnica trouxe bons resultados de detecção, porém, gerou uma quantidade significativa de falsos positivos, o que demandou a aplicação de filtros de formas, relacionados à área e à circularidade, para reduzir a taxa de falsas detecções.

Já na etapa de reconhecimento, usou-se rede neural artificial. Para os padrões de entrada desta rede, foram selecionados descritores de cor, forma e textura, onde a configuração deste conjunto foi realizada através de testes experimentais.

Ainda nesta etapa, foram utilizadas técnicas de balanceamento de dados, com a finalidade de melhorar o desempenho do classificador utilizado.

O problema de trabalhar com classes desbalanceadas pode interferir no aprendizado de máquina, assim muitos algoritmos vêm sendo estudados para solucionar este tipo de problema. Neste trabalho, utilizou-se duas técnicas, onde se pôde comparar a eficiência destas, o que proporcionou uma melhoria nos resultados finais. As técnicas utilizadas estão relacionadas às abordagens de sobreamostragem e subamostragem, e apresentaram resultados semelhantes nos testes, porém, a técnica de sobreamostragem mostrou-se levemente superior para este trabalho.

O conjunto redes neurais do tipo MLP e características relacionadas à cor, à forma e à textura, apresentou bom desempenho no reconhecimento de padrões. A avaliação desse conjunto para reconhecimento de semáforos, em comparação com outras técnicas na literatura, apresentou resultados bastante positivos, pois, conseguiu-se resultados médios de 98% de reconhecimento. O que se conclui que RNA do tipo MLP para reconhecimento de semáforos é um excelente classificador.

Para validar esta metodologia, foram utilizadas métricas relacionadas à análise ROC. Através dos resultados obtidos com esta análise, pode-se verificar que a metodologia conseguiu atingir o objetivo proposto e que os resultados obtidos são interessantes, quando comparados com os trabalhos encontrados na literatura.

## 5.1 CONTRIBUIÇÕES DESTE TRABALHO

Este trabalho apresentou uma nova metodologia para reconhecimento de semáforos. Na metodologia desenvolvida, utilizou-se histograma retroprojetado para detecção e redes neurais com descritores de Haralick estendido para o reconhecimento dos semáforos. As contribuições podem ser listadas como:

- uso do histograma retroprojetado modificado para detectar formas circulares, utilizando o modelo de cor HSV, que apresentou taxa de detecção em média de 99% de acertos e a média de falsos positivos 9 por imagens. Vale ressaltar que estes reconhecimentos foram realizados em imagens de ambientes urbanos que apresentam uma grande quantidade de objetos semelhantes aos semáforos;
- uso dos atributos de Haralick e a inclusão de novos atributos, como descritores geométricos e informações de cor para confecção do vetor de características. Estes atributos apresentaram bons resultados para uso com classificadores probabilísticos;

## 5.2 TRABALHOS FUTUROS

Uma primeira sugestão de trabalho futuro consiste em analisar o tempo e o custo computacional do conjunto de descritores, 13 características de Haralick mais os descritores adicionais de cor e forma.

Seria interessante também realizar a comparação anterior considerando os 14 descritores de Haralick, de modo a verificar o impacto desse último descritor no custo computacional.

Outra sugestão de trabalho tem o objetivo de otimizar os algoritmos para cálculo dos descritores de Haralick, pois os algoritmos utilizados neste trabalho apresentaram um tempo de processamento grande, o que significa que não são ideais para aplicações em tempo real.

Pretende-se investigar outros descritores de forma e informação de cor, que possam apresentar tempos de processamento menor que os utilizados neste trabalho, assim sendo outra possibilidade de continuação.

Outra sugestão de trabalho futuro seria incluir outros tipos de semáforos no estudo, visto que estes, dependendo das regiões, apresentam formas variadas, como: focos do tipo seta, semáforos com foco circular nas posições vertical e horizontal, entre outros.

Sobre balanceamento de dados, outra possibilidade de trabalho futuro é trabalhar com as três cores dos semáforos, pois a quantidade de semáforos com estado amarelo é bastante pequena na base utilizada.



## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ALMAGAMBETOV, A.; VELIPASALAR, S.; BAITASSOVA, A. Mobile Standards-Based Traffic Light Detection in Assistive Devices for Individuals with Color-Vision Deficiency. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 3, p. 1305–1320, jun. 2015.

ANTONIO, A., *et al.* Algoritmos para reconhecimento de padrões. p. 129–145, 2002.

BAGGIO, DANIEL LÉLIS; ESCRIVÁ, DAVID MILLÁN; MAHMOOD, NAUREEN; SHILKROT, ROY; EMAMI, SHERVIN; LEVGEN, KHVEDCHENIA; SARAGIH, J. **Mastering OpenCV with Practical Computer Vision Projects**. [s.l.] Packt open source, 2012. v. 11

BAXES, G. . J. W. & S. **Digital image processing: principles and applications**. NY: John Wiley & Sons, Inc, 1994.

BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. Neural Network Toolbox™ User's Guide. 2015.

BRAGA, A. DE P.; DE CARVALHO, A. L. F.; LUDEMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. 2 ed. Rio de Janeiro: LTC, 2007.

BRASIL, P. T. **Transito brasileiro mata mais que a guerra entre palestina e israel**. Disponível em: <<http://www.transportabrasil.com.br/2014/09/transito-brasileiro-mata-mais-que-guerra-entre-palestina-e-israel/>>. Acesso em: 9 abr. 2015.

CAI, Z.; LI, Y.; GU, M. **Real-time recognition system of traffic light in urban environment**. 2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications. **Anais... IEEE**, jul. 2012. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6291516>>

CHAWLA, N., *et al.* SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, jun. 2002.

CHEN, Q.; SHI, Z.; ZOU, Z. **Robust and real-time traffic light recognition based on hierarchical vision architecture**. 2014 7th International Congress on Image and Signal Processing. **Anais...** IEEE, out. 2014 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7003760>>

COELHO, J. C.; FREITAS, J. A.; MOREIRA, M. E. P. Implantações semaforicas são medidas eficazes para a redução de acidentes de trânsito? O caso de Fortaleza-CE. **XXII Congresso de Pesquisa e Ensino de Transportes**, v. v, 2008.

CONNELL, S. D.; JAIN, A. K. Template-based online character recognition. **Pattern Recognition**, v. 34, n. 1, p. 1–14, jan. 2001.

DE ALMEIDA NETO, A. Aplicações de Múltiplas Redes Neurais em Sistemas Mecatrônicos - Tese de Doutorado. [s.l.] ITA, 2003.

DE CHARETTE, R.; NASHASHIBI, F. **Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates**. 2009 IEEE Intelligent Vehicles Symposium. **Anais...** IEEE, jun. 2009. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5164304>>

DENATRAN, D. N. DE T. **Contran n. 483, volume v, gestão 2011**. Disponível em: <<http://www.denatran.gov.br/resolucoes.htm>>. Acesso em: 24 nov. 2014.

DIAZ-CABRERA, M.; CERRI, P.; MEDICI, P. Robust real-time traffic light detection and distance estimation using a single camera. **Expert Systems with Applications**, v. 42, n. 8, p. 3911–3923, maio 2015.

DUDA, R. O.; HART, P. E.; STOK, D. G. **Pattern Classification**. 2ed. New York: Wiley-Interscience, 2000.

EGAN, J. P. **Signal detection theory and ROC analysis, Series in Cognition and Perception**. New York: USA: Academic Press, 1975.

FILHO, O. M.; NETO, H. V. Processamento Digital de Imagens. **Revista Brasileira de**

**Geofísica**, v. 21/03, n. 1, p. 331, 1999.

GOMES, O. DA F. M. **Microscopia Co-Localizada: Novas Possibilidades na Caracterização de Minérios (Tese de Doutorado)**. [s.l.] Pontifícia Universidade Católica do Rio de Janeiro., 2007.

GOMEZ, A. E., *et al.* **Traffic lights detection and state estimation using Hidden Markov Models**. 2014 IEEE Intelligent Vehicles Symposium Proceedings. **Anais...** Michigan, USA: IEEE, jun. 2014. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6856486>>

GONG, J., *et al.* **The recognition and tracking of traffic lights based on color segmentation and CAMSHIFT for intelligent vehicles**. 2010 IEEE Intelligent Vehicles Symposium. **Anais...** IEEE, jun. 2010. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5548083>>

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [s.l.] Edgard Blucher, 2000.

HALL, M., *et al.* The WEKA data mining software. **ACM SIGKDD Explorations Newsletter**, v. 11, n. 1, p. 10, 2009.

HARALICK, R. M. Statistical and structural approaches to texture. **Anais...** IEEE, v. 67, n. 5, p. 786–804, 1979.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural Features for Image Classification. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 3, n. 6, p. 610–621, nov. 1973.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. New Jersey: Prentice Hall, 1999.

JAIN, A. **Fundamentals of Digital Image Processing**. Inc. Upper ed. NJ, USA: Prentice–Hall, 1989.

JOHN, V., *et al.* **Traffic light recognition in varying illumination using deep learning and saliency map.** 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). **Anais...** Qingdao, China: IEEE, out. 2014

KHAN, I. R.; FARBIZ, F. **A new similarity measure and back-projection scheme for robust object tracking.** 2010 10th International Symposium on Communications and Information Technologies. **Anais...** IEEE, out. 2010

KUO, Y.-C.; PAI, N.-S.; LI, Y.-F. Vision-based vehicle detection for a driver assistance system. **Computers & Mathematics with Applications**, v. 61, n. 8, p. 2096–2100, abr. 2011.

LARA, L. R. A. **Traffic Lights Recognition (TLR) public benchmarks.** Disponível em: <<http://www.lara.prd.fr/benchmarks/trafficlightsrecognition>>. Acesso em: 22 ago. 2015.

LI, Y., *et al.* **Traffic lights recognition based on morphology filtering and statistical classification.** 2011 Seventh International Conference on Natural Computation. **Anais...** IEEE, jul. 2011

OLIVEIRA, M. B. W. **Otimização De Tempos Semafóricos Via Múltiplas Redes Neurais. Dissertação (Mestrado).** [s.l.] Universidade Federal do Maranhão, 2013.

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações.** [s.l.: s.n.].

PITAS, I. **Digital image processing algorithms and applications.** NJ, USA: John Wiley & Sons, 2000.

PRATI, R. C. **Novas abordagens em aprendizado de máquina para a geração de regras, classes desbalanceadas e ordenação de casos - Tese de Doutorado.** [s.l.] USP - São Carlos, 2006.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning Internal Representations by Error Propagation, Parallel Distributed Processing**, Explorations in the Microstructure of Cognition. v. 1, p. 318–362, 1986.

SANTOS, A. P. DE O. Desenvolvimento de Descritores de Imagens para Reconhecimento de Padrões de Plantas Invasoras (Folhas Largas e Folhas Estreitas) - Dissertação de Mestrado. [s.l.] Universidade Federal de São Carlos, 2009.

SATHIYA, S.; BALASUBRAMANIAN, M.; PRIYA, D. V. **Real time recognition of traffic light and their signal count-down timings**International Conference on Information Communication and Embedded Systems (ICICES2014). **Anais... IEEE**, fev. 2014. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7033965>>

SINALIZAÇÃO, T. L. **Sinalização Vertical**. Disponível em: <<http://www.transitolivresinalizacao.com.br/sinalizacao-vertical.html>>. Acesso em: 24 nov. 2014.

SPACKMAN, K. A. Signal detection theory: Valuable tool for evaluating inductive learning. **Proc. Sixth Internat. Workshop on Machine Learning**, p. 160–163, 1989.

VAZ, F. A. Restauração de documentos eletrônicos degradados com binarização adaptativa. **Rios Eletrônica-Revist Científica da**, v. 0, 2011.

WANG, C., *et al.* Robust and Real-Time Traffic Lights Recognition in Complex Urban Environments. **International Journal of Computational Intelligence Systems**, v. 4, n. 6, p. 1383, 2011.

ZADROZNY, B.; LANGFORD, J.; ABE, N. **Cost-sensitive learning by cost-proportionate example weighting**. Third IEEE International Conference on Data Mining. **Anais... IEEE Comput. Soc**, 2003.