



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção e Reconhecimento Automáticos de Sinais de Trânsito

Wallace Bruno Silva de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Bruno Luigi Machiavello

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Coordenador Homero Luiz Piccolo

Banca examinadora composta por:

Prof. Dr. Bruno Luigi Machiavello (Orientador) — CIC/UnB
Prof. Dr. Alexandre Zaghetto — CIC/UnB
Prof. Dr. Eduardo Peixoto — ENE/UnB

CIP — Catalogação Internacional na Publicação

Souza, Wallace Bruno Silva de.

Detecção e Reconhecimento Automáticos de Sinais de Trânsito / Wallace Bruno Silva de Souza. Brasília : UnB, 2015.

157 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. Detecção, 2. Reconhecimento, 3. Sinais de trânsito, 4. SVM,
5. YCbCr

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção e Reconhecimento Automáticos de Sinais de Trânsito

Wallace Bruno Silva de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Bruno Luigi Machiavello (Orientador)
CIC/UnB

Prof. Dr. Alexandre Zaghetto Prof. Dr. Eduardo Peixoto
CIC/UnB ENE/UnB

Prof. Dr. Coordenador Homero Luiz Piccolo
Coordenador do Bacharelado em Ciência da Computação

Brasília, 06 de julho de 2015

Dedicatória

Dedico este trabalho a meu pai Marcondes, à minha mãe Maria, a meu irmão Matheus e à minha namorada Priscilla. Presentes a cada dia, sempre acreditaram em mim e foram determinantes para que eu chegasse até aqui. Amo vocês!

Agradecimentos

Antes de tudo, agradeço a Deus por sempre estar presente em todos os aspectos da minha vida, inclusive na minha caminhada dentro da UnB, e por nunca ter me deixado desistir dos meus sonhos.

Agradeço a meus pais Marcondes Freire e Maria José por sempre terem me apoiado e me ajudado a superar as dificuldades. Obrigado pelos ensinamentos, pelos conselhos, pelas críticas e pelo investimento na minha educação, sem os quais este trabalho não seria possível. Agradeço por serem meus primeiros amigos, por sempre terem tempo de ouvir minhas histórias e por me apoiar incondicionalmente.

A meu irmão e grande amigo Matheus José por discutir todos os assuntos comigo, bem como pelas várias e disputadas partidas de vídeo game, que quase sempre acabaram com vitórias minhas. Obrigado por essa descontração, pela parceria e por constantemente me fazer crescer, na tentativa de ser para você o melhor exemplo possível.

Agradeço à minha namorada Priscilla Bessoni por sempre fazer de tudo para estar comigo, apesar da distância. Pela companhia, pelos beijos e abraços, que nesses últimos três anos sempre me animaram, mesmo em momentos mais difíceis. Obrigado por cada olhar, sorriso e telefonema, pelas opiniões e pela curiosidade sobre tudo.

A minhas tias Arlete, Janine, Juliane, ao meu tio Vilmondes e às minhas avós Filomena e Luíza por torcerem por mim e comemorarem minhas conquistas.

Agradeço excessivamente ao meu professor orientador e amigo Bruno Macchiavello. Sou grato por ter aceitado me orientar, pela sua enorme contribuição para este trabalho, pelas revisões feitas e indicações do que devia ou não ser feito. Obrigado pelo bom humor de sempre, pela paciência para explicar tudo, da melhor maneira possível, e por me esperar para as reuniões que me atrasava. Obrigado pelos seis semestres de monitorias consecutivas, pela disponibilidade e pelos vários e-mails respondidos.

Agradeço ao professor Zaghetto por me ajudar na disciplina de introdução ao processamento de imagens, sanando minhas dúvidas com boa vontade. Ao professor Homero pela amizade e às professoras Cláudia Nalon e Carla Castanho pela atenção e apreço.

A meus amigos que estiveram comigo desde os primeiros jogos de truco na UnB, ainda no segundo semestre de 2010. Obrigado por todas as jogatinas, pelas partidas de futsal, pelos dribles tomados e pelas madrugadas em claro fazendo trabalhos. Merecem destaque Filipe Caldas, Felipe Rodopoulos, Heitor Henrique, Guilherme Alexander, Silas Fernandes, Paulo Afonso PA, Thiago Schmidt, André Belle, Ítalo Batista e o sempre sumido Thiago Santaguida. À Ana Luiza pelo carinho e por ter me apresentado sua irmã.

Aos professores Cláudio Viana e Thales Machado por acreditarem em mim. Às outras pessoas que não lembrei de citar. À Loba que me dá afeto e não pede nada em troca.

Agradeço a todos pelas risadas que me proporcionam e por rirem também das minhas ótimas brincadeiras. MUITÍSSIMO OBRIGADO!

Resumo

Este trabalho propõe um método para detectar e reconhecer, automaticamente, placas de trânsito. A abordagem utilizada para a detecção baseia-se na segmentação por cores no espaço de cores YCbCr. Essa segmentação é feita por meio do processo de limiarização. Desse modo, a obtenção dos limiares envolve a confecção de histogramas para as componentes de crominância Cb e Cr do espaço supracitado. A componente Y armazena informações relacionadas ao brilho e não é utilizada nesse procedimento, pois os limiares são muito sensíveis a variações de luminosidade. Foram ainda estabelecidos alguns critérios para tentar evitar os falsos positivos, isto é, objetos que não são de interesse, mas acabam sendo detectados como placas de trânsito. Exemplos desses critérios são a análise da sobreposição de placas e relação definida entre altura e largura das placas a serem identificadas. Complementarmente, propõe-se uma maneira simples de identificar a forma dos sinais candidatos, mediante a análise de características referentes à área espacial das figuras geométricas. Para o reconhecimento, são usadas máquinas de vetores de suporte (SVM, do inglês *Support Vector Machines*). Essas SVM's são organizados em um agrupamento que leva em consideração uma classificação prévia dos sinais, ou seja, uma vez detectada a cor e a forma da placa, o sistema sabe exatamente qual o subgrupo de máquinas de vetores de suporte que devem ser utilizadas. Assim, não há necessidade de executar todas as SVM's disponíveis. A utilidade de um sistema como esse consiste na manutenção dos sinais de trânsito, no desenvolvimento de sistemas automáticos de controle de veículos ou até no apoio a deficientes visuais. Além da proposta do método, são apresentados os satisfatórios resultados obtidos, sendo feita uma comparação com os resultados já existentes na área. As taxas de acerto para a detecção são em média 83%, mas variam de acordo com as condições climáticas. Por outro lado, as taxas do reconhecimento são superiores a 87%. Consensualmente, outros desafios a serem superados referem-se à oclusão parcial dos sinais, à orientação e desgaste deles, bem como à falta de padronização das placas entre os países.

Palavras-chave: Detecção, Reconhecimento, Sinais de trânsito, SVM, YCbCr

Abstract

This work proposes a method to detect and recognize, automatically, traffic signs. The approach used for detection is based on segmentation in the YCbCr color space. This segmentation is performed through a thresholding process. Thus, the search for thresholds involves making histograms for the chrominance components Cb and Cr of the aforementioned space. The component Y stores information related to the brightness and is not used in this process, because the thresholds are very sensitive to light variations. Some criteria have been established trying to avoid false positives, that is, objects that are not of interest, but end up being detected as traffic signs. Examples of such criteria are the analysis of the overlaying traffic sign and the defined relationship between height and width of the signs to be identified. In addition, it proposes a simple way to identify the shape of the candidate signs by analyzing features related to the spatial area of geometric figures. For recognition, Support Vector Machines (SVM) are used. These SVM's are arranged in sets that takes into account a previous classification of the signals. Therefore, once detected the color and the shape of the traffic sign, the system knows exactly which subset of support vector machines that will be used, there is no need to execute all the available SVM's. The usefulness of such a system consists in the maintenance of traffic signs, in the development of automated systems for vehicle control or even in the supporting for visually impaired people. Besides the proposed method, the satisfactory results obtained are shown and a comparison was made with existing results in the area. For detection, the hit rates are on average 83%, but vary according to weather conditions. Moreover, the recognition rates are above 87%. Consensually, other challenges to be overcome are the partial occlusion of the signs, the orientation and the state of them as well as the lack of standardization of the traffic signs between countries.

Keywords: Detection, Recognition, Traffic Signs, SVM, YCbCr

Sumário

1	Introdução	1
1.1	Descrição do Problema	2
1.2	Objetivo geral	3
1.3	Objetivos específicos	3
1.4	Revisão da Bibliografia	3
1.5	Organização	6
2	Fundamentação Teórica	7
2.1	Processamento digital de imagens e Visão computacional	7
2.2	Imagem Digital	8
2.2.1	Amostragem e Quantização	9
2.3	Espaços de Cores	9
2.3.1	RGB	9
2.3.2	HSV	11
2.3.3	YCbCr	13
2.4	Histograma	14
2.5	Segmentação de imagens	15
2.5.1	Limiarização	15
2.6	Morfologia Matemática	16
2.6.1	Erosão	18
2.6.2	Dilatação	19
2.6.3	Abertura	21
2.6.4	Fechamento	21
2.7	Relacionamento entre os <i>pixels</i> de uma imagem	23
2.8	Aprendizagem de máquina e reconhecimento de padrões	24
2.8.1	Máquinas de Vetores de Suporte	25
3	Método proposto	28
3.1	Detecção	29
3.1.1	Espaço de Cores	29
3.1.2	Histogramas, limiarização e vizinhança	30
3.1.3	<i>Padding</i>	33
3.1.4	Operações Morfológicas	34
3.1.5	Tamanho das Placas	35
3.1.6	Relação altura/largura	36
3.1.7	Porcentagem de Branco	37

3.1.8	Sobreposição de placas	37
3.1.9	Detecção de Formas	38
3.2	Reconhecimento	40
3.2.1	Seleção da abordagem a ser utilizada	40
3.2.2	Treinamento das <i>SVM's</i>	41
3.2.3	Execução e Comparação de probabilidades	42
4	Resultados	44
4.1	Histogramas obtidos	44
4.2	Configurações dos testes	47
4.3	Saídas do sistema	47
4.4	Detecção	51
4.4.1	Resultados Obtidos	51
4.5	Reconhecimento	54
4.6	Comparação com outros trabalhos	58
5	Conclusões	60
	Referências	62

Lista de Figuras

1.1	Exemplo de dois sinais diferentes com o mesmo propósito: indicação de possível passagem de pedestres. (a) Placa do Brasil [1]. (b) Placa do Reino Unido [2].	2
1.2	Exemplos de características <i>haar-like</i> definidas por Viola e Jones. (a) e (b) Exemplos de características retangulares-2 (c) Característica retangular-3. (d) Características retangular-4. Imagem adaptada de [3]	5
2.1	Representação de uma imagem digital [4]. Essa convenção de coordenadas é amplamente utilizada, inclusive por Gonzalez e Woods [4] para discutir acerca de imagens digitais. Encontram-se representadas as noções de origem (do inglês <i>origin</i>) e de <i>pixel</i>	8
2.2	Imagem original e posterior amostragem e quantização. Imagem de [4]. . .	9
2.3	Espaço de cores RGB. As cores primárias, as quais não podem ser obtidas por nenhuma mistura de cores possível, são mostradas nas partes sem intersecção, isto é, vermelho, verde e azul. As intersecções representam as cores provenientes da adição de uma ou mais das cores primárias. Adaptado de [5].	10
2.4	Cubo de cores RGB. Pontos na diagonal principal, na qual os valores de R , G e B são iguais, representam os cinzas. A cor preta refere-se à origem e a branca está no ponto $(1, 1, 1)$. Assim, essas e as demais cores são pontos no próprio cubo ou dentro dele, sendo estipuladas por vetores que se estendem a partir da origem. Imagem de [4].	10
2.5	Cilindro que representa o espaço de cores HSV. O ângulo ao redor do eixo vertical central refere-se à tonalidade (H), o raio corresponde à saturação (S) e a altura desse cilindro significa o valor (V). Adaptado de [5].	11
2.6	Espaço de cores HSV representado como um hexa-cone. O topo do cone corresponde a $V = 1$, isto é, cores com intensidade máxima, ao passo que na sua base está a cor preta, $V = 0$. Cores complementares são opostas 180° , o que é medido por H , o ângulo em torno do eixo vertical V . S é uma taxa que varia de 0, quando se refere ao centro de V , até 1 nas laterais do hexa-cone. O ponto $S = 0, V = 1$ é a cor branca e é possível perceber que H é irrelevante quando $S = 0$. Imagem de [6].	12

2.7	Comparação entre os espaços de cores RGB e YCbCr. As cores mais claras, que apresentam valores maiores de $luma$, Y , são vistas no topo, no entanto, a cor branca não chega a tocar o teto do paralelepípedo externo. No fundo, encontra-se a cor preta e, tanto para ela, quanto para a branca, os valores de Cb e Cr são 128. É notável um gasto intrínseco ao espaço YCbCr, o qual ocorre pois os <i>displays</i> são RGB, logo valores inválidos para RGB são inutilizados. Assim, estão identificados os valores possíveis e o conjunto de valores válidos. Adaptada de [7].	13
2.8	Histograma - $h(r_k) \times r_k$. Cada k -ésimo nível de intensidade da imagem é representado por r_k , ao passo que $h(r_k)$ refere-se à quantidade de <i>pixels</i> dessa imagem que possuem tal intensidade. Adaptada de [4].	14
2.9	Limiarização que usa apenas um limiar. Na imagem de saída, os <i>pixels</i> pretos representam os <i>pixels</i> da imagem de entrada que possuem intensidade menor que o limiar, assim como os brancos indicam os valores que superam tal limiar. Imagem original de [4].	16
2.10	Operações básicas entre conjuntos de <i>pixels</i> , A e B [4]. Mais acima, nota-se a união e a intersecção entre os dois conjuntos. Mais abaixo, percebe-se o complemento de A e a diferença entre A e B	17
2.11	Exemplos de elementos estruturantes. O centro de cada um é identificado por um ponto preto [4].	17
2.12	Translação de R no ponto $z = (z_1, z_2)$ [4].	18
2.13	Reflexão do conjunto S [4].	18
2.14	Funcionamento da erosão [4]. (a) Região inicial. (b) Elemento estruturante quadrado. (c) Resultado da erosão. (d) Novo elemento estruturante. (e) Resultado da erosão usando o novo elemento estruturante definido em (d).	19
2.15	Exemplo de erosão. (a) Imagem Original 296×296 . (b) Imagem resultante da aplicação de um elemento estruturante de tamanho 12×12 . Nota-se que as porções da imagem menores que o elemento estruturante desapareceram.	19
2.16	Funcionamento da dilatação [4]. (a) Conjunto inicial R . (b) Elemento estruturante quadrado. (c) Dilatação de R pelo elemento quadrado. (d) Elemento estruturante alongado. (e) Dilatação com o elemento de (d).	20
2.17	Exemplo de dilatação [4]. (a) Uma amostra de um texto de baixa resolução, com letras incompletas. (b) Elemento estruturante. (c) Dilatação de (a) por (b). Nota-se um preenchimento dos elementos incompletos.	20
2.18	Funcionamento da abertura. Ela pode ser vista como o resultado de todos os deslocamentos do elemento estruturante B pela parte de dentro da fronteira de A . Imagem de [4].	21
2.19	Funcionamento do fechamento. Percebe-se o elemento estruturante B “rolando” pela fronteira externa de A . Imagem de [4].	22
2.20	Exemplo de aplicação da abertura e do fechamento [4]. (a) Imagem com ruído. (b) Elemento estruturante. (c) Resultado da erosão. (d) Abertura de A . (e) Resultado da dilatação posterior à abertura. (f) Fechamento da abertura.	22
2.21	$N4(p)$ - Vizinhança 4 de p . Nota-se que o <i>pixel</i> central claramente possui quatro vizinhos destacados em azul: 2 horizontais e 2 verticais. Adaptada de [4].	23

2.22	$ND(p)$ - Vizinhança diagonal de p . Nesse tipo de vizinhança, percebe-se que o <i>pixel</i> central também possui quatro vizinhos, mas, como sugere os nome, eles são diagonais. Adaptada de [4].	23
2.23	Vizinhança 8 de p , isto é, $N8(p) = ND(p) \cup N4(p)$. Adaptada de [4].	24
2.24	Exemplo de uma classificação feita por SVM. (a) Duas classes de pontos (círculos brancos e pretos) e três separadores lineares candidatos. (b) A linha mais escura indica o separador máximo de margem. Percebe-se que os vetores de suporte (pontos com círculos maiores ao redor) são as amostras mais perto do separador. Imagem de [8].	26
3.1	Diagrama de blocos que fornece uma noção geral da metodologia adotada. Merecem destaque os dois subprocessos maiores: detecção e reconhecimento (indicado como “Reconhecimento e classificação”).	28
3.2	Algumas das diferentes tonalidades utilizadas para a confecção dos histogramas e análise dos limiares. (a) Quatro tonalidades da cor azul. (b) Quatro tonalidades da cor vermelha. (c) Quatro tonalidades da cor amarela.	30
3.3	Histogramas de Cb e Cr obtidos a partir de uma das tonalidades de azul. O eixo das ordenadas apresenta a quantidade de <i>pixels</i> e o eixo das abscissas mostra o valor de cada componente. (a) Histograma gerado para a componente Cb . (b) Histograma para componente Cr	31
3.4	Exemplo da segmentação por cores adotada. (a) Imagem original, contendo uma placa de trânsito de interesse. (b) Imagem binarizada resultante da limiarização: Em branco, as porções da imagem que foram detectadas como vermelhas na imagem original; Em preto, tudo o que não é considerado como vermelho.	32
3.5	Exemplo da importância do <i>padding</i> , a margem extra acrescida na imagem de entrada do sistema. (a) Imagem original. (b) Imagem de saída com as devidas marcações características do processo de detecção.	33
3.6	Funcionamento e resultados das operações morfológicas feitas. (a) Imagem de entrada com a placa de trânsito. (b) Imagem resultante do processo de limiarização, sem a realização das operações morfológicas. (c) Imagem após a realização das operações da morfologia matemática, fechamento e abertura. É possível perceber o desaparecimento da ruptura presente na imagem anterior.	34
3.7	Diferentes tamanhos que as placas de trânsito podem assumir em uma imagem. Mais à esquerda e ao fundo (em amarelo), há uma placa praticamente imperceptível. Mais ao centro (em verde) tem-se uma placa de tamanho razoável que pode ser detectada e reconhecida pelo sistema. Mais à direita (em vermelho) existe uma placa não tão pequena como a primeira, no entanto ela também não tem tamanho suficiente quanto a segunda.	35
3.8	Sinal de trânsito que respeita a relação definida entre altura e largura, podendo ser detectado pelo sistema. Para ser mais específico, o valor numérico dessa razão é aproximadamente 0,97, dentro do intervalo previamente discutido.	36

3.9	Sinal de trânsito que não pode ser detectado pelo sistema. (a) Imagem original. (b) Imagem segmentada. Nota-se que a segmentação apresenta um ótimo resultado e que, embora esse sinal tenha mais de 40% da altura da imagem, sua razão altura/largura é de aproximadamente 0,61, estando fora do intervalo determinado pela Equação 3.4.	36
3.10	Imagem limiarizada que representa a detecção de uma placa losangular amarela. Os <i>pixels</i> brancos são aqueles considerados como amarelos em uma imagem de entrada. Os <i>pixels</i> pretos não são de interesse.	37
3.11	Representação das fronteiras das placas detectadas. (a) Fronteira externa detectada. (b) Fronteira interna detectada.	38
3.12	Preenchimento das formas identificadas. (a) Preenchimento da imagem mais à esquerda da Figura 3.11. (b) Preenchimento da imagem mais à direita da Figura 3.11.	38
3.13	Ilustração da análise complementar que o sistema faz para confirmar a suspeita de que um sinal de trânsito é de fato triangular. Em amarelo, as indicações aproximadas das linhas superiores e inferiores que são usadas para o teste de proporcionalidade.	39
3.14	Ilustração de como é feita a detecção da forma losangular. Em vermelho, as marcações que indicam as regiões analisadas. Verifica-se que o interior da placa candidata deve ser preenchido com a cor de interesse antes desse teste, pois os <i>pixels</i> pretos podem causar classificações equivocadas de forma.	40
3.15	Algumas das placas de treinamento das SVM's. (a) Sinais vermelhos triangulares (<i>Pedestrian</i>). (b) Placas vermelhas circulares (<i>Zone 40</i>). (c) Placas azuis circulares (<i>Mini Roundabout</i>). (d) Sinais amarelos losangulares de treinamento (<i>Pista Sinuosa</i>).	41
3.16	Classes de placas de trânsito que o sistema é treinado para reconhecer. Nota-se que a placa "STOP" é detectada como vermelha circular, o que não interfere no reconhecimento desse sinal de trânsito.	41
3.17	Organização dos SVM's. Nota-se o ganho computacional e a eficácia desse modelo no que diz respeito ao reconhecimento das placas de trânsito. Dessa forma, não é necessário executar todas as SVM's disponíveis, mas sim apenas aquelas condizentes com cada caso.	42
4.1	Uma das tonalidades usadas para se encontrar os limiares referentes à cor vermelha.	44
4.2	Histogramas YCbCr obtidos a partir da imagem da Figura 4.1. O eixo das ordenadas apresenta a quantidade de <i>pixels</i> e o eixo das abcissas mostra o valor de cada componente. (a) Histograma gerado para a componente <i>Cb</i> . (b) Histograma da componente <i>Cr</i>	45
4.3	Histogramas HSV obtidos a partir de 4.1. O eixo das ordenadas apresenta o número de <i>pixels</i> e o eixo das abcissas mostra o valor de cada componente. (a) Em azul, histograma gerado para a componente <i>H</i> . Em preto, histograma gerado para a componente <i>V</i> . (b) Histograma da componente <i>S</i>	45

4.4	Histogramas RGB obtidos a partir de 4.1. O eixo das ordenadas apresenta a quantidade de <i>pixels</i> e o eixo das abcissas mostra o valor de cada componente. (a) Histograma gerado para a componente <i>R</i> . (b) Histograma da componente <i>G</i> . (c) Histograma da componente <i>B</i>	46
4.5	Detecção e reconhecimento de uma placa azul circular em condições ensolaradas. Na parte de cima, a marcação que indica o local da placa. Na parte de baixo, a classificação do sistema para o sinal. A imagem original é do banco <i>Sunny</i> [9].	48
4.6	Detecção e reconhecimento de placas vermelhas com oclusão parcial. A marcações indicam o locais das placas e mais abaixo estão as classificações do sistema para os sinais. Apesar dos galhos na frente da placa, tanto o processo de detecção, quanto o de reconhecimento foram bem sucedidos. A imagem original é do banco <i>Sunny</i> [9].	48
4.7	Detecção e reconhecimento de uma placa vermelha triangular. Há a indicação da cor e da forma no local onde a placa foi detectada. Na parte de baixo está a classificação feita pelo sistema. A imagem original é do banco <i>Cloudy</i>	49
4.8	Detecção e reconhecimento de uma placa vermelha circular. Indicação da cor e da forma no local onde a placa foi detectada. A parte de baixo mostra a classificação feita pelo sistema. A imagem original é do banco <i>Rainy</i> . . .	49
4.9	Detecção de duas placas em uma mesma imagem: uma vermelha circular e uma amarela losangular. Nota-se que as marcações possuem as mesmas cor e forma das respectivas placas. A imagem original é do banco de placas brasileiras.	49
4.10	Detecção de duas placas losangulares amarelas em uma mesma imagem. As marcações possuem as mesmas cor e forma dos sinais detectados. A imagem original é do banco de placas brasileiras.	50
4.11	Detecção de uma placa amarela losangular. Imagem original é encontrada em [10]. (a) Percebe-se a quantidade de falsos positivos antes do análise da sobreposição das placas encontradas. (b) Marcação única para a placa identificada, após considerar o critério definido em 3.1.8.	50
4.12	Detecção da lanterna traseira do veículo como uma placa vermelha triangular. A imagem original é do banco <i>Rainy</i> [9].	51

Lista de Tabelas

4.1	Resultados da detecção em condições ensolaradas (<i>Sunny</i>). Na primeira coluna estão os resultados esperados e na primeira linha estão os resultados obtidos. O valor em cada célula da tabela refere-se a quantidade de placas de um determinado tipo. FP significa falsos positivos e FN falsos negativos. ‘Circ.’ significa Circular, ‘Triang.’ Triangular e ‘Ret.’ Retangular. A última coluna é a taxa de acertos de cada tipo.	52
4.2	Resultados da detecção em condições nubladas (<i>Cloudy</i>). A primeira coluna indica os resultados esperados, enquanto a primeira linha representa as classificações feitas pelo sistema. ‘Circular’, ‘Triangular’ e ‘Retangular’ são abreviadas como na Tabela 4.1. FP significa os falsos positivos e FN os falsos negativos. A última coluna refere-se à porcentagem de acertos do sistema.	52
4.3	Resultados da detecção em condições chuvosas, usando o banco <i>Rainy</i> . Na primeira coluna estão os resultados esperados e na primeira linha, encontram-se os resultados obtidos. ‘Circular’, ‘Triangular’ e ‘Retangular’ aparecem como ‘Circ.’, ‘Triang.’ e ‘Ret.’ respectivamente. FP diz respeito aos falsos positivos e FN aos falsos negativos. Na última coluna está a taxa de acerto do sistema.	53
4.4	Resultados da detecção usando o banco de imagens brasileiras. ‘Am.’ e ‘Verm.’ significam Amarelo e Vermelho respectivamente. ‘Los.’ representa a forma Losangular; ‘Ret.’ Retangular; ‘Tri’ triangular; e ‘Circ.’ Circular. As classes reais estão na primeira coluna e as classes preditas na primeira linha. FP são os falsos positivos e FN os falsos negativos. A última coluna refere-se aos acertos do sistema.	53
4.5	Resultado do processo de reconhecimento de placas brasileiras. Na primeira coluna estão as classes reais, isto é, os resultados esperados. A primeira linha representa as classes preditas, que são os resultados obtidos. FN são os falsos negativos e FP são os falsos positivos. Todas as placas são amarelas losangulares.	54
4.6	Resultado do reconhecimento em condições ensolaradas (<i>Sunny</i>). ‘Round.’ significa <i>Roundabout</i> . Verticalmente estão as classes reais e horizontalmente estão as classes preditas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.	55
4.7	Resultado do reconhecimento em condições nubladas (<i>Cloudy</i>). ‘Round.’ significa <i>Roundabout</i> . Verticalmente estão as classes reais e horizontalmente estão as classes preditas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.	56

4.8	Resultado do reconhecimento em condições chuvosas (<i>Rainy</i>). ‘Round.’ significa <i>Roundabout</i> . Primeira coluna: classes reais. Primeira linha: classes preditas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.	57
-----	---	----

Lista de Siglas, Abreviações e Acrônimos

ANN	<i>Artificial Neural Networks</i> , traduzido para o português como Redes Neurais Artificiais (RNA)
CIECAM97	CIE, do francês <i>Commission Internationale de l'Eclairage</i> , traduzido para o inglês como <i>International Commission on Illumination</i> e CAM de <i>Color Appearance Model</i> . O número 97 refere-se ao ano do modelo
CIELab	CIE, do francês <i>Commission Internationale de l'Eclairage</i> . Esse espaço também é conhecido como $L^*a^*b^*$, por isso Lab, <i>Lightness-color opponent dimensions color space</i>
CPU	<i>Central Processing Unit</i>
DENATRAN	Departamento Nacional de Trânsito
DT	<i>Distance Transform Image</i>
IHLS	<i>Improved Hue, Lumination and Saturation</i>
$L^*a^*b^*$	<i>Lightness-color opponent dimensions color space</i>
LDA	<i>Linear Discriminant Analysis</i>
LCH	<i>Lightness, Chroma and Hue</i>
HSI	<i>Hue, Saturation e Intensity</i> , respectivamente Matiz, Saturação e Intensidade
HSL	<i>Hue, Saturation e Lightness</i> , respectivamente Matiz, Saturação e Brilho
HSV	<i>Hue, Saturation e Value</i> , respectivamente Matiz, Saturação e Valor
MPEG	<i>Moving Picture Experts Group</i>
RBF	<i>Radial Basis Function s</i>
RGB	<i>Red Green Blue</i> , isto é, vermelho, verde e azul respectivamente
SVM	<i>Support Vector Machines</i> , em português, Máquinas de Vetores de Suporte
YCbCr	Espaço de Cores que separa a luminância, Y da Crominância Azul, Cb , e da Crominância Vermelha, Cr
YIQ	Espaço de cores com uma componente para a luminância, Y , e duas para a crominância, I e Q
YUV	Espaço de cores com uma componente para a luminância, Y , e duas para a crominância, U e V

Capítulo 1

Introdução

A sinalização de trânsito refere-se ao meio utilizado pela Engenharia de Trânsito para orientar a circulação de pedestres e condutores de veículos. Ela aparece de diferentes maneiras, mas este trabalho trata dos sinais de trânsito contidos na sinalização vertical, conforme classificação do DENATRAN (Departamento Nacional de Trânsito) [1].

Assim, esses sinais são entendidos aqui como placas de trânsito e são fundamentais, pois fornecem informações que ajudam os usuários das vias (pedestres, motoristas e ciclistas, por exemplo) a adotarem comportamentos adequados, no sentido de aumentar a segurança do trânsito e ordenar os fluxos do tráfego. Essas placas podem assumir as mais diferentes formas, aparecendo como triângulos, círculos, retângulos, losangos ou octógonos, por exemplo. Além disso, essas formas estão vinculadas ao tipo de mensagem que desejam transmitir, como indicações, regulamentações ou advertências. Dessa maneira, definem uma linguagem visual de fácil interpretação.

Apesar do uso desses sinais de trânsito já ser consolidado, não se trata de uma solução perfeita para que se dirija de uma maneira segura. Entre outros aspectos, pode-se considerar o fato de que os condutores são humanos e portanto falíveis, podendo não perceber a presença dessas placas, o que pode ocasionar graves acidentes, por exemplo, quando um motorista não sabe o sentido de uma via. Diante disso, sistemas de detecção e reconhecimento de sinais de trânsito têm ganhado cada vez mais importância, sendo grande a possibilidade de suas aplicações [11].

Assim, um sistema com essas características, entre outras finalidades, pode ser utilizado para:

- Desenvolver Sistemas de apoio ao motorista: Podem servir de suporte para limitar automaticamente a velocidade do veículo, enviar um sinal de alerta indicando excesso de velocidade ou indicar previamente a presença do sinal para o condutor.
- Apoiar os deficientes visuais: Sistemas de detecção e reconhecimento de placas de trânsito podem auxiliar a orientação e o deslocamento de tais pessoas, indicando onde há uma placa de faixa de pedestre, por exemplo.
- Manutenção dos sinais de trânsito: atualmente, é necessário que um humano assista gravações para verificar a presença e a condição dos sinais [11]. Isso é muito tedioso, pois as placas só aparecem de tempo em tempo, além da grande possibilidade de erros.

- Auxílio no deslocamento de carros autônomos: um sistema como o proposto por este trabalho poderia ser incorporado a carros sem motoristas, tal qual os desenvolvidos pela Google [12].

Dessa forma, o método desenvolvido neste trabalho tem como foco segmentar e identificar as placas de trânsito em uma imagem digital, de modo que possa ser utilizado em tarefas como as mencionadas anteriormente.

1.1 Descrição do Problema

Essencialmente os sinais de trânsito limitam a capacidade de locomoção dos deficientes visuais, os quais não têm acesso à sua informação. Além do mais, os motoristas não se concentram exclusivamente na condução do veículo, tendo que prestar atenção nas placas e, muitas vezes, sua interpretação pode ser equivocada, em um momento crítico que requer uma rápida tomada de decisão, por exemplo.

Vale destacar que em condições adversas de tempo, como a chuva, a neve ou a neblina, esses sinais chamam ainda menos a atenção dos condutores. De maneira complementar, durante a noite, a visibilidade dos motoristas pode ser ofuscada pelos faróis dos demais veículos.

Outros fatores que podem comprometer a interpretação de um sinal de trânsito são a oclusão parcial, como galhos e folhas na frente das placas; o desgaste desses sinais, seja ele natural ou não; ou mesmo as diversas angulações que os sinais podem assumir.

É importante também tratar de um problema maior, que se refere à falta de padronização dos sinais entre os diferentes países. Desse modo, com uma rápida análise entre manuais de sinalização de trânsito, como um do Brasil [1] e outro do Reino Unido [2], é possível notar que duas placas completamente diferentes entre os países podem ter o mesmo significado, como ilustra a Figura 1.1. Além do mais, algumas placas apresentam texto, dificultando bastante a condução de motoristas estrangeiros.



Figura 1.1: Exemplo de dois sinais diferentes com o mesmo propósito: indicação de possível passagem de pedestres. (a) Placa do Brasil [1]. (b) Placa do Reino Unido [2].

Portanto, idealmente, deve-se desenvolver um sistema capaz de detectar e reconhecer os sinais de trânsito de um modo eficaz, independente da posição, da orientação, da escala e do estado dos sinais. Deve-se considerar também seu desempenho nas diversas condições climáticas, em situações que houver oclusão parcial, ou mesmo na presença de outros objetos que apresentem a mesma cor desses sinais.

1.2 Objetivo geral

Desenvolver um método para detecção e reconhecimento de placas de trânsito em imagens digitais.

1.3 Objetivos específicos

- Selecionar as ferramentas adequadas para atingir de modo satisfatório o proposto no objetivo geral;
- Adquirir bancos de imagens que contenham sinais de trânsito brasileiros e britânicos considerados essenciais, de modo a comparar o desempenho do método com relação a países diferentes. Os sinais britânicos foram escolhidos, pois existe um banco de imagens de livre acesso na *internet*, utilizado por [9];
- Estabelecer quais tipos dessas imagens adquiridas serão consideradas, determinando limites de escala para as placas de trânsito presentes nas imagens obtidas;
- Definir de forma precisa qual será a saída do processo de detecção, a qual servirá como entrada para o processo de reconhecimento desses sinais;
- Analisar e comparar os resultados obtidos com aqueles já obtidos na área.

1.4 Revisão da Bibliografia

Os crescentes aumentos no poder da computação têm colocado a visão computacional na grade das aplicações comerciais [13]. À medida que os computadores ganham poder de processamento, a possibilidade de se ter a detecção e o reconhecimento de placas de trânsito em tempo real é cada vez mais plausível. Nesse sentido, é importante tratar do estado da arte referente à este trabalho, merecendo destaque dois artigos que fazem isso muito bem.

Brkić [13] esboça um panorama geral dos métodos mais populares presentes na literatura, no que diz respeito à detecção e ao reconhecimento dos sinais de trânsito. Diante disso, afirma que esses dois processos muitas vezes estão interligados, citando inclusive que o método desenvolvido por Ruta et al. [14] utiliza a segmentação por cor como um dos passos iniciais para o reconhecimento. Assim, primeiro segmentam a imagem normalmente, mas logo depois realçam as cores obtidas para auxiliar o processo subsequente.

A autora ainda constatou que o espaço RGB (*Red Green Blue*) é considerado como muito sensível a iluminação, fazendo com que muitos trabalhos escolham espaços como o HSI (*Hue Saturation Intensity*), o HSV (*Hue Saturation Value*) e o $L^*a^*b^*$ (*lightness-color opponent dimensions*, que possui uma componente para a luminância, L , e duas para a crominância, a e b) para a detecção dos sinais. Quanto ao processo de reconhecimento, geralmente são usadas técnicas de aprendizagem de máquina, como as Máquinas de Vetores de Suporte (SVM, do inglês *Support Vector Machines*), as redes neurais artificiais (RNA ou ANN, do inglês *Artificial Neural Networks*), ou ainda a LDA (*Linear Discriminant Analysis*).

Fleyeh e Dougherty [15], por sua vez, explicam o primeiro artigo apareceu em 1984 no Japão, com o objetivo de tentar a utilização de vários métodos da visão computacional para detectar objetos em cenas ao ar livre. Esses autores afirmam que, desde então, muitos grupos de pesquisa se interessam por esse campo e a quantidade de trabalhos feitos têm sido enorme, com avanços progressivos.

Esses dois autores também afirmam que o uso dos espaços de cores HSI e HSV é muito comum nessas tarefas, mas alguns outros espaços, como YUV (espaço de cores com uma componente para a luminância, Y , e duas para a crominância, U e V) e YIQ (espaço de cores com uma componente para a luminância, Y , e duas para a crominância, I e Q) também são utilizados. Por fim, destacam as redes neurais artificiais como a alternativa mais escolhida para o reconhecimento dos sinais.

Uma vez estabelecida essa visão geral, pode-se tratar dos trabalhos efetivamente desenvolvidos com o intuito de detectar e/ou reconhecer os sinais de trânsito. Inicialmente, vale citar o trabalho de Piccioli et al. [16]. Esses autores utilizam o modelo de cor HSV e, em linhas gerais, sua detecção considera sinais triangulares e circulares. Assim, perceberam que o primeiro tipo de sinal apresenta sempre bordas vermelhas, sendo que a segunda classe abrange sinais vermelhos e azuis.

Para a detecção da forma das placas, foram analisados os contornos das bordas dos sinais, buscando a obtenção de contornos circulares ou triangulares, os quais são candidatos a serem sinais de trânsito. A informação inicial para os algoritmos de detecção da forma dessas placas foi representada em um mapa de bordas, extraído a partir da região de interesse por meio do algoritmo de Canny [17]. Quanto ao reconhecimento, foi analisado um índice de correlação entre a imagem de entrada e uma imagem padrão (*template*).

Outro trabalho que merece destaque é o de Gao et al. [9], o qual usa um banco de imagens do Reino Unido que discrimina sinais obtidos em condições ensolaradas (*Sunny*), nubladas (*Cloudy*) e chuvosas (*Rainy*). Para a segmentação, utilizam o espaço de cor LCH (*Lightness, Chroma and Hue*) definido por CIECAM97 (CIE, do francês *Commission Internationale de l'Eclairage* [18], traduzido para o inglês como *International Commission on Illumination* e CAM de *Color Appearance Model*) [19]. Dessa maneira, os autores conseguiram limiares relativos às três componentes do espaço LCH para cada condição climática. Os limiares superiores são tidos como as médias dos valores encontrados somadas aos desvios-padrão, ao passo que os limiares inferiores são as médias menos os desvios-padrão. Porém, somente as componentes *Hue* e *Chroma* foram consideradas, sob a alegação de que a luminosidade varia bastante de acordo com as condições climáticas.

Para o reconhecimento, Gao et al. utilizam uma técnica denominada FOSTS, baseada nas características geométricas obtidas pela comparação entre o centro da placa e sua borda. Mais detalhes dessa abordagem podem ser obtidos em [20] e [21]. Vale destacar que esses autores reconhecem a forma do sinal de trânsito, mas não há definição de qual é a placa realmente.

Vega et al. [22] propõem um método de detecção baseado na segmentação no espaço IHLS (*Improved Hue, Lumination and Saturation*). Feito isso, identifica-se a forma do sinal, a partir de representações de formas que utilizam curvas Gielis [23]. Os autores afirmam que, além de não precisar de um treinamento inicial, a vantagem disso é que não há a necessidade de uma arquitetura com várias camadas, como as usadas em redes neurais ou em máquinas de vetores de suporte. Assim, defendem que o método é rápido, simples e independente da escala dos sinais de trânsito.

Lopez e Fuentes [24] utilizam o espaço de cores CIELab [18], também conhecido como $L^*a^*b^*$, já citado anteriormente. Para a segmentação, mesclam três modelos Gaussianos, com o intuito de caracterizar cada cor dos sinais. Descrevem o algoritmo em três etapas básicas: inicialização, geração dos modelos e detecção do alvo. A cada passo, os resultados vão sendo filtrados, utilizando-se uma função de aprendizagem que atualiza o valor médio das distribuições. Também afirmam que o sistema proposto é sensível à cor, mas invariante a mudanças de intensidade baseadas apenas nas componentes relacionadas a luminância.

Varan et al. [25] acreditam terem desenvolvido um sistema bastante confiável utilizando o próprio espaço RGB. Desse modo, adotaram uma fórmula simples para a detecção de sinais vermelhos, expressa por $1,5 \times R > G + B$. Por outro lado, Gavrilin [26] usa uma abordagem baseada no casamento de modelos (do inglês *template matching*) para detectar a forma dos sinais. O primeiro passo é encontrar as bordas e os cantos da imagem original para então construir uma imagem denotada como DT (*Distance Transform Image*). Em uma imagem DT, cada *pixel* significa a distância com relação à borda ou ao canto mais próximo. Assim, computa-se também a imagem DT do modelo utilizado, a qual é comparada com a DT obtida a partir da imagem original. Para o reconhecimento, o autor utiliza uma rede de funções de base radial (RBF, do inglês *Radial Basis Function*).

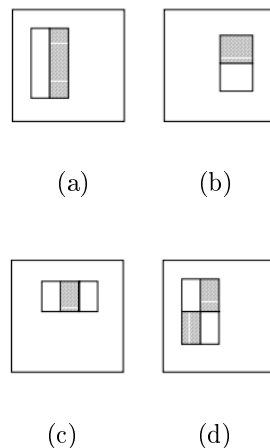


Figura 1.2: Exemplos de características *haar-like* definidas por Viola e Jones. (a) e (b) Exemplos de características retangulares-2 (c) Característica retangular-3. (d) Características retangular-4. Imagem adaptada de [3]

Viola e Jones [3] desenvolveram um algoritmo capaz de detectar objetos em tempo real e de maneira confiável, o qual Brkić [13] destaca como um “marco na visão computacional”.

O treinamento desse detector usa um conjunto de exemplos positivos e negativos. Originalmente destinado à detecção de rosto, esse método têm sido utilizado por vários outros pesquisadores para as mais diversas classes de objetos, entre eles, os sinais de trânsito. Um exemplo disso é o trabalho desenvolvido por Brkić et al. [27].

Sumariamente, tal detector pode ser entendido como uma cascata que combina classificadores mais complexos de maneira sucessiva, o que fornece um melhor desempenho, já que processamentos mais elaborados são reservados somente a regiões promissoras, ou seja, aquelas que têm mais possibilidade de apresentar objetos de interesse. Essa cascata combina dois conceitos: classificadores *Haar-like* e *AdaBoost* [28].

Classificadores *Haar-like* são construídos a partir de características retangulares simples, as quais representam diferenças de somas de *pixels* específicos em uma imagem. Nesse sentido, Viola e Jones [3] propõem uma nova representação para imagens, denominada imagem integral (do inglês, *integral image*). Esse método permite a computação das características supracitadas de um modo muito rápido em várias escalas, sendo que a imagem integral pode ser obtida de uma imagem original usando poucas operações por *pixel*. Uma vez calculadas, essas características *Haar-like* podem ser computadas em qualquer escala ou localização em tempo constante. Viola e Jones estabelecem 3 tipos dessas características, como ilustra a Figura 1.2.

Em contrapartida, *AdaBoost* é uma técnica usada para combinar certo número de fracos classificadores em um forte [13]. Porém, os autores perceberam que a quantidade total de características *Haar-like* era muito ampla, muito maior que o número de *pixels*. Baseados em [29], os autores então fizeram uma pequena adaptação no procedimento *AdaBoost*, a qual estipula certas restrições para os classificadores, de modo que todo classificador fraco retornado possa depender apenas de uma única característica. Desse modo, os autores afirmam que cada etapa do processo escolhe um novo classificador fraco, podendo ser vista como um mecanismo de seleção de características.

Por fim, vale destacar o método proposto por Maldonado et al. para detecção das placas de trânsito da Espanha. Os autores citam como dificuldades as diferentes condições luminosas e a possível deterioração dos sinais. Contudo, afirmam que as componentes H (*hue*, matiz) e S (*saturation*, saturação) são suficientes para isolar as placas de trânsito em uma cena, utilizando limiares fixos, os quais foram obtidos a partir da elaboração de diversos histogramas.

Além disso, foram utilizados dois tipos de SVM's. Para a detecção da forma, foram utilizados SVM's lineares, valendo destacar que os sinais são rotacionados antes de serem submetidos a esse processo, a fim de obter um método invariante à orientação do sinal. Uma vez que a placa tem sua forma identificada, o processo de reconhecimento se inicia. Ele foi implementado por SVM's com *kernels* Gaussianos, pois os autores afirmam que, muitas vezes, as SVM's lineares não são suficientes para separar os dados adequadamente.

Percebe-se que Maldonado et al. [30] buscaram desenvolver um sistema bastante completo, capaz de detectar todas as formas de sinais de trânsito possíveis na Espanha. Assim, a organização e o objetivo do artigo supracitado serviram de base para o desenvolvimento do presente trabalho.

1.5 Organização

Este trabalho está dividido em 5 capítulos, estando o presente capítulo incluído nessa quantidade. Os conceitos básicos relacionados ao processamento de imagens, que são utilizados para o desenvolvimento da metodologia adotada estão presentes no Capítulo 2. Depois disso, no Capítulo 3, explica-se o método proposto detalhadamente e exhibe-se um diagrama de blocos dos passos que foram seguidos. O Capítulo 4 mostra as saídas do processo de detecção e reconhecimento dos sinais de trânsito, bem como as matrizes de confusão referentes aos resultados obtidos e a configuração dos testes utilizados. As conclusões, as contribuições deste trabalho e as considerações finais, como sugestões de trabalhos futuros, são mostradas no quinto e último capítulo.

Capítulo 2

Fundamentação Teórica

Este capítulo refere-se aos conceitos necessários para o entendimento do método desenvolvido neste trabalho. Assim, inicialmente é importante tratar de algumas diferenças entre o Processamento digital de imagens e a Visão computacional. É preciso explicar também o conceito de imagem digital, que pode ser representada em diversos espaços de cores, também conhecidos como modelos de cores. Desse modo, cada espaço pode influenciar a maneira como se detectam os sinais de trânsito em uma imagem, sendo conveniente descrever os modelos de cores mais recorrentes nos trabalhos prévios. Outro conceito que deve ser explicado é o de histograma, o qual pode ser entendido como uma representação gráfica que relaciona o número de *pixels* aos níveis de cor presentes em uma imagem.

Esses histogramas variam de acordo com o espaço de cores escolhido e podem ser úteis para processos de segmentação de imagens, como a limiarização. Tudo isso será mais bem explicado adiante, mas vale ressaltar que a análise estatística de limiares correspondentes a um objeto de interesse pode contribuir para detectar esse mesmo objeto em outras imagens digitais. Nesse sentido, convém descrever algumas operações concernentes à morfologia matemática que podem ser aplicadas às imagens digitais, com o intuito de melhorar o resultado da detecção. É pertinente abordar ainda a noção de relacionamento entre os *pixels* de uma imagem, pois ela também é utilizada no processo de segmentação.

Por fim, uma vez expostos os conceitos referentes à detecção das placas de trânsito, é importante discorrer acerca da aprendizagem de máquina e do reconhecimento de padrões, para assim tratar do processo de reconhecimento desses sinais. Dessa maneira, é necessário considerar as máquinas de vetores de suporte, as quais precisam de um processo de treinamento antes de serem realmente utilizadas.

2.1 Processamento digital de imagens e Visão computacional

Inicialmente, um esclarecimento é necessário. Embora não haja uma fronteira bem definida entre o processamento digital de imagens e a visão computacional, o primeiro termo refere-se basicamente a uma subdisciplina do processamento digital de sinais, na qual a entrada e a saída do processamento são imagens digitais. Por outro lado, o segundo termo pode ser entendido como um ramo da Inteligência Artificial, tendo como objetivo principal a emulação da visão humana, o que inclui aprendizado e inferência [4].

Entre esses dois extremos, ainda podemos citar a chamada Análise de imagens, que, nas palavras de Gonzalez e Woods [4] envolve “dar sentido” a um conjunto de objetos reconhecidos. No processo de detecção de sinais de trânsito são usados conceitos e ferramentas que permeiam todas essas áreas, tais como técnicas de realce ou mesmo reconhecimento de padrões, por exemplo.

2.2 Imagem Digital

Uma imagem digital pode ser representada por uma função bidimensional $f(., .)$, onde $f(x, y)$ é o valor ou o brilho na posição (x, y) . Logo, imagens digitais podem ser vistas como matrizes de duas dimensões que contém M linhas e N colunas. Gonzalez e Woods [4] estabelecem números inteiros para representar essas coordenadas discretas, sendo $x = 0, 1, 2, \dots, M - 1$ e $y = 0, 1, 2, \dots, N - 1$.

Nesse sentido, pode-se dizer que o valor da imagem digital na origem é $f(0, 0)$ e a notação convencionada que $f(0, 1)$ representa a segunda amostra da primeira linha dessa matriz, ou seja, x cresce de cima para baixo e y cresce da esquerda para direita na Figura 2.1.

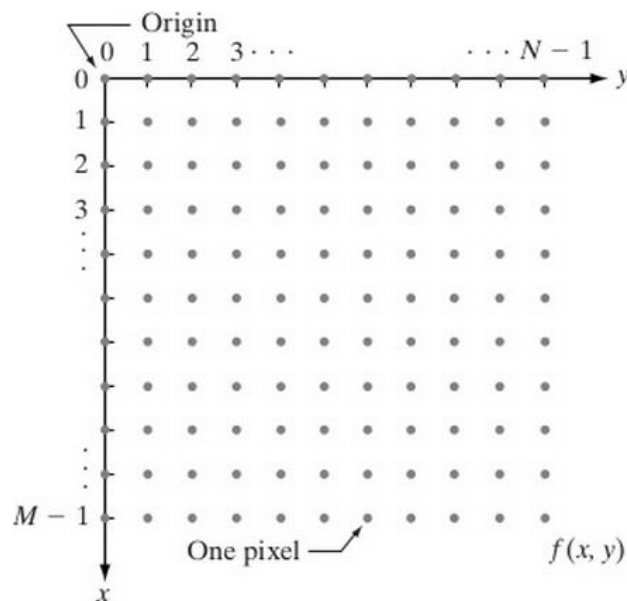


Figura 2.1: Representação de uma imagem digital [4]. Essa convenção de coordenadas é amplamente utilizada, inclusive por Gonzalez e Woods [4] para discorrer acerca de imagens digitais. Encontram-se representadas as noções de origem (do inglês *origin*) e de *pixel*.

Cada elemento da matriz bidimensional correspondente à imagem é chamado de *pixel*, elemento de imagem ou elemento pictórico.

É importante ressaltar que imagens digitais são representações discretas de uma função $f(s, t)$ contínua de duas variáveis também contínuas, s e t . Para converter tal função em uma imagem digital é necessária a realização de dois processos, a amostragem e a quantização, os quais serão explicados logo a seguir.

2.2.1 Amostragem e Quantização

Muito utilizados no contexto de processamento de sinais, são conceitos fundamentais a amostragem e a quantização. No que tange à aquisição de uma imagem, esses dois processos são extremamente úteis, uma vez que há a necessidade de gerar imagens digitais a partir dos dados contínuos captados pelos sensores de imageamento.

São esses sinais digitais que são adequadamente tratados pelo computador, como é ilustrado mais à direita da figura 2.2.

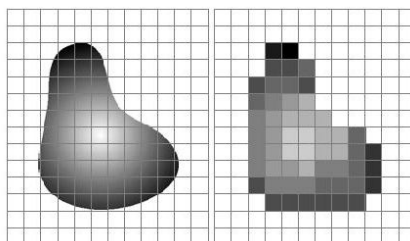


Figura 2.2: Imagem original e posterior amostragem e quantização. Imagem de [4].

A amostragem implica a discretização das coordenadas x e y da imagem, ao passo que a quantização é relacionada com o conceito de amplitude, isto é, torna finita a quantidade de níveis de cinza possíveis.

2.3 Espaços de Cores

Nas palavras de Ford e Roberts [31], um modelo de cor “é uma forma de especificar, visualizar e criar cores”. Mais especificamente, pode-se dizer que um espaço de cores pode ser entendido como um modelo matemático de formalização da descrição de cores por meio de agrupamentos, ou tuplas, de números, geralmente de três ou quatro elementos. Uma cor pode ser representada pela combinação de tais números, cabendo destacar que esses espaços de cores são, via de regra, tridimensionais.

Constatou-se que os espaços de cores mais frequentemente utilizados para a detecção de sinais de trânsito são o HSV e o YCbCr. No entanto, faz-se necessária uma introdução inicial ao espaço de cores talvez mais comum quando se trata de imagens digitais, o RGB.

2.3.1 RGB

RGB é a abreviação dos termos em inglês *Red*, *Green* e *Blue*. Sendo assim, esse espaço representa o sistema de cores aditivas formado basicamente pelas cores primárias Vermelho, Verde e Azul, que são as traduções respectivas das palavras descritas anteriormente. Essas cores primárias podem ser vistas na Figura 2.3 e não podem ser geradas por nenhuma mistura de cores. Por outro lado, elas podem ser misturadas, de maneira que uma adição envolvendo duas ou mais delas resulte em outras cores que não sejam as primárias. Embora essa figura apresente uma boa ideia do espaço RGB, deve-se estabelecer que ele de fato se baseia em um sistema de coordenadas cartesianas, sendo o subespaço de interesse o cubo, como explicam Gonzalez e Woods [4] com a Figura 2.4.

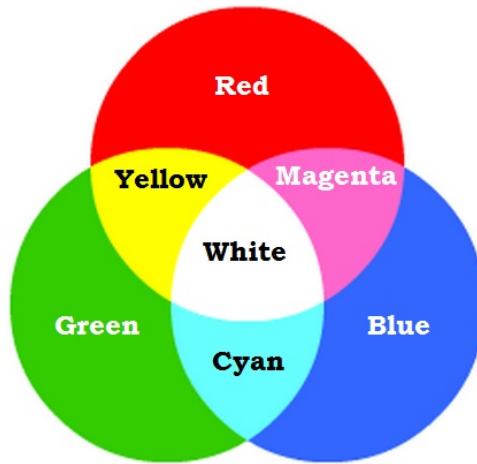


Figura 2.3: Espaço de cores RGB. As cores primárias, as quais não podem ser obtidas por nenhuma mistura de cores possível, são mostradas nas partes sem intersecção, isto é, vermelho, verde e azul. As intersecções representam as cores provenientes da adição de uma ou mais das cores primárias. Adaptado de [5].

Diferentemente do modelo subtrativo CMYK (Ciano, Magenta, Amarelo e Preto, do inglês *Cyan*, *Magenta*, *Yellow* e *Key*), utilizado pelas impressoras, o RGB é mais adequado para exibição de imagens por meio de dispositivos com capacidade de emissão de luz, como as imagens de computadores, e só foi viável devido ao desenvolvimento tecnológico de tubos de raios catódicos, os quais tornaram possível o *display* de cores ao invés de uma fosforescência. A representação mais comum para o RGB utiliza até 24 *bits* de informação para cada *pixel*, sendo que cada componente (R , G ou B) corresponde a 8 *bits* e pode assumir 256 valores diferentes [32].

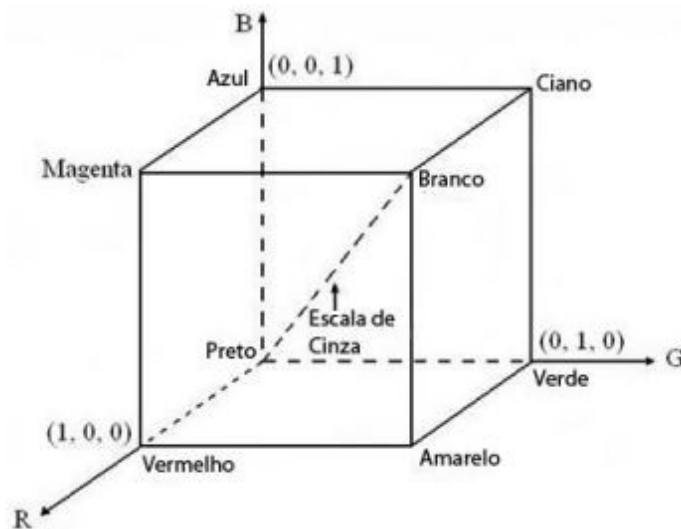


Figura 2.4: Cubo de cores RGB. Pontos na diagonal principal, na qual os valores de R , G e B são iguais, representam os cinzas. A cor preta refere-se à origem e a branca está no ponto $(1, 1, 1)$. Assim, essas e as demais cores são pontos no próprio cubo ou dentro dele, sendo estipuladas por vetores que se estendem a partir da origem. Imagem de [4].

Contudo, o espaço de cores RGB é bastante sensível a iluminação [13]. Esse é um aspecto que merece atenção, pois o olho humano é mais sensível ao brilho que à cor, por conta da diferente quantidade de cada tipo de receptor de luz ao longo da retina. Desse modo, existem por volta de 6 a 7 milhões de receptores denominados cones, que são muito sensíveis à cor. O outro tipo refere-se aos bastonetes, os quais não estão relacionados com a visualização das cores e são sensíveis a baixos níveis de iluminação. Estão presentes aproximadamente 75 a 150 milhões desses receptores no olho humano [4].

Diante dessa premissa, muitos espaços de cores foram desenvolvidos para tentar resolver esse problema, mas os modelos a seguir podem ser todos derivados a partir das informações RGB fornecidas por dispositivos como câmeras e *scanners* [6].

2.3.2 HSV

Na metade dos anos 1970, pesquisadores da área de computação gráfica desenvolveram o HSL (*Hue Saturation Lightness*) e o HSV (*Hue Saturation Value*), os quais rearranjam o espaço de cores RGB em coordenadas cilíndricas, o que resulta em uma representação mais próxima da percepção visual dos humanos, de acordo com Brkić [13]. Cabe destacar que o modelo HSV também pode ser definido como HSB (*Hue Saturation Brightness*) e ainda que existe um modelo bem similar ao HSL denominado HSI (*Hue Saturation Intensity*), constantemente utilizado em visão computacional, preferível quando se trata de processamento tradicional de imagens, como a equalização e a convolução [6].

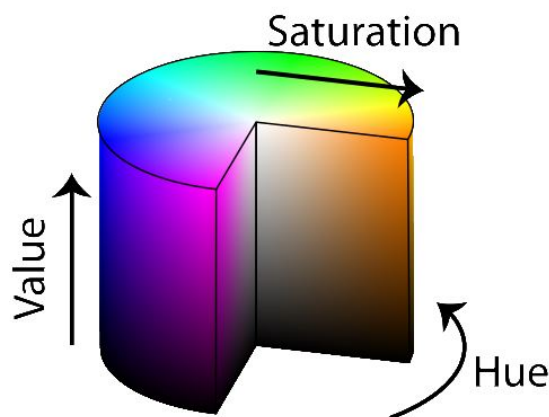


Figura 2.5: Cilindro que representa o espaço de cores HSV. O ângulo ao redor do eixo vertical central refere-se à tonalidade (H), o raio corresponde à saturação (S) e a altura desse cilindro significa o valor (V). Adaptado de [5].

Os modelos HSL, HSV e HSI diferem apenas na definição da sua terceira componente: L representa luz, luminância, V é de valor e I de intensidade. As duas primeiras componentes são Hue (matiz, tonalidade), que pode variar de 0° a 360° , e $Saturation$ (saturação), a qual varia de 0 até 1. No cilindro HS^* , o ângulo ao longo do eixo vertical central corresponde à tonalidade, o raio à saturação e a altura representa o brilho, o valor ou a intensidade, que também estão entre os valores 0 e 1. Essa ideia geral pode ser esboçada pela Figura 2.5. Pode-se ainda representar o HSV por meio de um cone, como o que foi feito por Jack [6], ilustrado na Figura 2.6.

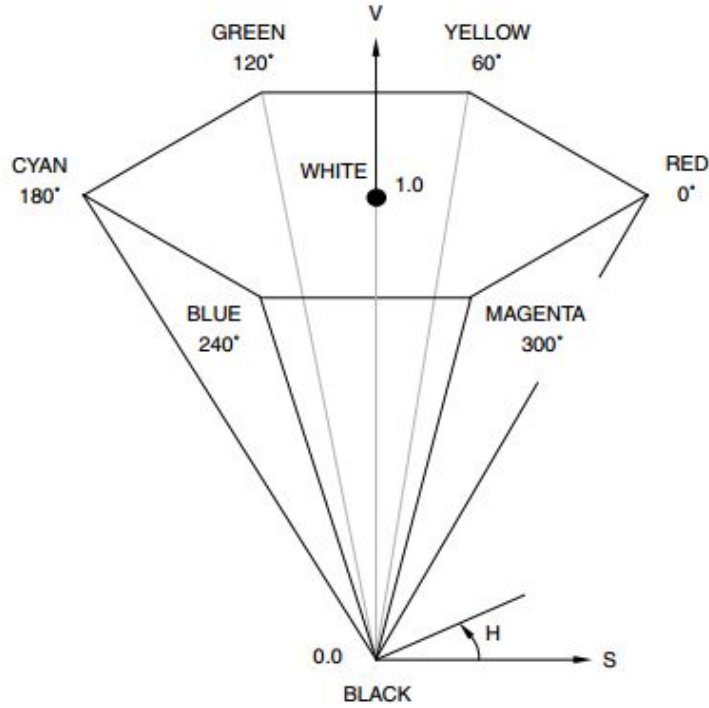


Figura 2.6: Espaço de cores HSV representado como um hexa-cone. O topo do cone corresponde a $V = 1$, isto é, cores com intensidade máxima, ao passo que na sua base está a cor preta, $V = 0$. Cores complementares são opostas 180° , o que é medido por H , o ângulo em torno do eixo vertical V . S é uma taxa que varia de 0, quando se refere ao centro de V , até 1 nas laterais do hexa-cone. O ponto $S = 0, V = 1$ é a cor branca e é possível perceber que H é irrelevante quando $S = 0$. Imagem de [6].

Nessas representações, a componente matiz permite uma menor sensibilidade quanto à variação da iluminação. Porém, para baixos valores de brilho, experimentalmente a matiz (H) é mais sensível às variações de valores das componentes no espaço RGB, como afirma Brys [33]. Além disso, há outra característica indesejável desses espaços, porque o custo computacional para o tratamento do “brilho”, seja a luminância, o valor ou a intensidade, é elevado [34].

Por fim, cabe destacar os passos envolvidos na conversão do espaço RGB para o HSV. Antes de mais nada, é necessário dividir as componentes R, G e B por 255, obtendo-se as componentes normalizadas R', G' e B' respectivamente. Definindo-se o valor MAX como sendo o máximo entre essas novas componentes, MIN como o menor valor entre elas e $\Delta = MAX - MIN$, a relação entre esses dois espaços é dada pelas equações a seguir:

$$Hue = \begin{cases} 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6\right) & \text{se } MAX = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2\right) & \text{se } MAX = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4\right) & \text{se } MAX = B' \end{cases} \quad (2.1)$$

$$Saturation = \frac{(MAX - MIN)}{MAX} \quad (2.2)$$

$$Value = MAX \tag{2.3}$$

onde $mod6$ indica o resto da divisão de um valor por 6.

Se $Hue < 0$, então $Hue = Hue + 360^\circ$.

2.3.3 YCbCr

Outro espaço de cores muito útil é o YCbCr, o qual, assim como o HSV, também possui uma componente que armazena separadamente o brilho da imagem, Y , *luma*. A grande diferença é que a transformação de RGB para YCbCr é muito simples, conforme a Equação 2.4. Isso faz sentido, pois os humanos são mais sensíveis ao brilho que à cor, como já foi dito antes. Esse aspecto pode ser conveniente na compressão de imagens, por exemplo, haja vista que esse modelo é de fato utilizado na compressão MPEG (*Moving Picture Experts Group*), sendo amplamente usado em sistemas de vídeo. As duas outras componentes são Cb , que representa a cromaticidade azul, isto é, o azul subtraído do brilho ($blue - luma$), e Cr , a qual significa a cromaticidade vermelha, ou seja, o vermelho subtraído do brilho ($red - luma$).

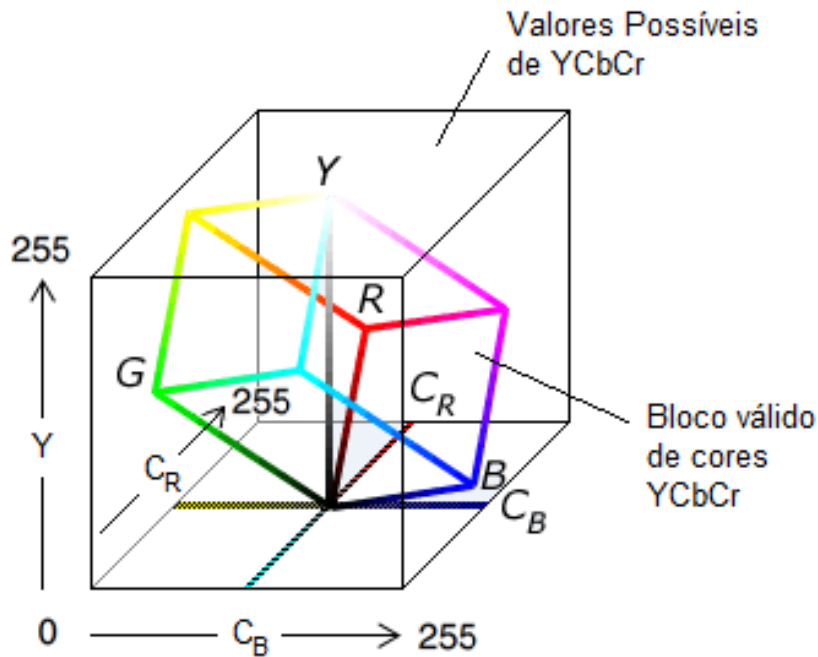


Figura 2.7: Comparação entre os espaços de cores RGB e YCbCr. As cores mais claras, que apresentam valores maiores de *luma*, Y , são vistas no topo, no entanto, a cor branca não chega a tocar o teto do paralelepípedo externo. No fundo, encontra-se a cor preta e, tanto para ela, quanto para a branca, os valores de Cb e Cr são 128. É notável um gasto intrínseco ao espaço YCbCr, o qual ocorre pois os *displays* são RGB, logo valores inválidos para RGB são inutilizados. Assim, estão identificados os valores possíveis e o conjunto de valores válidos. Adaptada de [7].

Detalhando melhor, a componente Y é definida para abranger um alcance nominal 8 *bits* de 16 até 235, à medida que Cb e Cr possuem um alcance nominal de 16 a 240 [6]. Nota-se que, apesar dos 256 valores possíveis para as componentes teoricamente, o desperdício é de 75% no espaço YCbCr, Isso ocorre porque todos os *displays* são RGB, então quaisquer valores não válidos de R , G e B não podem ser utilizados, como ilustra a Figura 2.7.

O espaço YCbCr foi desenvolvido como uma parte do ITU-R BT.601 [35], quando se estabeleceu um padrão mundial para componentes digitais de vídeos. Vale destacar, porém, que o YCbCr não é um espaço de cores absoluto, mas sim um modo diferente de codificar a informação do RGB.

A conversão do espaço RGB para o YCbCr utiliza a seguinte transformação [32]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65,481 & 128,553 & 24,966 \\ -37,797 & -74,203 & 112,000 \\ 112,000 & -93,786 & -18,214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.4)$$

2.4 Histograma

Dada uma imagem digital com L níveis de intensidade no intervalo $[0, L - 1]$, pode-se definir um *histograma* como uma função discreta $h(r_k) = n_k$, sendo r_k o k -ésimo valor de intensidade e n_k o número de *pixels* da imagem com intensidade r_k , como exemplificado pela Figura 2.8.

Geralmente esses histogramas são normalizados, dividindo-se cada componente pelo número total de *pixels* da imagem, expresso pelo produto MN , no qual M representa o número de linhas e N o número de colunas da imagem.

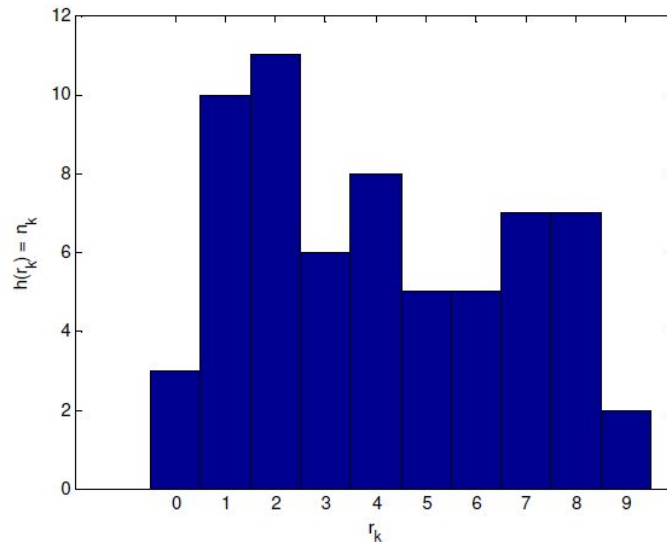


Figura 2.8: Histograma - $h(r_k) \times r_k$. Cada k -ésimo nível de intensidade da imagem é representado por r_k , ao passo que $h(r_k)$ refere-se à quantidade de *pixels* dessa imagem que possuem tal intensidade. Adaptada de [4].

Assim, um histograma normalizado pode ser descrito por $p(r_k) = r_k/MN$, onde $k = 0, 1, 2, \dots, L - 1$. Desse modo, a soma de todos os componentes de um histograma normalizado é igual a 1, uma vez que $p(r_k)$ representa uma estimativa da probabilidade de ocorrência do nível r_k na imagem.

Desse modo, percebe-se que os histogramas fornecem estatísticas importantes da imagem, além de serem facilmente calculados pelo computador, pois podem ser implementados de maneira econômica em termos de *hardware*. Assim os histogramas possuem grandes aplicações no processamento de imagens, sendo útil em técnicas de domínio espacial, compressão ou mesmo podendo ser utilizados como base para processos de segmentação, como a limiarização. Esses processos são explicados logo adiante.

2.5 Segmentação de imagens

De acordo com Gonzalez e Woods [4], a segmentação é o processo de subdividir uma imagem em regiões ou objetos que a compõem. Essa subdivisão pode ser feita considerando níveis de detalhes diferentes, dependendo da aplicação e do propósito. Um exemplo é o caso da inspeção automatizada de componentes eletrônicos, no qual se analisam imagens desses produtos para, entre outros, identificar a falta de componentes e circuitos de conexão interrompidos. Assim, a segmentação deve ser executada de modo que não exceda o nível necessário de detalhamento para realizar esse procedimento.

Diante disso, a precisão da segmentação é fundamental, pois depende dela o fracasso ou o sucesso das tarefas de análise computadorizada. Entretanto, a segmentação de imagens não convencionais é uma das missões mais desafiadoras no processamento de imagens.

Gonzalez e Woods [4] ainda tratam de dois grandes grupos de processos de segmentação. O primeiro se refere à análise de descontinuidades presentes nas imagens, como as bordas, dividindo uma imagem com base nas mudanças bruscas de intensidade. O segundo diz respeito à divisão de uma imagem em regiões que apresentam certas similaridades embasadas em uma série de critérios predefinidos, como é o caso da limiarização, do crescimento de regiões, da divisão e fusão de regiões.

Dito isso, o que realmente interessa aqui é a limiarização, que foi de fato utilizada neste trabalho. Esse conceito estabelece o referido critério predefinido como sendo a escolha de um ou mais limiares, visando à segmentação da imagem de interesse.

2.5.1 Limiarização

Limiarização é um processo utilizado na segmentação de imagens baseado na diferença dos níveis de intensidade que compõe diferentes objetos de uma imagem.

A partir de um limiar T estabelecido, de acordo com as características dos objetos que se deseja isolar, a imagem pode ser segmentada em dois grupos: o grupo de *pixels* com níveis de intensidade abaixo do limiar e o grupo de *pixels* com níveis de intensidade acima do limiar. Em uma imagem limiarizada, atribui-se um valor fixo para todos os *pixels* de mesmo grupo.

Logo, uma imagem $g(x, y)$ limiarizada pode ser definida como a equação:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (2.5)$$

Essa imagem $g(x,y)$ é uma imagem binária, isto é, os seus *pixels* assumem o valor 0 ou 1, representando o preto e o branco, respectivamente, como ilustra a Figura 2.9. No entanto, dependendo da aplicação específica, pode ser necessário preservar alguns valores originais da imagem quando se faz a limiarização, não necessariamente sendo apenas o preto e o branco as cores almejadas. Desse modo, a função não seria especificada como na equação 2.5, mas poderia ser da forma:

$$h(x,y) = \begin{cases} 1 & \text{se } f(x,y) > T \\ f(x,y) & \text{se } f(x,y) \leq T \end{cases} \quad (2.6)$$

Os níveis de intensidade da imagem original que estiverem acima do limiar definido podem ser saturados desse modo, sem que se perca o valor dos demais *pixels*. Além disso, é possível definir mais de um limiar, isto é, realizar uma limiarização de vários níveis.

Assim, vale ressaltar que a limiarização tem importância fundamental nos processos de segmentação de imagens, sendo utilizada inclusive em boa parte dos trabalhos feitos na área de detecção de sinais de trânsito. Isso se deve pela simplicidade de implementação e pelo bom desempenho computacional intrínsecos à limiarização.

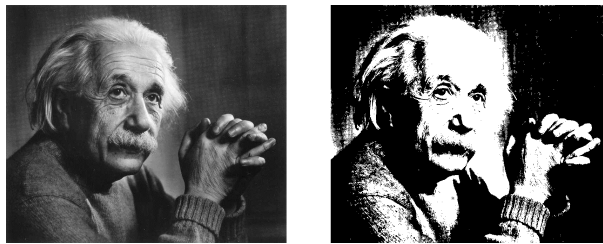


Figura 2.9: Limiarização que usa apenas um limiar. Na imagem de saída, os *pixels* pretos representam os *pixels* da imagem de entrada que possuem intensidade menor que o limiar, assim como os brancos indicam os valores que superam tal limiar. Imagem original de [4].

2.6 Morfologia Matemática

Basicamente, operações morfológicas em imagens são muito úteis no sentido de pré processar imagens para uma aplicação específica. Pode ser entendida como uma ferramenta para extrair componentes das imagens que auxiliam na descrição da forma de uma região; obtêm-se assim, atributos da imagem, extraindo significados dela, como resultado do processamento.

Gonzalez e Woods [4] explicam que a morfologia utiliza como linguagem a teoria dos conjuntos e, portanto, as imagens em níveis de cinza podem ser representadas como conjuntos cujos elementos estão no Z^3 , sendo que os dois primeiros componentes são as coordenadas dos *pixels* e o terceiro corresponde ao seu valor discreto de intensidade. Nesse contexto, é importante exemplificar algumas operações básicas entre conjuntos de *pixels* ou regiões, dado que são recorrentes no processamento morfológico. A Figura 2.10 mostra exemplos de união, intersecção e diferença entre dois conjuntos, A e B , além de apresentar o conjunto complementar de A .

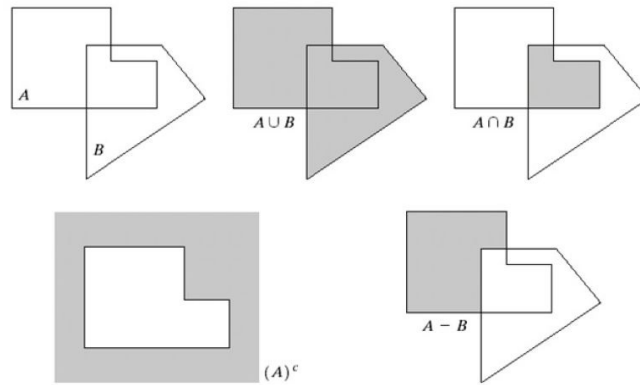


Figura 2.10: Operações básicas entre conjuntos de *pixels*, A e B [4]. Mais acima, nota-se a união e a intersecção entre os dois conjuntos. Mais abaixo, percebe-se o complemento de A e a diferença entre A e B .

Dessa maneira, um conceito central aqui é o de elementos estruturante, os quais são pequenos conjuntos ou subimagens usadas para fazer a varredura de uma imagem, buscando propriedades de interesse. A Figura 2.11 mostra alguns exemplos desses elementos.

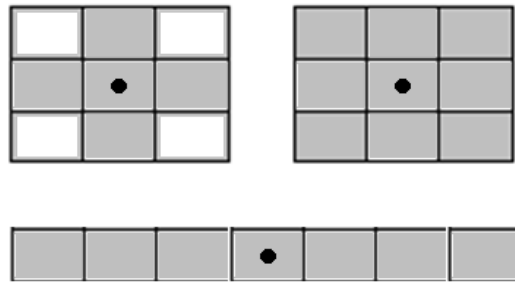


Figura 2.11: Exemplos de elementos estruturantes. O centro de cada um é identificado por um ponto preto [4].

Cabe ainda definir a translação e a reflexão de conjuntos. A translação de um conjunto R no ponto $z = (z_1, z_2)$ é:

$$(R)_z = \{c | c = r + z, r \in R\} \quad (2.7)$$

Sendo $(R)_z$ o conjunto de pontos em R cujas coordenadas (x, y) foram trocadas por $(x + z_1, y + z_2)$.

Por outro lado, a reflexão de um conjunto S é definida por:

$$\hat{S} = \{w | w = -s, s \in S\} \quad (2.8)$$

ou seja, é simplesmente o conjunto dos pontos em S cujas coordenadas (x, y) foram substituídas por $(-x, -y)$.

Dito isso, a reflexão e a translação são exhaustivamente empregadas na morfologia, a fim de formular as operações baseadas nos elementos estruturantes.

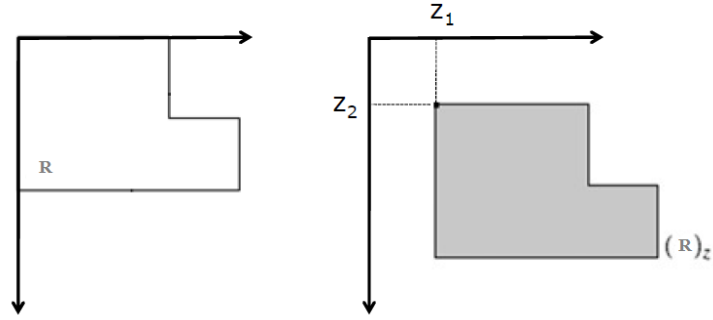


Figura 2.12: Translação de R no ponto $z = (z_1, z_2)$ [4].

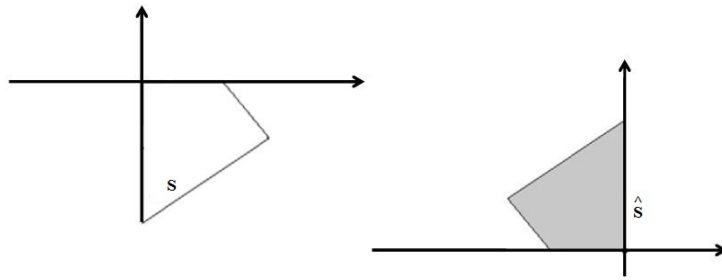


Figura 2.13: Reflexão do conjunto S [4].

2.6.1 Erosão

Sendo S o elemento estruturante, pode-se definir a erosão por:

$$R \ominus S = \{z | (S)_z \subseteq R\} \quad (2.9)$$

Em outras palavras, isso significa que a erosão de R por S é o conjunto formado por todos os pontos z , tal que S transladado de z está contido em R . Pode ser usada para remover componentes indesejáveis de uma imagem e seu funcionamento pode ser demonstrado na Figura 2.14.

Descrevendo melhor essa figura, tem-se uma região inicial R , ilustrada em (a), um elemento estruturante S em (b) e em (c) consta a erosão de R por S , mostrada sombreada. Em (d) define-se um novo elemento estruturante alongado S e em (e) mostra-se a erosão de R por S . A borda pontilhada em (c) e (e) estão presentes apenas como referência e representam a fronteira da região inicial R .

De acordo com Salomon e Breckon [36] a erosão tem o efeito de remover pequenas características isoladas, como pode ser visto na Figura 2.15. Nela, os objetos menores que o elemento estruturante definidos sumiram, o que pode ser útil para localizar elementos de um tamanho específico em uma imagem, por exemplo.

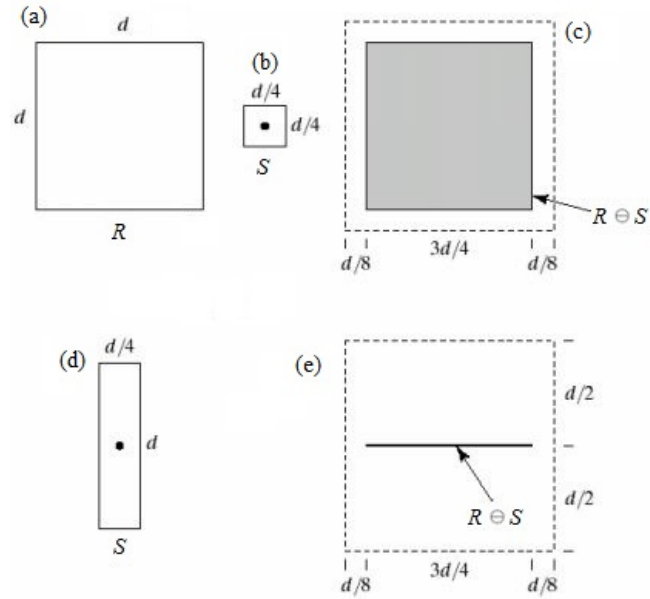


Figura 2.14: Funcionamento da erosão [4]. (a) Região inicial. (b) Elemento estruturante quadrado. (c) Resultado da erosão. (d) Novo elemento estruturante. (e) Resultado da erosão usando o novo elemento estruturante definido em (d).

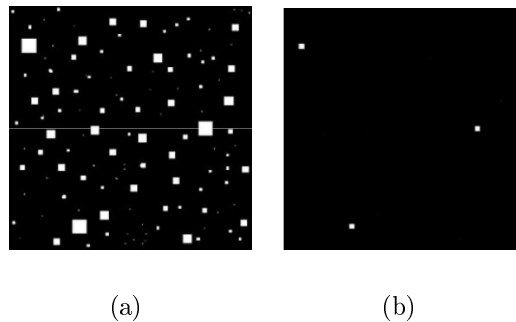


Figura 2.15: Exemplo de erosão. (a) Imagem Original 296×296 . (b) Imagem resultante da aplicação de um elemento estruturante de tamanho 12×12 . Nota-se que as porções da imagem menores que o elemento estruturante desapareceram.

2.6.2 Dilatação

Mais uma vez, usando S como o elemento estruturante, a dilatação pode ser expressa como:

$$R \oplus S = \{z | (\hat{S})_z \cap R \neq \emptyset\} \quad (2.10)$$

Ou seja, a dilatação de R por S é o conjunto de todos os deslocamentos z de forma que S refletido e R se sobrepõem em pelo menos um elemento. Segundo Gonzalez[4], essa operação, ao contrário da erosão, serve para “engrossar” os objetos em uma imagem binária, sendo que a forma e a extensão desse espessamento podem ser controladas pelo formato do elemento estruturante usado.

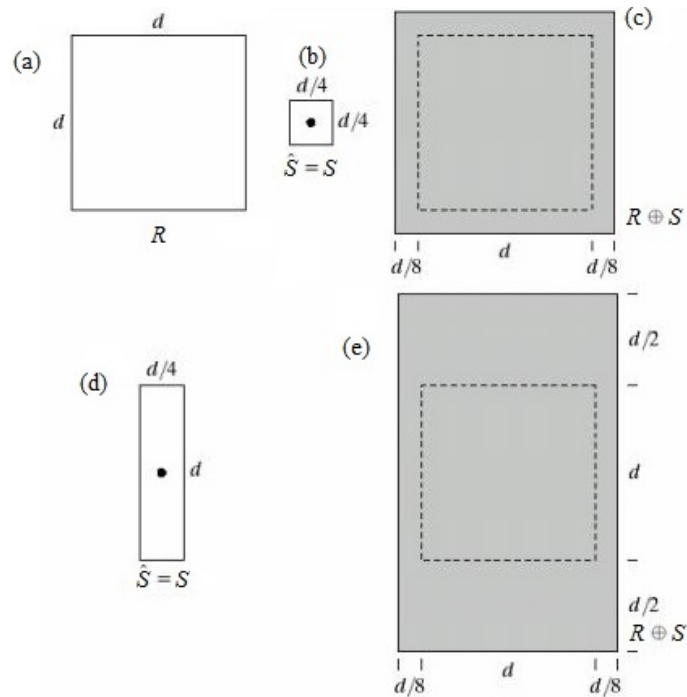


Figura 2.16: Funcionamento da dilatação [4]. (a) Conjunto inicial R . (b) Elemento estruturante quadrado. (c) Dilatação de R pelo elemento quadrado. (d) Elemento estruturante alongado. (e) Dilatação com o elemento de (d).

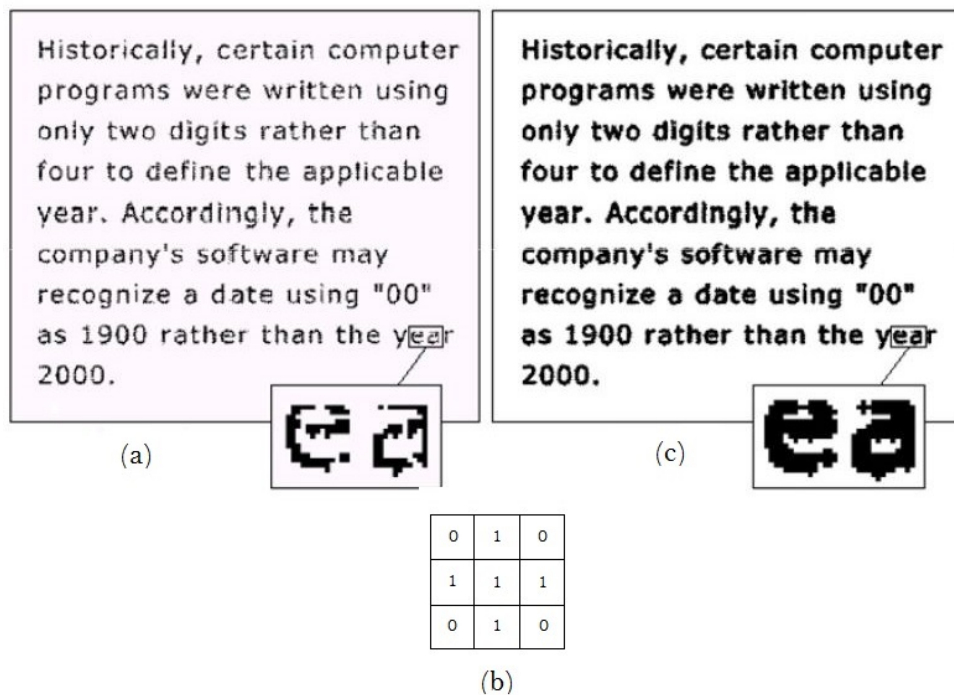


Figura 2.17: Exemplo de dilatação [4]. (a) Uma amostra de um texto de baixa resolução, com letras incompletas. (b) Elemento estruturante. (c) Dilatação de (a) por (b). Nota-se um preenchimento dos elementos incompletos.

Explicando a Figura 2.16, tem-se uma região inicial ou um conjunto R , ilustrado em (a), um elemento estruturante S em (b) e em (c) é mostrada a dilatação de R por S , mostrada sombreada. Em (d) define-se um novo elemento estruturante alongado \hat{S} e em (e) está presente a dilatação de R por \hat{S} . A borda pontilhada em (c) e (e) estão presentes apenas para efeito de comparação e representam a fronteira da região inicial R . Um exemplo real do resultado desse processamento é dado pela Figura 2.17.

É relevante destacar a dualidade inerente às operações de dilatação e erosão em relação à complementação e à reflexão de conjuntos. Essa característica é importante no caso do elemento estruturante ser simétrico em relação à sua origem, como ocorre frequentemente, de modo que $\hat{S} = S$. Assim, tem-se a dualidade expressa em:

$$(R \ominus S)^c = R^c \oplus \hat{S} \quad (2.11)$$

e

$$(R \oplus S)^c = R^c \ominus \hat{S} \quad (2.12)$$

Dessa forma, a erosão de uma imagem por S pode ser obtida fazendo simplesmente a dilatação do fundo, isto é, dilatando R^c .

2.6.3 Abertura

Geralmente, a abertura suaviza contornos de objetos, rompe istmos, bem como elimina saliências finas. É definida por uma erosão seguida de uma dilatação:

$$R \circ S = (R \ominus S) \oplus S \quad (2.13)$$

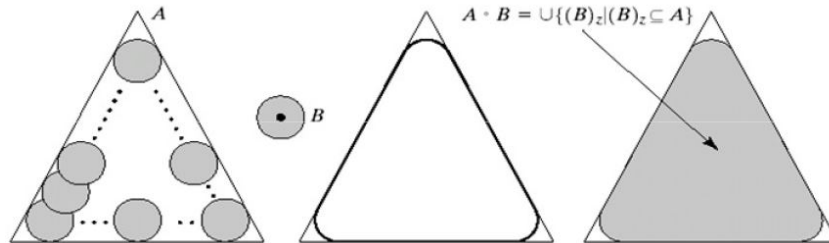


Figura 2.18: Funcionamento da abertura. Ela pode ser vista como o resultado de todos os deslocamentos do elemento estruturante B pela parte de dentro da fronteira de A . Imagem de [4].

2.6.4 Fechamento

Também tende a suavizar contornos, no entanto, diferentemente da abertura, no geral funde as descontinuidades estreitas e alonga os golfos finos, além de eliminar pequenos buracos e preencher lacunas em um contorno. É uma dilatação seguida de uma erosão, sendo expresso como:

$$R \bullet S = (R \oplus S) \ominus S \quad (2.14)$$

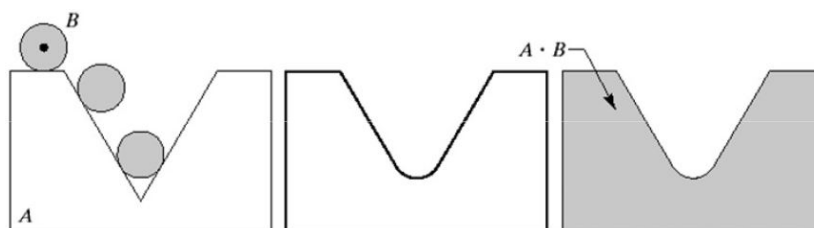


Figura 2.19: Funcionamento do fechamento. Percebe-se o elemento estruturante B “rolando” pela fronteira externa de A . Imagem de [4].

Vale destacar que, dependendo da aplicação específica, muitas vezes a abertura e o fechamento podem ser combinados, como no caso da obtenção de impressões digitais, ilustrada na Figura 2.20. Antes do reconhecimento das impressões, essas imagens são pré-processadas com o intuito de remover os ruídos de aquisição, mas sem perder as minúcias, que são informações essenciais para distinguir unicamente as digitais de cada pessoa.

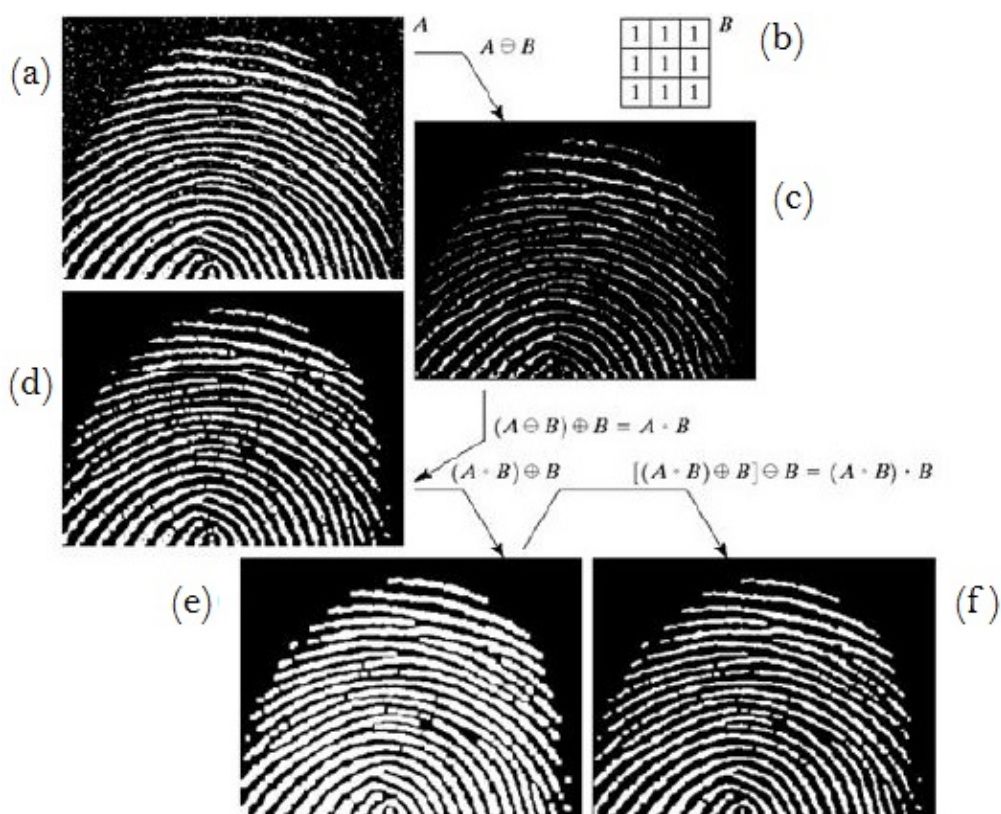


Figura 2.20: Exemplo de aplicação da abertura e do fechamento [4]. (a) Imagem com ruído. (b) Elemento estruturante. (c) Resultado da erosão. (d) Abertura de A . (e) Resultado da dilatação posterior à abertura. (f) Fechamento da abertura.

Como no caso da dilatação e da erosão, a abertura e o fechamento também são duais entre si, no que se trata da complementação e da reflexão de conjuntos, ou seja:

$$(R \bullet S)^c = R^c \circ \hat{S} \quad (2.15)$$

e

$$(R \circ S)^c = R^c \bullet \hat{S} \quad (2.16)$$

2.7 Relacionamento entre os *pixels* de uma imagem

Nas linhas que se seguem, serão definidos os termos vizinhança, adjacência, conectividade, região e contorno. Cada pixel p , localizado em (x, y) , possui quatro vizinhos no sentido horizontal e vertical, identificados pelos pares ordenados $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ e $(x, y - 1)$.

$x-1, y-1$	$x-1, y$	$x-1, y+1$
$x, y-1$	x, y	$x, y+1$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

Figura 2.21: $N4(p)$ - Vizinhança 4 de p . Nota-se que o *pixel* central claramente possui quatro vizinhos destacados em azul: 2 horizontais e 2 verticais. Adaptada de [4].

Esses quatro vizinhos, chamados de 4-vizinhos, como mostra a Figura 2.21, são denotados por $N4(p)$. Já os vizinhos de p localizados nas diagonais são dados por $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ e denotados por $ND(p)$, como demonstrado na Figura 2.22.

$x-1, y-1$	$x-1, y$	$x-1, y+1$
$x, y-1$	x, y	$x, y+1$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

Figura 2.22: $ND(p)$ - Vizinhança diagonal de p . Nesse tipo de vizinhança, percebe-se que o *pixel* central também possui quatro vizinhos, mas, como sugere os nome, eles são diagonais. Adaptada de [4].

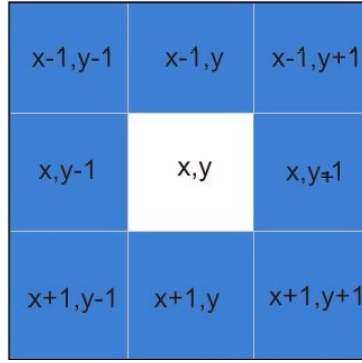


Figura 2.23: Vizinhança 8 de p , isto é, $N8(p) = ND(p) \cup N4(p)$. Adaptada de [4].

Diante disso, os pontos de $ND(p)$ unidos aos pontos de $N4(p)$ são chamados de 8-vizinhos de p , como na Figura 2.23 e são denotados por $N8(p)$. Está, então, definida a vizinhança entre *pixels*.

Seja considerado agora um conjunto V que contenha níveis de cinza tidos como similares. Pode-se definir três tipos de adjacência:

- 4-adjacência: dois *pixels* p e q são considerados 4-adjacentes se ambos possuem seus níveis de cinza em V e se q é 4-vizinho de p , ou seja, pertence ao conjunto $N4(p)$ definido anteriormente.
- 8-adjacência: dois *pixels* p e q são considerados 8-adjacentes se ambos possuem seus níveis de cinza em V e se q é 8-vizinho de p , ou seja, pertence ao conjunto $N8(p)$.
- m-adjacência (também chamada adjacência mista): dois *pixels* p e q são considerados m-adjacentes se ambos possuem seus níveis de cinza em V e se:

$\Rightarrow q$ está em $N4(p)$, ou

$\Rightarrow q$ está em $ND(p)$ e não há interseção entre $N4(p)$ e $N4(q)$.

Dois *pixels* pertencentes a um subconjunto S são considerados conectados caso, seguindo um dos critérios de adjacência, exista um caminho entre eles também pertencente S . Para um pixel p , o conjunto de *pixels* que o une a q é chamado de conjunto conectado. Caso q seja adjacente a p , q é chamado de componente conectado. Uma região R é definida como sendo um conjunto conectado. Por fim, um contorno C de uma região R é definido como os *pixels* que têm vizinhos não pertencentes a R [4].

2.8 Aprendizagem de máquina e reconhecimento de padrões

O objetivo da aprendizagem de máquina (do inglês *machine learning*) é programar computadores para aprender um determinado comportamento ou padrão automaticamente a partir de exemplos ou observações, ou seja, por meio de um processo inicial de treinamento [37].

Atualmente, muitas aplicações utilizam algoritmos de aprendizagem de máquina, incluindo sistemas para prever o comportamento de clientes a partir de dados de compras, reconhecer faces ou voz, extrair conhecimento de dados biológicos, ou mesmo reconhecer sinais de trânsito. Desse modo, pode-se citar alguns tópicos centrais tais como classificação indutiva, árvores de decisão, aprendizado de regras, redes neurais e máquinas de vetores de suporte.

Em especial, este trabalho utiliza as máquinas de vetores de suporte (SVM's) para o reconhecimento de padrões, com o intuito de classificar as placas de trânsito previamente detectadas.

2.8.1 Máquinas de Vetores de Suporte

Máquinas de Vetores de Suporte consistem em uma técnica de análise de dados e reconhecimento de padrões. Assim, trata-se de um método usado para classificação e análise de regressão, sendo embasado pela teoria de aprendizado estatístico, desenvolvida por Vapnik [38].

Essa técnica tem ganhado cada vez mais destaque no ramo da aprendizagem de máquina e, de acordo com Batista et al. [39], ela apresenta resultados comparáveis e muitas vezes superiores aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (*RNA's*). Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como na categorização de textos, na análise de imagens e na Bioinformática.

De acordo com Russell e Norvig [8], três propriedades tornam as SVM's atrativas:

- SVM's constroem um separador máximo de margem, que se refere a uma fronteira de decisão a qual abrange a maior distância possível com relação aos pontos de uma amostra. Com isso, proporcionam uma boa generalização.
- SVM's criam um hiperplano de separação linear, mas abrangem os dados em um espaço dimensional maior, utilizando o chamado *kernel trick*, que será explicado mais adiante. Assim, os dados que não podem ser separados de maneira linear no espaço original de entrada são separados com facilidade nesse espaço maior.
- SVM é um método não paramétrico, ou seja, conserva as amostras de treinamento e possivelmente armazena todas elas. Entretanto, na prática ele conserva apenas uma pequena parte do total das amostras. Desse modo, SVM's possuem as vantagens, tanto dos modelos não paramétricos, quanto das abordagens paramétricas, pois demonstram flexibilidade para representar funções complexas, mas são resistentes ao sobreajuste (do inglês *overfitting*).

Na Figura 2.24 (a), percebe-se que as três linhas serviriam para classificar as amostras em duas classes corretamente. No entanto, não seriam linhas que uma SVM estabeleceria, pois, analisando a linha mais abaixo, nota-se que os pontos pretos estão muito perto da fronteira. Assim, vale destacar que alguns exemplos de entrada são mais importantes que outros para as SVM's. Na Figura 2.24 (b) é possível verificar o funcionamento de uma SVM, sendo a linha mais escura definida com base nos vetores de suporte, que são os pontos com círculos maiores ao redor. Essa linha é denominada separador máximo de margem, já citado anteriormente.

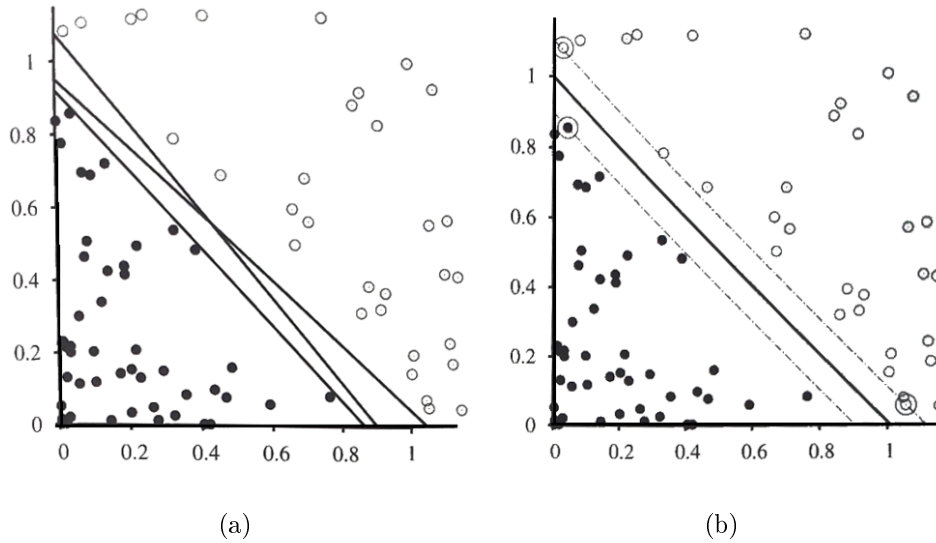


Figura 2.24: Exemplo de uma classificação feita por SVM. (a) Duas classes de pontos (círculos brancos e pretos) e três separadores lineares candidatos. (b) A linha mais escura indica o separador máximo de margem. Percebe-se que os vetores de suporte (pontos com círculos maiores ao redor) são as amostras mais perto do separador. Imagem de [8].

Dessa maneira, as SVM's tentam minimizar a perda de generalização, ao invés de minimizar a perda empírica com relação aos dados de treinamento.

Para descrever a matemática por trás das SVM's, Russell e Norvig [8] estabelecem algumas notações. Dessa maneira, definem os rótulos para as classes de interesse como $+1$ e -1 e estabelecem um vetor de pesos \mathbf{w} que contém um valor referente à intersecção (diferentemente do que as SVM's fazem realmente, guardando a intersecção em um parâmetro separado, b). Com isso, estipulam que o separador é o conjunto de pontos $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 0\}$, onde \cdot é o produto ponto, também conhecido como produto escalar.

Diante disso, os autores apresentam uma representação denominada representação dual, a qual se trata de um problema de otimização de programação quadrática. Assim, para encontrar o separador ótimo, é preciso saber os valores dos parâmetros α_j que maximizam a expressão a seguir:

$$\sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j \cdot \mathbf{x}_k) \quad (2.17)$$

sendo que $\alpha_j \geq 0$ e $\sum_j \alpha_j y_j = 0$. Encontrando-se o vetor α , é possível continuar na representação dual ou recuperar o vetor \mathbf{w} por meio da fórmula $\mathbf{w} = \sum_j \alpha_j \mathbf{x}_j$.

Quanto à Equação 2.17, é importante ressaltar que a expressão é convexa, portanto tem um único máximo global, que pode ser encontrado com eficiência. Além disso, os dados entram nessa expressão apenas sob a forma de produtos escalares dos pares de pontos. Por fim, vale destacar que os pesos α_j associados com cada ponto dos dados valem 0, exceto para os vetores de suporte, que são os pontos mais próximos do separador, sendo chamados assim pelo fato de sustarem o plano que divide as classes. A quantidade desses vetores de suporte é geralmente muito menor que o número de amostras.

Por outro lado, percebe-se que nem sempre os dados de entrada de uma SVM podem ser separados linearmente. No entanto, isso pode não ser possível no espaço de entrada, mas se os dados forem mapeados para um espaço de dimensões suficientes, eles quase sempre poderão ser separados de modo linear. Esse mapeamento é feito por uma função específica $K(\mathbf{x}_j, \mathbf{x}_k)$, denominada função de *kernel*, que pode ser aplicada aos pares dos dados de entrada para avaliar os produtos escalares em algum espaço de características de interesse. Desse modo, basta substituir $\mathbf{x}_j \cdot \mathbf{x}_k$ por $K(\mathbf{x}_j, \mathbf{x}_k)$ na Equação 2.17.

Assim, pode-se falar em espaços dimensionais maiores, no entanto somente funções de *kernel* são de fato computadas, ao invés de toda a lista de característica de cada ponto da entrada. Esse é justamente o *kernel trick* a que Russell e Norvig [8] se referem, pois a alocação de tais funções na Equação 2.17 permite que separadores lineares ótimos possam ser encontrados de maneira eficiente em espaços de características com bilhões (ou, em alguns casos, um número indefinido) de dimensões.

No Capítulo 3, será abordada a metodologia adotada por este trabalho para detectar e reconhecer, automaticamente, as placas de trânsito em imagens digitais.

Capítulo 3

Método proposto

Este capítulo detalha o método proposto para detectar e reconhecer os sinais de trânsito de maneira automática.

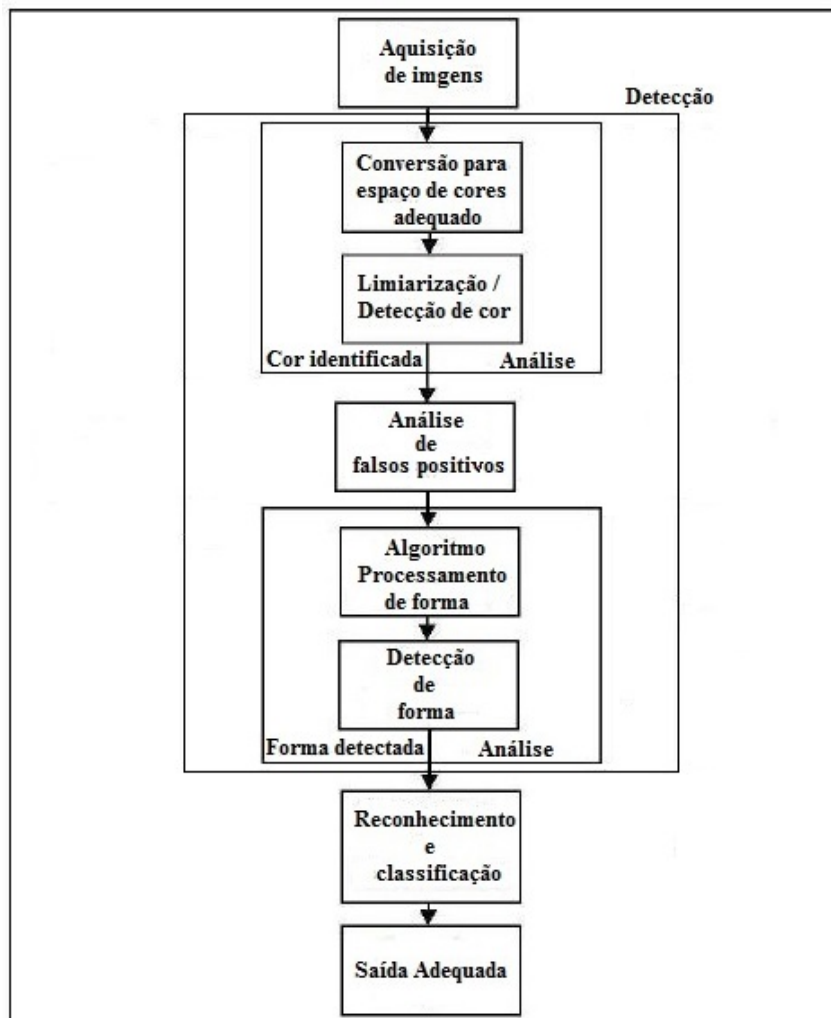


Figura 3.1: Diagrama de blocos que fornece uma noção geral da metodologia adotada. Merecem destaque os dois subprocessos maiores: detecção e reconhecimento (indicado como “Reconhecimento e classificação”).

A Figura 3.1 apresenta uma visão geral do algoritmo desenvolvido. É possível perceber que o processamento principal pode ser dividido em dois grandes subprocessos: o de detecção e o de reconhecimento das placas de trânsito. Assim, o primeiro engloba tanto a detecção de cores, quanto a detecção de formas. Já o segundo realiza o reconhecimento por meio de máquinas de vetores de suporte.

Tendo em vista algumas utilidades de um sistema de detecção e reconhecimento, descritas no Capítulo 1, o método desenvolvido tem como foco segmentar e identificar as placas de trânsito em uma imagem digital. No entanto, esse método deve considerar algumas dificuldades características desse tipo de processamento, que serão descritas adiante.

Em linhas gerais, pode-se até citar um passo prévio a todos os outros, a aquisição de imagens que contenham um ou mais sinais de trânsito. Uma vez adquiridas, essas imagens podem ser submetidas ao sistema. O primeiro desafio na detecção desses sinais diz respeito à luminosidade, pois tais imagens podem ser obtidas em qualquer hora do dia e em diversas condições climáticas. Assim, foi preciso escolher um espaço de cores mais adequado para essa análise.

Feito isso, estabeleceram-se limiares que permitiram a identificação das placas. Entretanto, isso apresenta mais uma dificuldade, que se refere à ampla variação nas tonalidades de uma mesma cor. Além do mais, no caso específico do azul, existem tons muito semelhantes à cor do céu. Com esses limiares foi possível fazer a segmentação por cores, utilizando o processo de limiarização. Como já foi visto, a ideia de limiarização pode ser vislumbrada na equação 2.5, definida por Gonzalez e Woods [4], e a explicação de como ela é feita, em detalhes, neste trabalho encontra-se em 3.1.2.

É importante ressaltar que, inicialmente, foi necessário acrescentar uma borda à imagem de entrada, o que é denominado *padding*, fazer tratamentos morfológicos para reduzir os ruídos e remover algumas descontinuidades, bem como estabelecer o tamanho das placas que seriam detectadas.

Complementarmente, estipulou-se uma relação entre a altura e a largura das placas, sendo igualmente importante a análise da porcentagem de branco para as placas vermelhas. Essa abordagem tem como objetivo evitar falsos positivos, isto é, detectar o mínimo de objetos que não sejam placas como sinais de trânsito.

Por fim, para facilitar o reconhecimento dos sinais de trânsito, classificou-se previamente a forma geométrica de cada sinal. Assim, foram utilizadas máquinas de vetores de suporte binárias. O programa executa, então, apenas o conjunto de máquinas específicas para cada forma e obtém a classificação final da placa de trânsito.

3.1 Detecção

3.1.1 Espaço de Cores

A primeira etapa do processo de detecção é a conversão da imagem de entrada do sistema para o espaço de cores YCbCr, sendo que esse passo se refere ao quadro “Conversão para espaço de cores adequado” da Figura 3.1.

Depois desse passo, o intuito é segmentar a imagem de acordo com as possíveis cores de uma placa, que de acordo com os bancos de imagens utilizados neste trabalho são: vermelho, azul e amarelo.

Uma segmentação similar foi proposta anteriormente por Maldonado et al. [30] no espaço de cores HSV. Não obstante, durante o desenvolvimento deste trabalho, foi verificada a possibilidade de realizar uma segmentação no espaço YCbCr, já que de forma similar ao HSV, ele separa as informações de luminância e crominância.

Essa é uma característica desejável nesse tipo de sistema, haja vista que o efeito da luminosidade pode alterar bastante os valores numéricos das componentes do espaço RGB, tornando ineficaz a análise dos limiares, os quais podem ser muito inconstantes. Isso é ainda mais relevante ao se considerar que as imagens das placas de trânsito podem ser obtidas em diferentes condições climáticas e em diferentes períodos do dia.

3.1.2 Histogramas, limiarização e vizinhança

Essa subseção remete ao quadro “Limiarização / Detecção de cor” da Figura 3.1 Inicialmente foram selecionadas 8 tonalidades das cores desejadas (obtidas das placas dos bancos de imagens), como pode ser visto na Figura 3.2. Feito isso, para cada uma dessas tonalidades, foi elaborado um histograma para o nível Cb e outro para o nível Cr , como ilustra a Figura 3.3.



Figura 3.2: Algumas das diferentes tonalidades utilizadas para a confecção dos histogramas e análise dos limiares. (a) Quatro tonalidades da cor azul. (b) Quatro tonalidades da cor vermelha. (c) Quatro tonalidades da cor amarela.

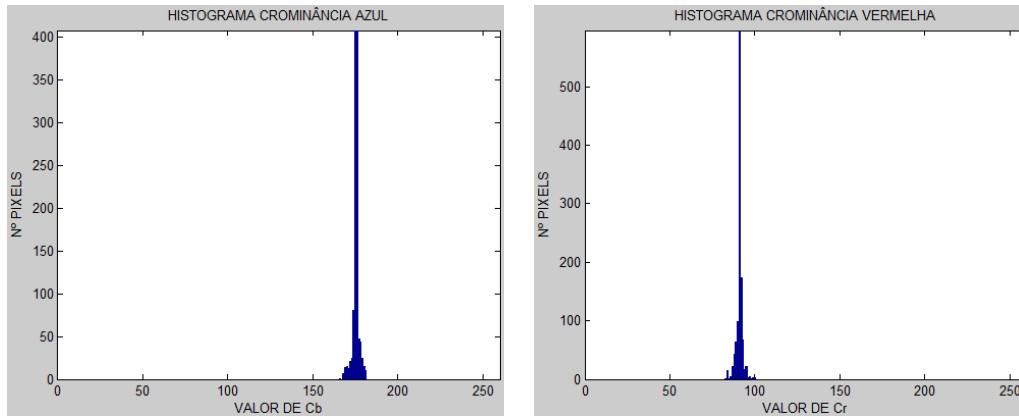
De posse desses histogramas, primeiro se considera a crominância azul. Assim, calcula-se a média do nível Cb verificado para cada tonalidade encontrada, conforme a equação 3.1:

$$\text{Nível Médio } Cb = \frac{\sum \text{Nível } Cb}{\text{Quantidade de Pixels}} \quad (3.1)$$

Em seguida, computa-se a variância dos níveis Cb identificados e, dessa forma, pode-se estabelecer os Limiares Superior e Inferior do nível Cb para cada tonalidade. Esses limiares são denominados limiares pontuais e são exibidos por 3.2 e 3.3.

$$\text{Limiar Superior Pontual} = \sum \text{Nível Médio } Cb + \text{Variância} \quad (3.2)$$

$$\text{Limiar Inferior Pontual} = \sum \text{Nível Médio } Cb - \text{Variância}. \quad (3.3)$$



(a)

(b)

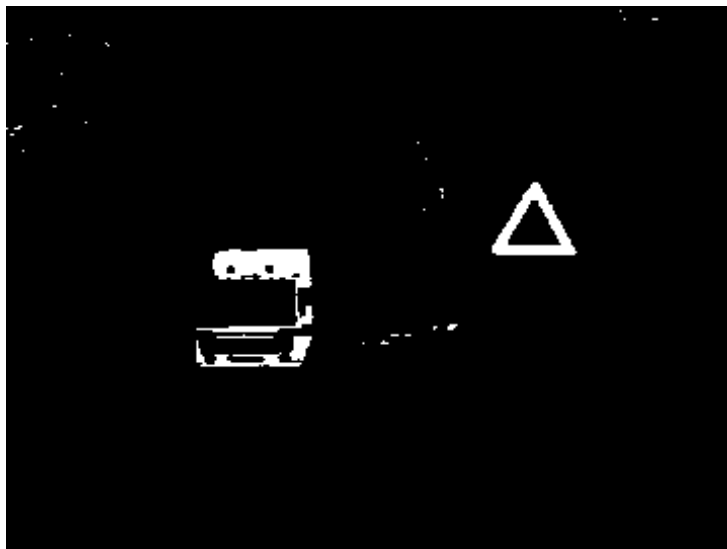
Figura 3.3: Histogramas de Cb e Cr obtidos a partir de uma das tonalidades de azul. O eixo das ordenadas apresenta a quantidade de *pixels* e o eixo das abcissas mostra o valor de cada componente. (a) Histograma gerado para a componente Cb . (b) Histograma para componente Cr .

A partir das equações anteriores, o Limiar Superior Global é definido como o maior dos Limiares Superiores Pontuais, assim como o Limiar Inferior Global é definido como o menor dos Limiares Inferiores Pontuais. Esses limiares globais é que são de fato utilizados pelo processo de segmentação por cor, pois representam uma estatística mais bem elaborada para esse tipo de análise.

Posteriormente, todas essas etapas de cálculo de limiares pontuais e globais foram também aplicadas para a crominância vermelha, isto é, para o nível Cr de cada tonalidade. Então, de posse dos limiares globais, tanto os referentes a Cb , quanto os relacionados a Cr , tornou-se possível o processo de limiarização no espaço YCbCr, ignorando a componente Y , luminância.



(a)



(b)

Figura 3.4: Exemplo da segmentação por cores adotada. (a) Imagem original, contendo uma placa de trânsito de interesse. (b) Imagem binarizada resultante da limiarização: Em branco, as porções da imagem que foram detectadas como vermelhas na imagem original; Em preto, tudo o que não é considerado como vermelho.

Convém, ainda nessa subseção, destacar que a vizinhança utilizada na segmentação por cor foi a vizinhança-8, mais completa que a vizinhança-4 e suficiente para esse tipo de detecção, não sendo necessária a análise da vizinhança-m. Diante disso, a Figura 3.4 mostra um exemplo de imagem binarizada, que é o tipo de imagem obtido como saída desse processamento inicial.

Nessa figura, os *pixels* brancos representam as intensidades que estão entre os Limiars Globais Inferiores e Superiores de C_b e C_r para a cor vermelha, conforme explicado anteriormente. Logo, a parte branca da imagem se refere a tudo aquilo que foi identificado como vermelho e a parte preta abrange o que não é de interesse, ou seja, tudo o que não foi considerado como vermelho, estando fora da faixa estabelecida para C_b ou da faixa determinada para C_r .

Percebe-se, porém, que essa limiarização por si só não é suficiente, sendo necessário examinar mais artifícios nessa análise. Tais aspectos serão descritos logo mais e servirão de base para a definição de certas propriedades, buscando desconsiderar os objetos que não são de interesse nesse processo, como o caminhão.

3.1.3 *Padding*

O sistema assume que as imagens de entrada, de preferência, não possuem placas que toquem as bordas da imagem, caso contrário, a classificação da forma e o reconhecimento desses sinais podem ser comprometidos. Por outro lado, para evitar que as indicações de placa detectada, como na Figura 3.5, extrapolem os limites da imagem, é necessário um *padding*, isto é, uma margem com tamanho definido de 10 *pixels* deve ser acrescida na imagem original. Esse passo pode ser identificado na Figura 3.1 como um dos meios que ajudam a promover uma “Saida Adequada” para o sistema.



(a)



(b)

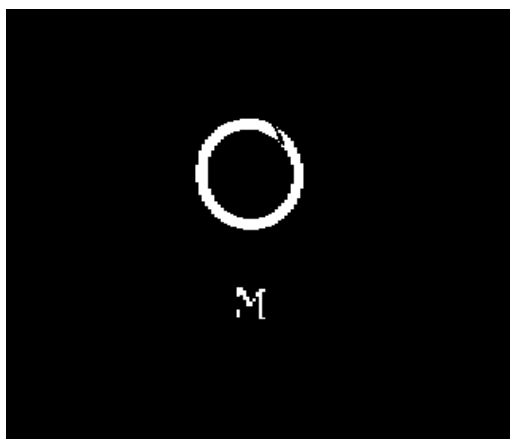
Figura 3.5: Exemplo da importância do *padding*, a margem extra acrescida na imagem de entrada do sistema. (a) Imagem original. (b) Imagem de saída com as devidas marcações características do processo de detecção.

3.1.4 Operações Morfológicas

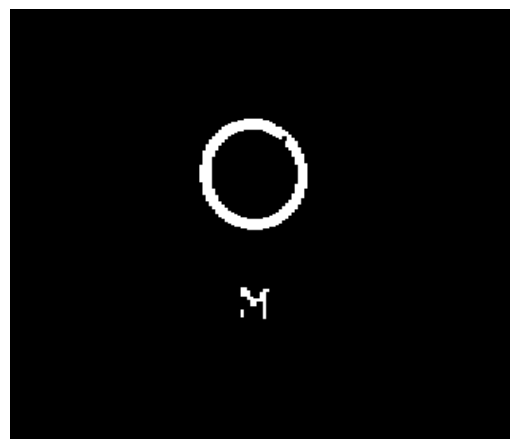
O próximo passo é aplicar as operações da morfologia matemática descritas no capítulo anterior, objetivando a remoção dos ruídos presentes na imagem que contém um ou mais sinais de trânsito. Além disso, essas operações também contribuem para remover descontinuidades causadas por possíveis objetos que apareçam obstruindo a visualização das placas, como galhos, folhas ou mesmo pessoas. Essa etapa pode ser identificada na Figura 3.1 como uma maneira de aprimorar a “Limiarização / Detecção de cor” do sistema.



(a)



(b)



(c)

Figura 3.6: Funcionamento e resultados das operações morfológicas feitas. (a) Imagem de entrada com a placa de trânsito. (b) Imagem resultante do processo de limiarização, sem a realização das operações morfológicas. (c) Imagem após a realização das operações da morfologia matemática, fechamento e abertura. É possível perceber o desaparecimento da ruptura presente na imagem anterior.

Portanto, foi feito o fechamento e depois a abertura nas regiões de interesse das imagens obtidas a partir da limiarização, melhorando significativamente a máscara binária, como indica a Figura 3.6. Os elementos estruturantes utilizados são quadrados, semelhantes aos da Figura 2.11 e têm um tamanho definido de 3×3 *pixels* para o fechamento e 2×2 *pixels* para a abertura.

Ainda acerca da Figura 3.6, nota-se que até uma simples sombra na imagem pode ser responsável por desconectar um elemento conexo. Isso pode prejudicar o processo de detecção, no sentido de impedir a correta classificação da forma e, conseqüentemente, inviabilizar o processo de reconhecimento desse sinal. Assim, as operações morfológicas foram escolhidas de modo que removesses essas discontinuidades, mas não alterassem o aspecto geral da placa de trânsito.

3.1.5 Tamanho das Placas

As placas consideradas no processo de detecção, para fins de levantamento de resultados, são aquelas que possuem uma altura válida de pelo menos 4% da altura total da imagem. Tal limitação é necessária para diminuir falsos positivos durante a detecção das placas e pode ser envolvida pelo quadro “Análise de falsos positivos” da Figura 3.1.



Figura 3.7: Diferentes tamanhos que as placas de trânsito podem assumir em uma imagem. Mais à esquerda e ao fundo (em amarelo), há uma placa praticamente imperceptível. Mais ao centro (em verde) tem-se uma placa de tamanho razoável que pode ser detectada e reconhecida pelo sistema. Mais à direita (em vermelho) existe uma placa não tão pequena como a primeira, no entanto ela também não tem tamanho suficiente quanto a segunda.

A Figura 3.7 fornece uma boa noção dos tamanhos das placas que são ou não consideradas. Nota-se que as duas menores placas, uma delas quase imperceptível ao fundo, não são computadas como erro de detecção, pois o sistema é feito para ignorar esses sinais. De fato, a única placa a qual o sistema deve detectar é a que está localizada entre essas duas menores.

3.1.6 Relação altura/largura

A razão altura/largura dos elementos conectados na máscara binária obtida da segmentação de cores ajuda a detectar as formas realmente condizentes com as placas. Dessa maneira a razão será considerada válida se:

$$0,7 < \frac{\text{altura}}{\text{largura}} < 1,4 \quad (3.4)$$

Com isso, placas como a da Figura 3.8 seriam consideradas no processo de detecção. Entretanto, sinais como o da Figura 3.9 (a) seriam ignorados, pois se percebeu que essas placas, via de regra, apresentam uma carga textual, que não é o escopo principal deste trabalho. Esse passo pode ser entendido como um meio de melhorar os resultados do sistema, estando dentro do quadro “Análise de falsos positivos” da Figura 3.1.



Figura 3.8: Sinal de trânsito que respeita a relação definida entre altura e largura, podendo ser detectado pelo sistema. Para ser mais específico, o valor numérico dessa razão é aproximadamente 0,97, dentro do intervalo previamente discutido.



(a)

(b)

Figura 3.9: Sinal de trânsito que não pode ser detectado pelo sistema. (a) Imagem original. (b) Imagem segmentada. Nota-se que a segmentação apresenta um ótimo resultado e que, embora esse sinal tenha mais de 40% da altura da imagem, sua razão altura/largura é de aproximadamente 0,61, estando fora do intervalo determinado pela Equação 3.4.

3.1.7 Porcentagem de Branco

Para as placas vermelhas ainda se definiu que a placa deve conter uma porcentagem de branco, estipulada entre 8% e 62% da área da placa. Essa etapa também está situada no quadro “Análise de falsos positivos” da Figura 3.1, pois também objetiva minimizar os falsos positivos.

Dessa forma, é necessário identificar a cor branca e, para tal, usou-se a componente Y do espaço YCbCr, definindo-se que um *pixel* seria branco se ela fosse maior que o valor 100. Complementarmente, para os *pixels* superiores a 100, utilizou-se a fórmula definida por Maldonado et al. [30], ainda no espaço RGB:

$$f(R, G, B) = \frac{(|R - G| + |G - B| + |B - R|)}{3D} \quad (3.5)$$

Na qual valores de f maiores ou iguais a 1 representam cores cromáticas; caso contrário, trata-se de um pixel branco. O valor de D foi estabelecido do mesmo modo que no artigo supracitado, isto é, $D = 20$.

3.1.8 Sobreposição de placas

Esse procedimento está relacionado com o quadro “Análise de falsos positivos” da Figura 3.1, dado que visa aprimorar os resultados do processo de detecção. Assim, é importante considerar a análise da sobreposição das placas, uma vez que podem existir sinais que possuam bordas de cor diferente ou ainda que apresentem algumas descontinuidades, mesmo depois da aplicação das operações morfológicas. Esse fator pode acabar segmentando uma única placa em várias outras, o que não faz sentido, como demonstra a Figura 4.11 (a) do capítulo seguinte.

Para se ter uma ideia melhor da análise que foi feita, pode-se supor que a Figura 3.10 mostra uma detecção de uma placa losangular amarela com uma borda suficientemente espessa de modo que o sistema possa interpretar a parte interna como uma outra placa. Nessa imagem limiarizada, os *pixels* brancos se referem aos *pixels* que foram considerados amarelos na imagem original e os pretos tratam-se dos *pixels* que não são de interesse.

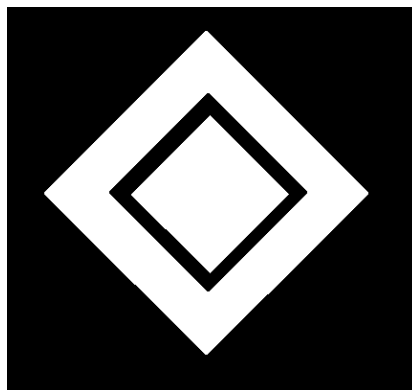


Figura 3.10: Imagem limiarizada que representa a detecção de uma placa losangular amarela. Os *pixels* brancos são aqueles considerados como amarelos em uma imagem de entrada. Os *pixels* pretos não são de interesse.

Diante disso, são computadas imagens contendo as fronteiras dos objetos detectados inicialmente pelo processo de limiarização, como mostra a Figura 3.11. Assim, é preciso preencher o interior das formas identificadas para poder analisar as intersecções, obtendo-se as imagens da Figura 3.12.

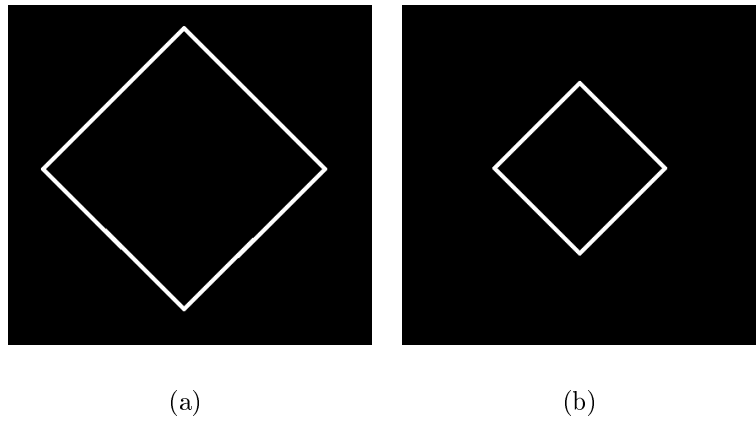


Figura 3.11: Representação das fronteiras das placas detectadas. (a) Fronteira externa detectada. (b) Fronteira interna detectada.

Finalmente, faz-se um *AND*, isto é, um “e” lógico entre a imagem da Figura 3.12 (a) e a imagem da Figura 3.12 (b) e atribui-se o resultado a uma matriz. Se houver algum *pixel* com valor diferente de 0 nessa matriz, os dois objetos detectados possuem uma intersecção e, portanto, fazem parte da mesma placa. Assim, deve-se seleccionar o maior objeto como sendo a placa detectada, descartando os objetos menores.

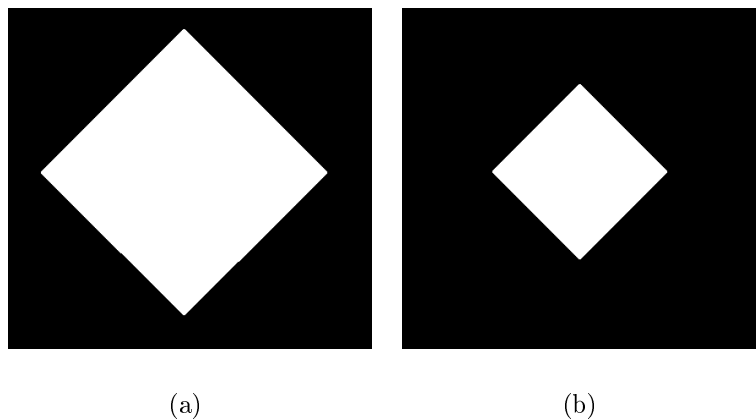


Figura 3.12: Preenchimento das formas identificadas. (a) Preenchimento da imagem mais à esquerda da Figura 3.11. (b) Preenchimento da imagem mais à direita da Figura 3.11.

3.1.9 Detecção de Formas

A seguir estão os passos que fazem parte dos quadros “Algoritmo Processamento de forma” e “Detecção de forma” da Figura 3.1.

- Círculo / Octógono:

A detecção da forma é feita inicialmente pela aplicação da *Transformada Circular de Hough*, para detectar as placas circulares e octogonais. Se a placa não for classificada assim, parte-se para a próxima análise, a da forma triangular.

- Triângulo:

Após detectada a cor da placa, faz-se a contagem dos *pixels* brancos da imagem segmentada, obtendo-se a *Área da Placa*, e calcula-se a '*Box Area*', isto é, a área do menor retângulo que engloba a placa identificada. Depois disso, comparam-se os valores obtidos das duas áreas e, com base em dados experimentalmente obtidos, se a razão for do tipo da equação 3.6, é bem provável que a forma em questão seja um triângulo.

$$Razão_{triangular} = \frac{Área da Placa}{Box Area} < 0,6 \quad (3.6)$$

Posteriormente, ainda se fez mais um teste para assegurar que a placa é de fato um triângulo. Compararam-se então, algumas linhas da parte superior da placa (6 linhas aproximadamente) com algumas linhas da parte inferior, como mostra a Figura 3.13. Se essa relação for da proporção de 1:2 ou 2:1, confirma-se a suspeita de que é um triângulo.



Figura 3.13: Ilustração da análise complementar que o sistema faz para confirmar a suspeita de que um sinal de trânsito é de fato triangular. Em amarelo, as indicações aproximadas das linhas superiores e inferiores que são usadas para o teste de proporcionalidade.

- Losango:

Para o teste do losango, a abordagem é semelhante à usada no teste das bordas do triângulo. Entretanto, é necessário tratar o centro da placa candidata antes, utilizando um processo semelhante ao preenchimento descrito na Subseção 3.1.8. Assim, compara-se a quantidade de *pixels* da cor de interesse dos quatro extremos com o centro, como mostra a Figura 3.14. Se nas quatro bordas a quantidade for metade da quantidade de *pixels* do centro, a placa é considerada losangular. Isso é feito independentemente do valor de sua *Razão_{triangular}*.



Figura 3.14: Ilustração de como é feita a detecção da forma losangular. Em vermelho, as marcações que indicam as regiões analisadas. Verifica-se que o interior da placa candidata deve ser preenchido com a cor de interesse antes desse teste, pois os *pixels* pretos podem causar classificações equivocadas de forma.

- Retângulo:

Se a $Razã_{\text{triangular}}$ de uma placa não for da forma da equação 3.6, deve-se ainda compará-la com outro limiar inferior, também obtido experimentalmente. Desse modo, a placa é retangular se atender à equação 3.7, sendo que a razão em questão agora pode ser considerada como uma razão retangular.

$$Razã_{\text{retangular}} = Razã_{\text{triangular}} > 0,75 \quad (3.7)$$

Nota-se que as detecções que tiverem uma $Razã_{\text{triangular}}$ entre 0,6 e 0,75 serão definidas como amorfas, não sendo detectadas como placas. Esse caso particular ocorre com o caminhão da Figura 3.4, por exemplo.

3.2 Reconhecimento

Abaixo, encontram-se os passos que fazem parte do quadro “Reconhecimento e classificação” na Figura 3.1. As duas primeiras subseções são passos prévios necessários para o processo de reconhecimento. A última subseção trata de como a classificação é realizada realmente pelo sistema.

3.2.1 Seleção da abordagem a ser utilizada

Para o reconhecimento dos sinais foram desenvolvidas máquinas de vetores de suporte (SVM’s). As utilizadas aqui, tratam-se de SVM’s binárias em paralelo, sendo que cada uma delas foi treinada para uma placa de forma e cor específica. Assim, toda SVM indica se a placa pertence ou não ao tipo indicado.

Optou-se por essa abordagem pelo fato de que as SVM’s lineares binárias são mais simples e requerem menos esforço computacional que outras técnicas, como as Redes Neurais Artificiais (ANN). Entretanto, acredita-se que resultados semelhantes teriam sido obtidos com o uso de outro método que não fosse baseado no uso das SVM’s.

3.2.2 Treinamento das SVM's

Foram utilizadas, em média, 15 imagens de dimensão 60×60 pixels para o treinamento de cada tipo de placa. As placas circulares podem ser vermelhas ou azuis. Para as circulares vermelhas, 5 tipos de placas diferentes foram treinadas, já para as azuis, 3 classes foram selecionadas. As placas triangulares são todas vermelhas, sendo que 6 tipos de placas foram utilizados. No que se refere às placas losangulares, elas são todas amarelas (estão somente no banco de imagens brasileiras) e contemplam 4 classes possíveis.

A Figura 3.15 mostra exemplos das diferentes imagens utilizadas para o treinamento de cada uma das SVM's, sendo possível perceber as diversas angulações e condições climáticas nas quais os sinais de trânsito podem aparecer. A Figura 3.16 exhibe todas as classes de placas que o sistema é treinado para reconhecer.



Figura 3.15: Algumas das placas de treinamento das SVM's. (a) Sinais vermelhos triangulares (*Pedestrian*). (b) Placas vermelhas circulares (*Zone 40*). (c) Placas azuis circulares (*Mini Roundabout*). (d) Sinais amarelos losangulares de treinamento (*Pista Sinuosa*).



Figura 3.16: Classes de placas de trânsito que o sistema é treinado para reconhecer. Nota-se que a placa “STOP” é detectada como vermelha circular, o que não interfere no reconhecimento desse sinal de trânsito.

Finalmente, é importante destacar que nenhuma das imagens de treinamento das SVM's foram utilizadas durante os testes.

3.2.3 Execução e Comparação de probabilidades

O reconhecimento usa como entrada as próprias imagens de saída do processo de detecção, sendo assim necessário saber a localização da região de interesse. Porém, antes de encaminhar essa região de interesse para as SVM's, é preciso redimensioná-la de acordo com as imagens de treinamento. Cabe destacar que o banco de imagens [9] divide as imagens de acordo com a condição climática: *Sunny*, *Cloud* e *Rainy*, imagens obtidas em dias de sol, em dias nublados e em dias de chuva respectivamente.

As SVM's são dispostas conforme a Figura 3.17, que considera inicialmente a cor dos sinais detectados e, mais adiante, verifica a forma de tais sinais, com o intuito de refinar a análise. Dessa maneira, percebe-se que essa abordagem permite um menor custo computacional e uma maior eficácia para o processo de reconhecimento, pois o sistema não precisa sempre executar todas as SVM's disponíveis, mas sim apenas aquelas pertinentes à cada forma geométrica e à cada cor.

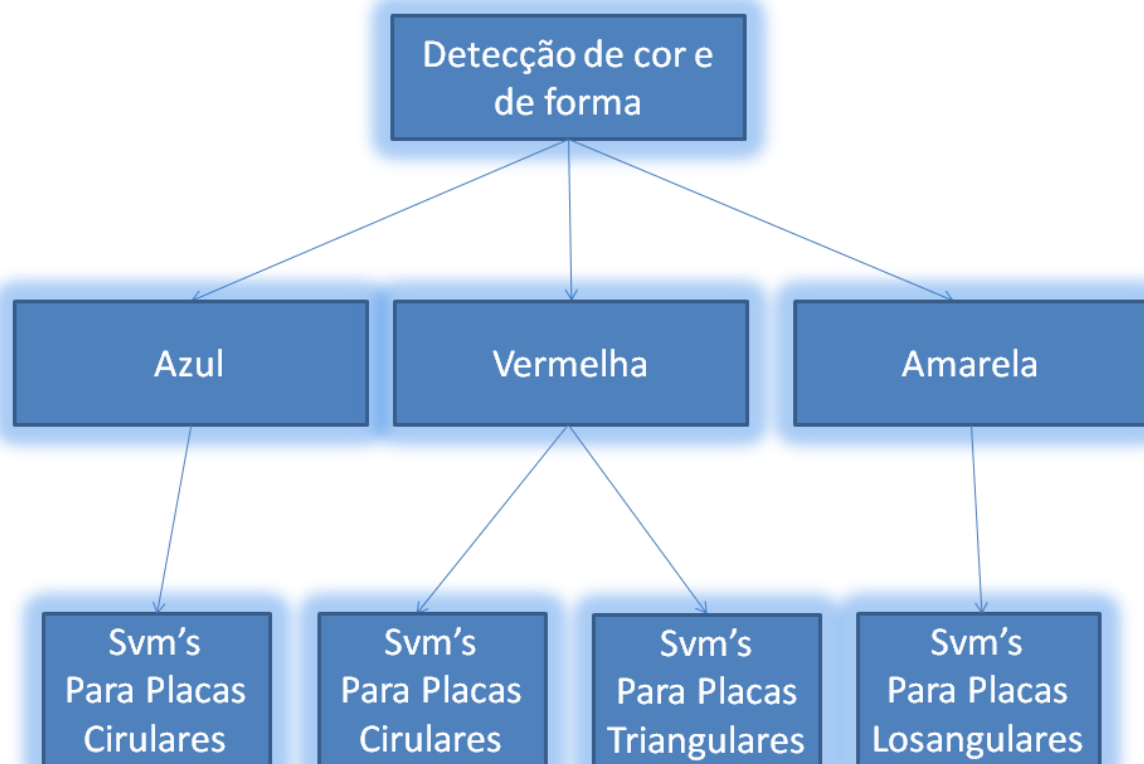


Figura 3.17: Organização dos SVM's. Nota-se o ganho computacional e a eficácia desse modelo no que diz respeito ao reconhecimento das placas de trânsito. Dessa forma, não é necessário executar todas as SVM's disponíveis, mas sim apenas aquelas condizentes com cada caso.

As SVM's utilizadas foram lineares. Em um mesmo grupo, elas são organizadas em paralelo, isto é, todas as máquinas desse subconjunto são executadas e fornecem uma probabilidade da imagem de entrada ser da classe para a qual a SVM em questão foi treinada. Assim, comparam-se as probabilidades, e analisa-se qual a classificação que apresenta maior grau de proximidade.

No Capítulo 4, serão demonstrados os resultados esperados e os resultados obtidos pelo sistema, tanto para a detecção, quanto para o reconhecimento. Também serão detalhadas as saídas de cada subprocesso.

Capítulo 4

Resultados

Neste capítulo são mostrados os resultados obtidos pelo sistema detalhado no capítulo anterior. Inicialmente, como um resultado intermediário, vale exibir o tipo de histograma obtido para a definição dos limiares, explicada em 3.1.2. São ainda ilustradas algumas saídas do sistema, referentes à detecção e ao reconhecimento dos sinais de trânsito. Quanto aos resultados propriamente ditos, também é importante tratar do conjunto de testes utilizado, definindo-se o que deve ser de fato um acerto ou um erro.

Nesse sentido, foram obtidos dois tipos de resultados finais: os referentes ao processo de detecção, bem como os que dizem respeito ao reconhecimento dos sinais de trânsito. Isso foi feito dessa maneira, pois o segundo processo tende a apresentar um número menor de testes, uma vez que algumas placas podem não ser detectadas, logo não sendo possível a sua classificação.

4.1 Histogramas obtidos

Cabe destacar que, antes de escolher o espaço YCbCr, o HSV e inclusive o RGB foram considerados. Desse modo, a partir da imagem da Figura 4.1, que mostra um dos tons de vermelho possíveis para uma placa, foram feitos histogramas para as componentes dos três espaços, exceto para a componente Y do YCbCr. Isso não foi necessário, uma vez que os limiares utilizados procuram minimizar o impacto do brilho nas imagens.



Figura 4.1: Uma das tonalidades usadas para se encontrar os limiares referentes à cor vermelha.

Assim, os histogramas mostrados fornecem uma boa comparação entre as componentes dos três espaços de cores candidatos para esse trabalho. Porém, nota-se que o YcbCr apresenta uma menor dispersão geral na quantidade de *pixels* que assumem determinado valor (a distribuição se aproxima de uma distribuição laplaciana). Isso é fundamental para a análise de limiares, pois quanto menor a faixa de valores possíveis, mais precisa tende a ser a detecção. Esse aspecto foi comprovado na prática e contribuiu de maneira essencial para a escolha desse espaço.

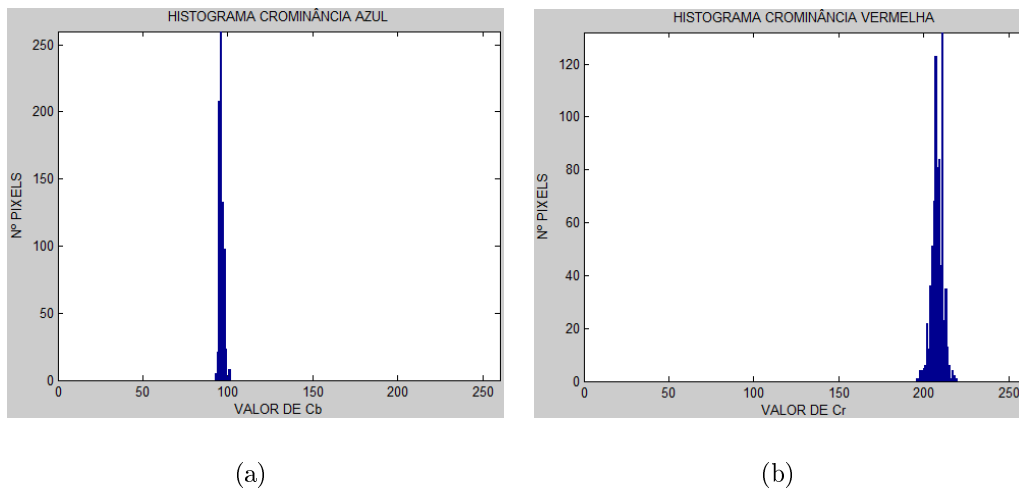


Figura 4.2: Histogramas YCbCr obtidos a partir da imagem da Figura 4.1. O eixo das ordenadas apresenta a quantidade de *pixels* e o eixo das abcissas mostra o valor de cada componente. (a) Histograma gerado para a componente Cb . (b) Histograma da componente Cr .

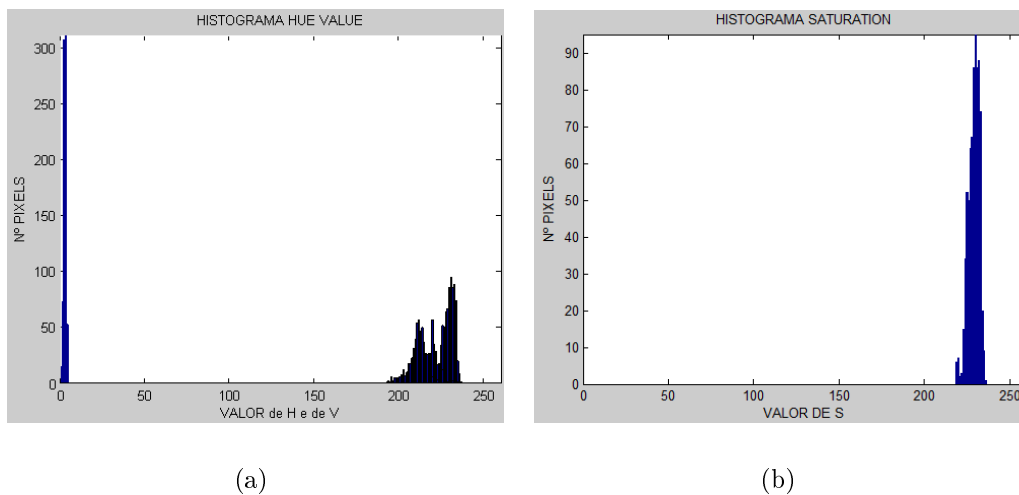
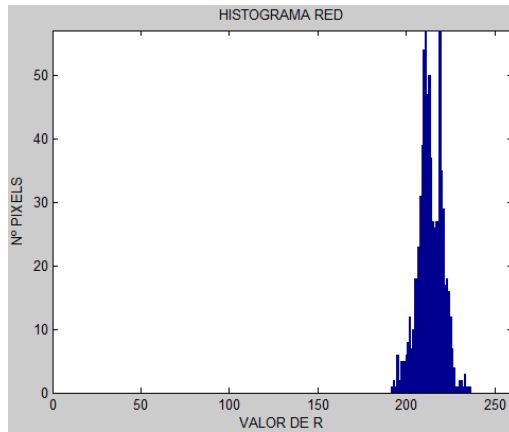
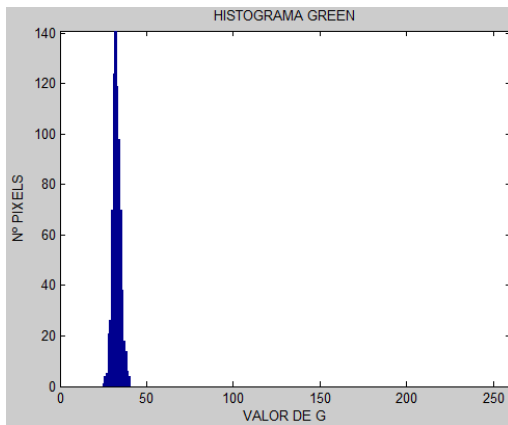


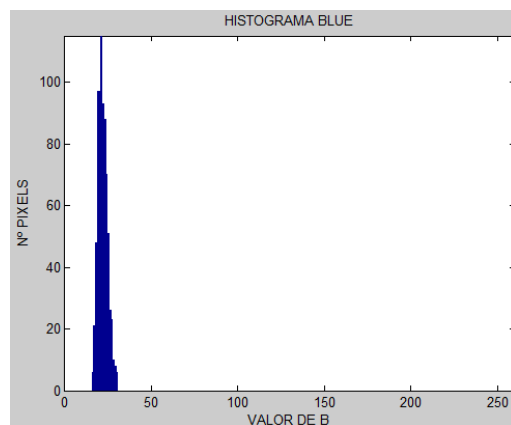
Figura 4.3: Histogramas HSV obtidos a partir de 4.1. O eixo das ordenadas apresenta o número de *pixels* e o eixo das abcissas mostra o valor de cada componente. (a) Em azul, histograma gerado para a componente H . Em preto, histograma gerado para a componente V . (b) Histograma da componente S .



(a)



(b)



(c)

Figura 4.4: Histogramas RGB obtidos a partir de 4.1. O eixo das ordenadas apresenta a quantidade de *pixels* e o eixo das abcissas mostra o valor de cada componente. (a) Histograma gerado para a componente *R*. (b) Histograma da componente *G*. (c) Histograma da componente *B*.

4.2 Configurações dos testes

O processo de reconhecimento utiliza como entrada as saídas da detecção. Assim, o conjunto de testes usado para verificar a precisão do detector de sinais de trânsito possui imagens obtidas durante o dia (pela manhã ou pela tarde), porém em diversas condições climáticas. Boa parte delas têm outros objetos, ou fundo, com cores similares às cores das placas, por exemplo, a presença do céu, que pode ter uma tonalidade de azul semelhante à de algumas placas. Isso é relevante, porque ajuda a aferir a taxa de falsos positivos, isto é, aqueles objetos que são detectados como se fossem sinais de trânsito, mas realmente não são.

Essas imagens têm diferentes resoluções espaciais e podem conter zero, uma ou mais placas. As imagens que não possuem placas são poucas e a sua utilização como teste é mais um artifício para analisar a taxa de falsos positivos. No caso de imagens que contenham mais de um sinal o qual possa ser detectado, os erros e acertos são contabilizados por placa existente e não por imagem. Dessa maneira, um sinal não detectado só é enquadrado como um erro se ele estiver dentro do critérios estabelecidos no Capítulo 3, como o tamanho das placas, 3.1.5, e a relação definida entre altura e largura, 3.1.6.

Vale destacar que o banco de testes tem imagens de placas do Reino Unido, também utilizado por Gao et al. [9] e de placas do Brasil, adquiridas durante o desenvolvimento deste trabalho. As primeiras são divididas em três categorias: *Sunny*, que possui imagens obtidas em condições ensolaradas; *Cloudy*, a qual tem imagens adquiridas quando o tempo estava nublado; e *Rainy*, que compreende imagens conseguidas em condições chuvosas. Os sinais desse banco ainda podem ser vermelhos ou azuis e possuem a forma circular, triangular ou retangular. Por outro lado, o segundo grupo de imagens contém basicamente placas amarelas, que são losangulares e retangulares. Além disso, ele abrange alguns sinais vermelhos circulares e um azul retangular.

Convém ressaltar que os resultados desse trabalho foram obtidos com o uso da ferramenta computacional MATLAB [40], utilizada na execução dos dois processos supracitados. O sistema operacional utilizado foi o *Windows 7 Home Premium 64 bits* e o processador foi um *Intel Core i5-2410M CPU* (Unidade Central de Processamento, do inglês *Central Processing Unit*) 2.30 *Gigahertz* com memória de 4,00 *Gigabytes*.

4.3 Saídas do sistema

Como já foi explicado, o sistema identifica uma placa na imagem e já faz uma marcação no local desse sinal, a qual possui a cor e a forma da placa detectada. Especialmente para as formas triangulares, a marcação acompanha a orientação da placa, verificando se ela está de cabeça para baixo, como pode ser visto na Figura 3.5 do capítulo anterior. As placas octogonais, como “STOP” da Figura 4.6, são detectadas como circulares, o que não impede o correto funcionamento do sistema. Quanto ao reconhecimento, exibe-se uma mensagem de texto indicando qual a classificação feita pelo sistema.

Primeiro, verificam-se detecções e reconhecimentos bem sucedidos de sinais de trânsito nas diferentes condições climáticas. Na Figura 4.6, isso foi possível mesmo com alguns objetos na frente da placa. Além disso, essa figura, bem como as Figuras 4.9 e 4.10 mostram a capacidade do sistema de identificar mais de uma placa na imagem, sem desconsiderar a cor e a forma de cada uma, como foi descrito anteriormente.



`classificacao = TURN left`

Figura 4.5: Detecção e reconhecimento de uma placa azul circular em condições ensolaradas. Na parte de cima, a marcação que indica o local da placa. Na parte de baixo, a classificação do sistema para o sinal. A imagem original é do banco *Sunny* [9].



`classificacao = zone 20`
`classificacao = STOP`
`classificacao = no ENTRY`

Figura 4.6: Detecção e reconhecimento de placas vermelhas com oclusão parcial. A marcações indicam o locais das placas e mais abaixo estão as classificações do sistema para os sinais. Apesar dos galhos na frente da placa, tanto o processo de detecção, quanto o de reconhecimento foram bem sucedidos. A imagem original é do banco *Sunny* [9].

Algumas das saídas exibidas aqui resumem os tipos de problemas encontrados no processo de detecção. Na Figura 4.12, por exemplo, o sistema confunde a lanterna traseira do carro com uma placa, configurando um falso positivo. Além disso, a Figura 4.11 mostra a importância do critério estabelecido em 3.1.8, pois a borda preta pode fazer com que o processo de segmentação divida a placa em mais de um objeto de interesse.



Figura 4.7: Detecção e reconhecimento de uma placa vermelha triangular. Há a indicação da cor e da forma no local onde a placa foi detectada. Na parte de baixo está a classificação feita pelo sistema. A imagem original é do banco *Cloudy*.



Figura 4.8: Detecção e reconhecimento de uma placa vermelha circular. Indicação da cor e da forma no local onde a placa foi detectada. A parte de baixo mostra a classificação feita pelo sistema. A imagem original é do banco *Rainy*.



Figura 4.9: Detecção de duas placas em uma mesma imagem: uma vermelha circular e uma amarela losangular. Nota-se que as marcações possuem as mesmas cor e forma das respectivas placas. A imagem original é do banco de placas brasileiras.



Figura 4.10: Detecção de duas placas losangulares amarelas em uma mesma imagem. As marcações possuem as mesmas cor e forma dos sinais detectados. A imagem original é do banco de placas brasileiras.



(a)

(b)

Figura 4.11: Detecção de uma placa amarela losangular. Imagem original é encontrada em [10]. (a) Percebe-se a quantidade de falsos positivos antes do análise da sobreposição das placas encontradas. (b) Marcação única para a placa identificada, após considerar o critério definido em 3.1.8.



Figura 4.12: Detecção da lanterna traseira do veículo como uma placa vermelha triangular. A imagem original é do banco *Rainy* [9].

4.4 Detecção

4.4.1 Resultados Obtidos

A seguir são exibidas as matrizes de confusão que representam os resultados referentes à detecção dos sinais de trânsito. Convém destacar que os nomes das classes reais, isto é, os resultados esperados, são exibidos na primeira coluna das tabelas. Em contrapartida, os nomes das classes preditas, que são os resultados obtidos pelo sistema, estão na linha de cima de cada matriz apresentada.

Desse modo, para exemplificar como essas matrizes devem ser interpretadas, pode-se analisar uma das linhas da Tabela 4.1: das 11 placas azuis circulares presentes no banco *Sunny*, 9 foram de fato detectadas como azuis circulares, mas 2 sequer foram identificadas como placas, ou seja, os chamados falsos negativos, representados por FN. Ainda na mesma tabela, percebe-se que 2 objetos que não eram placas foram identificados como placas azuis retangulares, tratando-se de falsos positivos (FP). A penúltima coluna apresenta a quantidade total dos sinais utilizados, ao passo que a coluna final representa a precisão (Prec.), taxa de acerto, para cada tipo de placa.

Assim, ao invés de exibir os resultados da detecção da cor e da forma das placas separadamente, elas agrupam esses dois atributos para que se tenha uma noção mais adequada dos resultados. Entretanto, ainda é possível verificar quando o sistema acerta a cor, mas erra a forma de um sinal, por exemplo, na Tabela 4.2 um dos sete sinais azuis circulares foi detectado como azul retangular.

A partir dessas tabelas, percebe-se que a taxa de acerto da detecção para os bancos britânicos *Sunny*, *Cloudy* e *Rainy* é por volta de 95%, com 14 falsos positivos; 78% com 11 falsos positivos; e 65%, com 8 falsos positivos, respectivamente. Essa redução progressiva da precisão já era esperada, pois mesmo com a utilização do espaço YCbCr, os limiares que definem as cores azul, vermelha e amarela variam de acordo com as condições luminosas.

Comparativamente, a precisão do sistema com relação à detecção dos sinais de trânsito brasileiros está em torno de 90%, com 19 falsos positivos. Por fim, analisando o resultado geral do processo de detecção, a taxa é superior a 83%.

Tabela 4.1: Resultados da detecção em condições ensolaradas (*Sunny*). Na primeira coluna estão os resultados esperados e na primeira linha estão os resultados obtidos. O valor em cada célula da tabela refere-se a quantidade de placas de um determinado tipo. FP significa falsos positivos e FN falsos negativos. ‘Circ.’ significa Circular, ‘Triang.’ Triangular e ‘Ret.’ Retangular. A última coluna é a taxa de acertos de cada tipo.

	Azul Circ.	Azul Ret.	Vermelho Circ.	Vermelho Triang.	Azul Triang.	Vermelho Ret.	Amarelo Ret.	FN	Σ	Prec.
Azul Circular	9	-	-	-	-	-	-	2	11	0,82
Azul Retangular	-	1	-	-	-	-	-	0	1	1,00
Vermelho Circular	-	-	11	-	-	-	-	0	11	1,00
Vermelho Triangular	-	-	-	14	-	-	-	0	14	1,00
FP	-	2	-	1	3	8	0	-	14	-
Total	-	-	-	-	-	-	-	2	37	0,95

Tabela 4.2: Resultados da detecção em condições nubladas (*Cloudy*). A primeira coluna indica os resultados esperados, enquanto a primeira linha representa as classificações feitas pelo sistema. ‘Circular’, ‘Triangular’ e ‘Retangular’ são abreviadas como na Tabela 4.1. FP significa os falsos positivos e FN os falsos negativos. A última coluna refere-se à porcentagem de acertos do sistema.

	Azul Circ.	Vermelho Circ.	Vermelho Triang.	Vermelho Ret.	Azul Ret.	Amarelo Ret.	FN	Σ	Precisão
Azul Circular	5	-	-	-	1	-	1	7	0,72
Vermelho Circular	-	5	-	-	2	-	1	8	0,63
Vermelho Triangular	-	-	8	-	-	-	0	8	1,00
FP	-	1	2	7	-	1	0	11	-
Total	-	-	-	-	-	-	-	23	0,78

Tabela 4.3: Resultados da detecção em condições chuvosas, usando o banco *Rainy*. Na primeira coluna estão os resultados esperados e na primeira linha, encontram-se os resultados obtidos. ‘Circular’, ‘Triangular’ e ‘Retangular’ aparecem como ‘Circ.’, ‘Triang.’ e ‘Ret.’ respectivamente. FP diz respeito aos falsos positivos e FN aos falsos negativos. Na última coluna está a taxa de acerto do sistema.

	Azul Circ.	Azul Circ.	Vermelho Circ.	Vermelho Triang.	Vermelho Ret.	Amarelo Triang.	FN	Σ	Precisão
Azul Circular	13	-	-	-	-	-	2	15	0,87
Azul Retangular	-	-	-	-	-	-	1	1	0,00
Vermelho Circular	-	-	6	-	6	-	2	14	0,43
Vermelho Triangular	-	-	-	3	-	-	1	4	0,75
FP	-	1	-	1	5	1	-	8	-
Total	-	-	-	-	-	-	-	34	0,65

Tabela 4.4: Resultados da detecção usando o banco de imagens brasileiras. ‘Am.’ e ‘Verm.’ significam Amarelo e Vermelho respectivamente. ‘Los.’ representa a forma Losangular; ‘Ret.’ Retangular; ‘Tri’ triangular; e ‘Circ.’ Circular. As classes reais estão na primeira coluna e as classes previstas na primeira linha. FP são os falsos positivos e FN os falsos negativos. A última coluna refere-se aos acertos do sistema.

	Am. Los.	Am. Ret.	Azul Ret.	Verm. Circ.	Am. Tri.	Am. Ret.	Azul Tri.	Azul Circ.	Verm. Tri.	Verm. Ret.	FN	Σ	Prec.
Am. Los.	43	0	-	-	0	0	-	-	-	-	5	48	0,90
Am. Ret.	-	4	-	-	-	-	-	-	-	-	1	5	0,80
Azul Ret.	-	-	1	-	-	-	-	-	-	-	-	1	1,00
Verm. Circ.	-	-	-	7	-	-	-	-	-	-	-	7	1,00
FP	2	-	4	-	3	1	5	1	2	1	-	19	-
Total	-	-	-	-	-	-	-	-	-	-	-	61	0,90

4.5 Reconhecimento

As tabelas a seguir detalham os resultados obtidos no processo de reconhecimento dos sinais de trânsito. Inicialmente, pode-se verificar a precisão do sistema com relação às placas brasileiras. A partir da Tabela 4.4, percebe-se que foram detectadas 53 placas amarelas, sejam elas losangulares ou retangulares. Mas, conforme a Tabela 4.5, 36 delas puderam ser reconhecidas. Essa redução é esperada, inclusive para os outros bancos de imagens, pois o sistema pode errar a forma de uma placa, inviabilizando o processo de reconhecimento. Além disso, há placas detectadas que o sistema não foi mesmo treinado para reconhecer (devido à dificuldade de se capturar placas suficientes para o treinamento).

Tabela 4.5: Resultado do processo de reconhecimento de placas brasileiras. Na primeira coluna estão as classes reais, isto é, os resultados esperados. A primeira linha representa as classes preditas, que são os resultados obtidos. FN são os falsos negativos e FP são os falsos positivos. Todas as placas são amarelas losangulares.

	Lombada	Animais	Pista Sinuosa	Passagem de Pedestres	FN	Σ	Precisão
Lombada	11	3	-	-	-	13	0,85
Animais	-	8	-	-	-	8	1,00
Pista Sinuosa	1	-	9	-	-	10	0,90
Passagem de Pedestres	1	1	2	1	-	5	0,20
FP	-	-	-	-	-	-	-
Total	-	-	-	-	-	36	0,81

A tabela acima mostra que a precisão do reconhecimento para o banco de placas do Brasil é aproximadamente 81%, não havendo nenhum falso positivo. As demais tabelas referem-se as imagens de placas do Reino Unido, percebendo-se que as taxas de acerto são de 100% para *Sunny*, não apresentando nenhum falso positivo; 82% para *Cloudy*, com 3 falsos positivos, provenientes de falsos positivos da detecção; e 81% para *Rainy*, com 1 falso positivo. Dessa maneira, a taxa geral de acerto do processo de reconhecimento é maior que 87%.

Por fim, é possível verificar que nas tabelas dessa seção não há falsos negativos. Isso se deve, pois o sistema analisa o grau de proximidade entre a placa detectada e as classes de treinamento da respectiva SVM, como já foi explicado. O sinal será atribuído à classe que apresentar a maior semelhança. Logo, nenhum sinal fica sem classificação, mesmo aqueles que o sistema não foi treinado para reconhecer.

Tabela 4.6: Resultado do reconhecimento em condições ensolaradas (*Sunny*). ‘Round.’ significa *Roundabout*. Verticalmente estão as classes reais e horizontalmente estão as classes preditas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.

	Pedestrian	Road Narrows	Side Road	Give Way	Round.	Two Way Ahead	Mini Round.	Keep Left	Turn Left	Zone 20	No Entry	Zone 40	Zone 30	Stop	FN	Σ	P
Pedestrian	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	1
Road Narrows	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
Side Road	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	2	1
Give Way	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	2	1
Round.	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	1	1
Two Way Ahead	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	1	1
Mini Round.	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	2	1
Keep Left	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	5	1
Turn Left	-	-	-	-	-	-	-	-	3	-	-	-	-	-	-	3	1
Zone 20	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-	3	1
No Entry	-	-	-	-	-	-	-	-	-	-	3	-	-	0	-	3	1
Zone 40	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	2	1
Zone 30	-	-	-	-	-	-	-	-	-	-	-	-	1	0	-	1	1
Stop	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	2	1
FP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	34	1

Tabela 4.7: Resultado do reconhecimento em condições nubladas (*Cloudy*). ‘Round.’ significa *Roundabout*. Verticalmente estão as classes reais e horizontalmente estão as classes previstas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.

	Pedestrian	Road Narrows	Side Road	Give Way	Round.	Two Way Ahead	Mini Round.	Keep Left	Turn Left	Zone 20	No Entry	Zone 40	Zone 30	Stop	FN	Σ	P
Pedestrian	4	-	-	-	0	-	-	-	-	-	-	-	-	-	-	4	1,00
Road Narrows	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-
Side Road	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	0	-
Give Way	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	2	1,00
Round.	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	1	1,00
Two Way Ahead	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	0	-
Mini Round.	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	4	1,00
Keep Left	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	0	-
Turn Left	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	0	-
Zone 20	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-
No Entry	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	0	-
Zone 40	-	-	-	-	-	-	-	-	-	1	-	3	-	-	-	4	0,75
Zone 30	-	-	-	-	-	-	-	-	-	-	-	2	0	-	-	2	0,00
Stop	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-
FP	2	-	-	-	-	-	-	-	-	1	-	-	-	-	-	3	-
Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	0,82

Tabela 4.8: Resultado do reconhecimento em condições chuvosas (*Rainy*). ‘Round.’ significa *Roundabout*. Primeira coluna: classes reais. Primeira linha: classes preditas. FP são os falsos positivos e FN são os falsos negativos. P significa precisão.

	Pedestrian	Road Narrows	Side Road	Give Way	Round.	Two Way Ahead	Mini Round.	Keep Left	Turn Left	Zone 20	No Entry	Zone 40	Zone 30	Stop	FN	Σ	P
Pedestrian	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-
Road Narrows	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-
Side Road	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	0	-
Give Way	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	2	1,00
Round.	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	1	1,00
Two Way Ahead	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	0	-
Mini Round.	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	2	1,00
Keep Left	-	-	-	-	-	-	1	2	-	-	-	-	-	-	-	3	0,67
Turn Left	-	-	-	-	-	-	2	-	5	-	-	-	-	-	-	7	0,71
Zone 20	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	-
No Entry	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	1	1,00
Zone 40	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	4	1,00
Zone 30	-	-	-	-	-	-	-	-	-	1	-	-	0	-	-	1	0,00
Stop	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	-
FP	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	1	-
Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21	0,81

4.6 Comparação com outros trabalhos

Existe uma dificuldade em comparar este trabalho com os outros trabalhos da área, pois não existe um banco de imagens que seja universalmente utilizado. Porém, mesmo assim, é importante comentar os resultados já existentes na literatura.

Inicialmente, pode-se citar Piccioli et al. [16], os quais utilizaram placas triangulares e circulares em seus testes. Para verificar a detecção das placas triangulares foram utilizadas 600 imagens que continham um ou mais sinais, apresentando um resultado de 92% de acerto com 11 falsos positivos.

Para a detecção das circulares, foram usadas 200 imagens que os autores chamaram de “*easy images*”, se referindo à imagens que continham alto contraste entre o sinal de trânsito e o fundo, sendo portanto de fácil detecção. A precisão foi de 96% com 25 falsos negativos. Por outro lado, foram usadas 200 “*hard images*”, ou seja, imagens com sinais em fundos confusos e conseqüente deterioração do limite do sinal, apresentando 93% de acerto e 27 falsos negativos. Quanto ao processo de reconhecimento, tratam de um método robusto com altas taxa de acerto, por volta de 98%, mesmo para sinais com um tamanho de 25×25 *pixels*.

Vega et al. [22] também desenvolveram um método que propõe a detecção e classificação dos sinais de trânsito, mas não o seu reconhecimento de fato. Nesse sentido, foram usados 174 sinais vermelhos e azuis, das formas circular, octogonal, triangular e retangular. Desses sinais, 141 foram detectados adequadamente e 114 tiveram suas formas identificadas de maneira correta, sendo 27 os erros na reconstrução da forma. Citam-se ainda 23 falsos positivos. Assim, a taxa de acerto da detecção é 81,03%, sendo que a precisão da identificação de forma é 80,85%.

Lopez e Fuentes [24] utilizaram 3 vídeos, cujas seqüências de imagens tinham tamanhos de 640×480 *pixels*, 720×480 *pixels* e 2272×1240 *pixels*. Esses vídeos foram categorizados em capturas referentes a rodovias, a estradas e a estacionamentos, respectivamente, sendo que a taxa média de acerto é de 97%. Os autores citam 6 falsos negativos e ainda afirmam que o método proposto é invariável quanto à rotação, podendo detectar sinais com cores e tamanhos vistos em qualquer orientação. Nota-se que não há classificação de forma nem reconhecimento dos sinais nesse trabalho.

Brkić et al. [27] utilizaram o método de detecção proposto por Viola e Jones [3]. Para o treinamento do sistema, usaram por volta de 1000 imagens de relativa baixa qualidade. Para os testes, um conjunto com 91 imagens foi usado, contendo 101 sinais, e outro grupo com 68 imagens, que possuía 72 placas. Além disso, usaram dois fatores de escala: 1.05 e 1,20. Para o primeiro fator, a detecção relativa ao primeiro grupo de testes teve 96% de acertos e 84% de falsos positivos, ao passo que a detecção do segundo grupo apresentou 93% de precisão, com 163% de falsos positivos. Para o segundo fator de escala, a precisão referente ao primeiro conjunto de testes foi de 93%, com 42% de falsos positivos, sendo que a taxa de acertos para o segundo grupo foi de 90%, apresentando 53% de falsos positivos.

Gavrila [26] utilizou um banco de 1000 imagens para teste, abrangendo imagens capturadas durante o dia (em condições ensolaradas ou chuvosas) e durante a noite. O autor afirma que o processo de detecção apresenta taxas de acerto superiores a 95% e que, em média, dois ou menos falsos positivos foram encontrados por imagem. O processo de reconhecimento apresenta uma precisão superior a 90%, com menos de 5% de falsos positivos. Contudo, Gavrila considera que condições adversas, tais como gotas de chuva, oclusões

parciais ou mesmo raios de sol direcionados diretamente à lente da câmera de captura podem reduzir as taxas de acerto da detecção em até 15%.

Gao et al.[9] obtiveram resultados em diferentes espaços de cores, sendo LCH (CIE-CAM97) e HSI os mais notáveis. Usando o primeiro espaço, os resultados da detecção são: 94% de acerto, com 23% de falsos positivos para *Sunny*; 90% de acerto, com 28% de falsos positivos para *Cloudy*; e 85% de acerto, com 27% de falsos positivos para *Rainy*. Em contrapartida, utilizando o segundo espaço a detecção resultou em: 88% de acerto, com 29% de falsos positivos para *Sunny*; 77% de acerto, com 37% de falsos positivos para *Cloudy*; e 73% de acerto, com 39% de falsos positivos para *Rainy*.

Esses autores ainda tratam de uma segmentação feita no espaço RGB, com 88% de acerto e 86% de falsos positivos para *Sunny*; 83% de acerto e 68% de falsos positivos para *Cloudy*; e 82% de acerto, com 65% de falsos positivos para *Rainy*. Cita-se ainda, sem muito detalhes, que o processo de reconhecimento tem uma precisão de 95%.

Finalmente, Maldonado et al. [30] utilizaram vídeos gravados de dentro de um automóvel em movimento. Essas sequências de vídeos foram então convertidas para imagens no formato BMP (*bitmap image file*) de tamanho 720×576 . Os autores se referem a 5 sequências: 3 referentes a condições ensolaradas, 1 correspondente a condições chuvosas durante o dia e, por fim, 1 que foi capturada no decorrer da noite.

Para o reconhecimento, utilizam-se três tamanhos distintos de máscaras de oclusão. Isso foi feito para verificar a precisão do sistema em situações reais nas quais possa haver algum objeto que dificulte a visualização das placas de trânsito. As probabilidades de sucesso são de 93,24% para as máscaras pequenas, 67.85% para as de tamanho médio e 44.90% para as máscaras grandes.

No próximo capítulo serão feitas as considerações finais sobre o trabalho, que envolvem as conclusões e os possíveis trabalhos futuros.

Capítulo 5

Conclusões

Neste trabalho foi proposto um método de detecção e reconhecimento de sinais de trânsito. A abordagem utilizada para o primeiro subprocesso foi a segmentação no espaço de cores YCbCr, que envolveu a elaboração de histogramas e a determinação de limiares. Foram também definidas algumas equações e alguns critérios para minimizar o número de falsos positivos encontrados. Além disso, foi proposta uma detecção das formas das placas como complemento para a análise das suas cores. O segundo subprocesso se baseia no uso de máquinas de vetores de suporte (SVM), para classificar as placas identificadas.

O sistema apresentou boas taxas de acerto para a detecção de cor e de forma, valendo destacar que uma detecção só é bem sucedida se a cor e a forma forem identificadas corretamente. Desse modo, as taxas foram de 95% para condições ensolaradas, 78% para condições nubladas e 65% em condições chuvosas, o que confirma a premissa da influência da luminosidade nas imagens, mesmo utilizando o espaço YCbCr. O desempenho com relação às placas brasileiras é próximo de 90%, configurando-se assim uma taxa de acerto maior que 83% para a detecção feita pelo sistema como um todo.

Quanto ao reconhecimento, a precisão para condições ensolaradas é de 100%, para condições nubladas é de 82% e em condições chuvosas é de 81%, a mesma verificada para o banco de placas do Brasil. A taxa total de acerto desse subprocesso é, portanto, ligeiramente maior que 87%. Os falsos positivos foram bem poucos, sendo que a maior taxa verificada foi aproximadamente 17% e se refere ao caso isolado do reconhecimento em condições nubladas.

Assim, a precisão do sistema foi satisfatória e torna a proposta deste trabalho competitiva diante dos resultados obtidos nessa área. Deve ser dito, contudo, que em algumas situações a porcentagem de falsos positivos da detecção pode chegar a 47,83%, quando as imagem de entrada são do banco de imagens obtidas em condições nubladas.

É relevante também citar as contribuições deste trabalho, as quais o diferenciam principalmente do trabalho de Maldonado et al. [30]. Primeiro pode-se destacar o espaço de cores escolhido para a segmentação, YCbCr, que não foi adotado por nenhum dos trabalhos referenciados no Capítulo 1. Além disso, a proposta da classificação das formas é bem simples e computacionalmente barata se comparada às demais abordagens. Finalmente, o reconhecimento baseado em um agrupamento de SVM's é essencialmente diferente, pois no caso de Maldonado et al., por exemplo, são usados dois tipos de SVM's. Esses autores utilizam SVM's lineares para a classificação de formas e SVM's de *kernels* gaussianos para o reconhecimento de fato.

Tal metodologia é diferente da deste trabalho, dado que o agrupamento descrito considera a classificação prévia das formas e das cores, ou seja, uma vez detectada a forma e a cor de uma placa já se sabe qual o subconjunto de SVM's a ser utilizado, não sendo necessário executar todas as máquinas de vetores de suporte disponíveis no sistema.

Convém ressaltar, por fim, os trabalhos futuros que podem ser desenvolvidos. Com o intuito de melhorar ainda mais os resultados obtidos, um dos focos é a redução dos falsos positivos. Nesse sentido, é possível promover um tratamento mais elaborado dos contornos das placas, que pode ser feito até mesmo no processo de reconhecimento para filtrar melhor os resultados da detecção. Pode-se também analisar as formas de dentro das fronteiras do objeto candidato a sinal de trânsito, buscando-se por padrões específicos. Com relação às condições climáticas, um processamento inicial, como um procedimento de realce de imagem, pode ser um bom caminho para deixar as imagens mais suscetíveis à análise de limiares. Além disso, pode-se comparar os resultados obtidos por SVM's com outras técnicas de reconhecimento como as Redes Neurais Artificiais (ANN) e as LDA's.

No mais, além da mensagem de texto, seria conveniente acrescentar como saída do processo de reconhecimento uma imagem padrão da placa reconhecida. Pode-se também tratar o caso no qual mais de uma placa da mesma cor aparecem tão próximas umas das outras a ponto de serem consideradas como um único objeto. Deve-se então evitar que essas placas configurem falsos negativos. Enfim, seria pertinente testar o sistema durante a noite.

Referências

- [1] CONTRAN-DENATRAN, *Manual Brasileiro de Sinalização de Trânsito: Sinalização Vertical de Regulamentação*, vol. 1. 2007. vii, 1, 2
- [2] Crown, “Know your traffic signs, official edition.” Disponível em https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/222621/dg_191955.pdf, 2007. Acessado em 01 de julho de 2015. vii, 2
- [3] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, 2001. vii, 5, 6, 58
- [4] R. Gonzalez and R. Woods, *Digital Image Processing*. Prentice Hall, 3rd ed., 2010. vii, viii, ix, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 29
- [5] Brandrops, “Színelmélet – a színterek (2. rész).” Disponível em <http://www.brandrops.com/design/szinelmelet-a-szinterek-2-resz/>, 2013. Acessado em 14 de junho de 2015. vii, 10, 11
- [6] K. Jack, *Video Demystified (Fourth Edition): A Handbook for the Digital Engineer*. Elsevier, 2005. vii, 11, 12, 14
- [7] BitJazz, “Synchrony: The perfect color converter.” Disponível em <http://www.bitjazz.com/en/products/synchrony/>. Acessado em 14 de junho de 2015. viii, 13
- [8] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd ed., 2010. ix, 25, 26, 27
- [9] X. Gao, K. Hong, L. Podladchikova, D. Shaposhnikov, N. Shevtsova, and P. Pas-smore, “Recognition of traffic signs based on vision models,” *Computer Vision and Image Understanding*, 2006. Manuscript number CV1U-06-359. xi, 3, 4, 42, 47, 48, 51, 59
- [10] D. Librarian, “<http://www.digital-librarian.com/caterpillars.html>.” <http://www.digital-librarian.com/sign.jpg>, 2006. Acessado em 03 de fevereiro de 2015. xi, 50
- [11] A. de la Escalera, J. Armingoland, and M. Mata, “Traffic sign recognition and analysis for intelligent vehicles,” *Image and Vision Computing*, vol. 11, no. 3, pp. 247–258, 2003. 1

- [12] Google, “Google self-driving car project.” Disponível em <http://www.google.com/selfdrivingcar/>, 2009-2015. Acessado em 09 de julho de 2015. 2
- [13] K. Brkić, “An overview of traffic sign detection methods,” *Department of Eletronics, Microeletronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska*, vol. 3, p. 10000, 2010. 3, 5, 6, 11
- [14] A. Ruta, Y. Li, and X. Liu, “Detection, tracking and recognition of traffic signs from video input,” *Intelligent Transportation Systems, ITSC 2008, 11th International IEEE Conference on, Beijing*, pp. 55–60, 2008. 3
- [15] H. Fleyeh and M. Dougherty, “Road and traffic sign detection and recognition,” *Proceedings of 16th Mini - EURO Conference and 10th Meeting of EWGT*, pp. 644–653, 2005. 4
- [16] G. Piccioli, E. D. Michelib, P. Parodi, and M. Campani, “Robust method for road sign detection and recognition,” *Image and Vision Computing*, pp. 209–223, setembro 1994. 4, 58
- [17] J. Canny, “A computational approach to edge detection,” *IEEE Trans. PAMI*, no. 8, pp. 679–698, 1986. 4
- [18] C. I. de l’Eclairage. <http://www.cie.co.at/>. Acessado em 20 de junho de 2015. 4, 5
- [19] CIE, *The CIE 1997 Interim Colour Appearance Model (Simple Version)*, CIE-CAM97s, 1998. Vienna: CIE Publication 131. 4
- [20] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, “Recognition of traffic signs based on their colour and shape features extracted using human vision models,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, p. 675–685, 2006. 4
- [21] X. Gao, L. Podladchikova, and D. Shaposhnikov, “Classification of images on the basis of the properties of informative regions,” *Pattern Recognition and Image Analysis*, vol. 13, pp. 349–352, 2003. 4
- [22] V. Vega, D. Sidibé, and Y. Fougerolle, “Road sign detection and shape reconstruction using gielis curves,” *International Conference on Computer Vision Theory and Applications, Rome: Italy*, 2012. 4, 58
- [23] J. Gielis, “A generic geometric transformation that unifies a wide range of natural and abstract shapes,” *American Journal Botany*, vol. 90, pp. 333–338, 2003. 4
- [24] L. D. Lopez and O. Fuentes, “Color-based road sign detection and tracking,” *Image Analysis and Recognition (ICIAR), Montreal, CA*, pp. 1138–1147, 2007. 5, 58
- [25] S. Varun, S. Singh, R. S. Kunte, R. D. S. Samuel, and B. Philip, “A road traffic tignal recognition system based on template matching employing tree classifier,” *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 3, pp. 360—365, 2007. 5

- [26] D. Gavrilu, "Traffic sign recognition revisited," *DAGM-Symposium*, p. p. 86–93, 1999. 5, 58
- [27] K. Brkić, A. Pinz, and Š. Segvić, "Traffic sign detection as a component of an automated traffic infrastructure inventory system," *Proceedings of AAPR/ÖAGM (Stainz, Austria)*., 2009. 5, 58
- [28] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Computational Learning Theory: Eurocolt '95*, pp. 23–37, 1995. 5
- [29] K. Tieu and P. Viola, "Boosting image retrieval," *International Conference on Computer Vision*, 2000. 6
- [30] B. Maldonado, A. Lafuente, J. Gil, M. Gómez, and F. López, "Road-sign detection and recognition based on support vector machines," *IEEE Transactions on Intelligent Transportation System*, vol. 8, pp. 264–278, junho 2007. 6, 30, 37, 59, 60
- [31] A. Ford and A. Roberts, "Colour space conversions." Disponível em <http://www.poynton.com/PDFs/coloureq.pdf>, 1998. Acessado em 22 de dezembro de 2014. 9
- [32] R. Gonzales, R. Woods, and S. Eddins, *Digital image processing using MATLAB*. Prentice Hall Upper Saddle River, 2004. 10, 14
- [33] L. M. Brys, *Página Dinâmica para Aprendizado do Sensoriamento Remoto. Dissertação (Mestrado em Sensoriamento Remoto)*. Centro de Pesquisa em Sensoriamento Remoto e Meteorologia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008. 12
- [34] R. S. Fibiger and H. Pistori, *Estudo Comparativo de Técnicas de Visão Computacional para Detecção de Pele Humana*. Universidade Católica Dom Bosco, 2004. 12
- [35] ITU-R BT.601-5, *Studio Encoding Parameters of Digital Television for Standard 4:3 and Widescreen 16:9 Aspect Ratios*, 1995. 14
- [36] C. Solomon and T. Breckon, *Fundamentos de Processamento Digital de Imagens*. LTC, 2013. 18
- [37] T. Mitchell, *Machine Learning*. McGraw Hill, 1997. 24
- [38] V. Vapnik, *The nature of Statistical learning theory*. Springer-Verlag, New York, 1995. 25
- [39] G. Batista, W. Silva, and O. Filho, "Classification of pattern using support vector machines: An application for automatic speech recognition," *ADVCOMP 2014, The Eighth International Conference on Advanced Engineering Computing and Applications in Sciences*, 2014. 25
- [40] Mathworks, "Matlab software." <http://www.mathworks.com/products/matlab/>, 1994-2015. Acessado em 05 de outubro de 2014. 47