

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROJETO DE GRADUAÇÃO**

DAVID WILKERSON KÜSTER

**IDENTIFICAÇÃO DE PLACA VEICULAR POR  
PROCESSAMENTO DE IMAGEM E OCR**

VITÓRIA – ES  
JANEIRO/2018

DAVID WILKERSON KÜSTER

## **IDENTIFICAÇÃO DE PLACA VEICULAR POR PROCESSAMENTO DE IMAGEM E OCR**

Parte manuscrita do Projeto de Graduação do aluno **David Wilkerson Küster**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador:  
Prof. Dr. Patrick Marques Ciarelli

VITÓRIA – ES  
JANEIRO/2018

DAVID WILKERSON KÜSTER

## IDENTIFICAÇÃO DE PLACA VEICULAR POR PROCESSAMENTO DE IMAGEM E OCR

Parte manuscrita do Projeto de Graduação do aluno **David Wilkerson Küster**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 04 de janeiro de 2018.

### COMISSÃO EXAMINADORA:



Prof. Dr. Patrick Marques Ciacelli  
Universidade Federal do Espírito Santo  
Orientador



---

Prof. Dr. Jorge Leonid Aching Samatelo  
Universidade Federal do Espírito Santo  
Examinador



---

MSc. Bruno Legora Souza da Silva  
Examinador

*Dedicado à minha família.*

## **AGRADECIMENTOS**

Agradeço aos colegas e mestres do curso de Engenharia Elétrica da UFES.

Agradeço ao meu orientador, pelas incontáveis contribuições, discussões e pelo aprendizado.

Agradeço aos membros da banca por avaliar este trabalho.

Agradeço aos meus amigos que, mesmo longe, se fizeram presentes.

Agradeço à minha noiva Fernanda, às minhas irmãs Deisiane e Dâmaris e ao meu irmão Deivison por toda força, principalmente nos momentos mais difíceis.

Agradeço a minha mãe Maria de Lourdes e ao meu pai Eliseu. Sem vocês eu não teria conseguido.

Muito obrigado!

## **RESUMO**

Os Sistemas Inteligentes de Tráfego são cada vez mais necessários devido à urbanização e aumento do número de veículos particulares. A implementação de Sistemas Inteligentes permitirá a identificação de veículos que foram roubados, a identificação automática de veículos em pedágios e fronteiras nacionais, a aplicação de multas por excesso de velocidade e uso de faixas exclusivas entre outros. O estudo dos Sistemas de Detecção de Placas Veiculares tem se intensificado muito nos últimos anos, com implementações e melhoramentos propostos por meio de diversas técnicas de processamento de imagens, processamento de sinais e uso de aprendizado de máquina. No presente trabalho, assim, é feita uma revisão de literatura buscando implementar uma abordagem de identificação de placas veiculares, através de processamento de imagens em ambiente de simulação para placas no padrão brasileiro, otimizando o algoritmo de forma que seja viável sua implementação em sistemas acoplados a câmeras de videomonitoramento de tráfego para identificação e localização de veículos.

Palavras-chave: Placa veicular. Identificação de veículos. Processamento de imagens.

## **ABSTRACT**

Intelligent Traffic Systems are becoming increasingly necessary due to urbanization and the increasing number of private owned vehicles. The implementation of such Intelligent Systems will make possible the identification of stolen vehicles, automatic identification of vehicles for toll collection and in international frontiers, the application of fines for speeding and the irregular use of exclusive tracks, among others. The study of Vehicle Plate Detection Systems has been intensified in recent years, with implementations and improvements proposed through various techniques of image processing, signal processing and machine learning. The present work consists of a literature review and the implementation of an approach to identify vehicle plates, through image processing in a simulation environment for Brazilian car plates, optimizing the algorithm so that it is feasible for implementation in systems mounted on video surveillance cameras for vehicle identification and localization.

Keywords: Car plate. Identification of vehicles. Image processing.

## LISTA DE FIGURAS

Figura 1 – Exemplos de Placas Veiculares.....	12
Figura 2 – Etapas da Identificação de Placa Veicular.....	16
Figura 3 – Operador Sobel.....	20
Figura 4 – Laplaciano .....	20
Figura 5 – Wavelet Bidimensional Discreta.....	21
Figura 6 – Resultado da Haar DWT.....	22
Figura 7 – Dilatação .....	23
Figura 8 – Erosão .....	23
Figura 9 – Morfologia .....	24
Figura 10 – Janelas Concêntricas.....	24
Figura 11 – Projeção Horizontal e Vertical .....	25
Figura 12 – Template Matching .....	27
Figura 13 – Posição das Placas Veiculares - Manual.....	29
Figura 14 – Fluxograma da Localização.....	30
Figura 15 – Imagem RGB lida e escala de cinza .....	31
Figura 16 – Histograma Equalizado.....	31
Figura 17 – <i>Lifting – Split/Predict/Update</i> .....	32
Figura 18 – Bordas Verticais – Haar Wavelet.....	32
Figura 19 – Imagem Binarizada .....	33
Figura 20 – Elemento Estruturante - Morfologia.....	34
Figura 21 – Morfologia - Fechamento .....	34
Figura 22 – Imagem após remoção dos elementos de borda e preenchimento .....	35
Figura 23 – Propriedades dos segmentos.....	35
Figura 24 – Resultado da Localização automática.....	37
Figura 25 – Fluxograma da Segmentação .....	37
Figura 26 – Placa RGB e escala de cinza .....	38
Figura 27 – Histograma Equalizado - Placa .....	39
Figura 28 – Binarização de Otsu e Adaptativa .....	39
Figura 29 – Elemento Estruturante - Placas .....	40
Figura 30 – Morfologia - Abertura.....	40
Figura 31 – Propriedades dos caracteres.....	41
Figura 32 – Resultado da Segmentação .....	41



Figura 33 – Padrão das Placas Brasileiras .....	42
Figura 34 – Fonte dos Caracteres Brasileiros .....	42
Figura 35 – Fonte Mandatory Segmentada.....	43
Figura 36 – Resultados da Localização automática (azul) .....	47
Figura 37 – Resultados da Localização automática (azul) .....	47
Figura 38 – Resultados da Localização automática (azul) .....	47
Figura 39 – Resultados da Localização automática (azul) .....	48
Figura 40 – Resultados da Localização automática (azul) .....	48
Figura 41 – Resultados da Localização automática (azul) .....	48
Figura 42 – Resultados da Localização no trabalho de Makaoui et al. (2016) .....	54
Figura 43 – Resultados da Localização automática com o método proposto (em azul).....	54
Figura 44 – Histograma de uma imagem.....	60

## LISTA DE TABELAS

Tabela 1 – Parâmetros de seleção da região de interesse .....	36
Tabela 2 – Parâmetros de seleção de caracteres .....	41
Tabela 3 – Sequências Alfanuméricas - ES.....	44
Tabela 4 – Resultado da Localização - placas brasileiras.....	46
Tabela 5 – Resultado da Segmentação - placas brasileiras.....	49
Tabela 6 – Resultado do Reconhecimento – Template Matching.....	50
Tabela 7 – Resultado do Reconhecimento – Conhecimento <i>a priori</i> .....	50
Tabela 8 – Matriz de Confusão – <i>Template Matching</i> .....	51
Tabela 9 – Resultado do Reconhecimento – Tesseract OCR .....	51
Tabela 10 – Resultados do Ciclo Completo .....	52
Tabela 11 – Resultado Localização - UCSD/Calit2 - Comparativo .....	53

## LISTA DE ABREVIATURAS E SIGLAS

ANN	<i>Artificial Neural Networks</i> – Redes Neurais Artificiais
BC/UFES	Biblioteca Central / UFES
CISNE/UFES	Laboratório de Computação e Sistemas Neurais (Pós-Graduação em Processamento de Sinais e Reconhecimento de Padrões/UFES)
CSV	<i>Comma-separated values</i> – Valores separados por vírgula
JPG	<i>Joint Photographic Experts Group</i>
MAT	Extensão de arquivo de dados MATLAB
DFT	<i>Discrete Fourier Transform</i> – Transformada Discreta de Fourier
DWT	<i>Discrete Wavelet Transform</i> – Transformada Wavelet Discreta
ITS	<i>Intelligent Traffic Systems</i> – Sistemas Inteligentes de Tráfego
LCEE/UFES	Laboratório de Computação da Engenharia Elétrica / UFES
LPD	<i>License Plate Detection</i> – Detecção de Placa Veicular
LPR	<i>License Plate Recognition</i> – Reconhecimento de Placa Veicular
OCR	<i>Optical Character Recognition</i> – Reconhecimento Óptico de Caracteres
RGB	<i>Red, Green and Blue</i> – Vermelho, Verde e Azul
ROI	<i>Region of Interest</i> – Região de Interesse
SWC	<i>Sliding Concentric Windows</i> – Janelas Concêntricas Deslizantes
THB	<i>Black Top-Hat</i>
THW	<i>White Top-Hat</i>
UFES	Universidade Federal do Espírito Santo
VLPCR	<i>Vehicle License Plate Character Recognition</i> – Reconhecimento de Placa Veicular
VLPL	<i>Vehicle License Plate Localization</i> – Localização de Placa Veicular

## LISTA DE SÍMBOLOS

$R$	Intensidade de Vermelho
$G$	Intensidade de Verde
$B$	Intensidade de Azul
$I(x, y)$	Ponto $(x, y)$ da imagem $I$ no domínio espacial
$I(u, v)$	Ponto $(u, v)$ da imagem $I$ no domínio da frequência
$T$	<i>Threshold</i> – limiar
$\sigma_b^2$	Variância entre duas classes
$P_i$	Probabilidade da $i$ -ésima classe
$\mu_i$	Média dentro da $i$ -ésima classe
*	Fechamento
◦	Abertura
$\Delta$	Operador Laplaciano
$\nabla^n$	$n$ -ésimo gradiente
$\frac{\partial^n}{\partial x^n}$	Derivada parcial em relação à $x$ de ordem $n$
$\psi_p$	<i>Wavelet</i> espacial discreta no ponto $p$
$f(i, j)$	Imagem no domínio espacial
$e(i, j)$	Elemento estrutural (círculo, retângulo etc.)
$\oplus$	Operador dilatação
$\ominus$	Operador erosão
$overlap(A, B)$	<i>Overlap Ratio</i> (razão de sobreposição) entre os conjuntos A e B

# SUMÁRIO

<b>1. APRESENTAÇÃO .....</b>	<b>11</b>
1.1. Introdução .....	11
1.2. Detecção e Reconhecimento de Placas Veiculares .....	11
1.2.1. Detecção de Placa Veicular .....	12
1.2.2. Segmentação e Reconhecimento de Caracteres .....	13
1.3. Justificativa .....	13
1.4. Objetivos e Escopo .....	14
1.4.1. Objetivos .....	14
1.4.1.1. Objetivo Geral .....	14
1.4.1.2. Objetivos Específicos .....	14
1.4.2. Escopo .....	14
1.5. Estrutura do Trabalho .....	14
<b>2. EMBASAMENTO TEÓRICO .....</b>	<b>16</b>
2.1. Identificação de Placa Veicular em uma imagem .....	16
2.2. Aquisição .....	16
2.2.1. Pré-processamento .....	16
2.2.1.1. Conversão .....	17
2.2.1.2. Binarização .....	17
2.2.1.3. Filtragem .....	18
2.3. Localização da Placa .....	19
2.3.1. Detecção de Bordas .....	19
2.3.1.1. Derivadas Espaciais .....	19
2.3.1.2. Sobel .....	19
2.3.1.3. Laplaciano .....	20
2.3.1.4. Haar Wavelet Discreto .....	21
2.3.2. Morfologia Matemática .....	22
2.3.3. Técnicas baseadas em Histograma .....	24
2.3.3.1. Janelas concêntricas .....	24
2.3.4. Aprendizado de Máquina .....	25
2.4. Segmentação e Reconhecimento .....	25
2.4.1. Segmentação de caracteres .....	25
2.4.1.1. Projeção Horizontal e Vertical .....	25

2.4.1.2.	Transformada Top-Hat.....	26
2.5.	Reconhecimento dos Caracteres.....	26
<b>3.</b>	<b>METODOLOGIA E IMPLEMENTAÇÃO.....</b>	<b>28</b>
3.1.	Aquisição .....	28
3.1.1.	Banco de Dados de Placas Brasileiras .....	28
3.1.2.	UCSD/Calit2 Car License Plate, Make and Model Database .....	29
3.2.	Localização .....	30
3.2.1.	Leitura, Redimensionamento e Conversão .....	30
3.2.2.	Processamento inicial.....	31
3.2.3.	Detecção de Bordas Verticais.....	31
3.2.4.	Binarização .....	33
3.2.4.1.	Morfologia Matemática.....	33
3.2.4.2.	Elementos de Borda e Preenchimento .....	34
3.2.5.	Segmentação e Seleção de Regiões Candidatas .....	35
3.3.	Segmentação .....	37
3.3.1.	Entrada, Redimensionamento e Conversão .....	37
3.3.2.	Processamento inicial.....	38
3.3.3.	Binarização .....	39
3.3.4.	Morfologia Matemática.....	39
3.3.5.	Segmentação e Seleção dos Caracteres.....	40
3.4.	Reconhecimento Óptico de Caracteres .....	42
3.4.1.	Template Matching .....	43
3.4.1.1.	Entrada e Redimensionamento .....	43
3.4.1.2.	Similaridade .....	43
3.4.1.3.	Conhecimentos <i>a priori</i> .....	44
3.4.2.	Tesseract OCR.....	44
<b>4.</b>	<b>RESULTADOS E DISCUSSÃO.....</b>	<b>46</b>
4.1.	Banco de Dados Brasileiro .....	46
4.1.1.	Localização .....	46
4.1.2.	Segmentação e Reconhecimento .....	49
4.1.2.1.	Segmentação .....	49
4.1.2.2.	Reconhecimento .....	49
4.1.2.3.	Template Matching .....	50

4.1.2.4.	Tesseract OCR.....	51
4.1.3.	Resultado da Primeira Etapa.....	52
4.2.	UCSD/Calit2 Car License Plate, Make and Model Database .....	52
<b>5.</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>55</b>
5.1.	Trabalhos Futuros.....	55
<b>6.</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>56</b>
<b>APÊNDICE A</b>	<b>.....</b>	<b>59</b>
A.1	Imagens no Domínio Espacial.....	59
A.1.1	Imagens Binárias e Escalares .....	59
A.1.2	Histograma .....	59
A.2	Imagens no domínio da Frequência .....	61
A.2.1	Transformada Discreta de Fourier .....	61
A.2.2	Transformada Inversa Discreta de Fourier .....	61
<b>APÊNDICE B</b>	<b>.....</b>	<b>62</b>
B.1	Localização .....	62
B.2	Segmentação .....	64
B.3	Reconhecimento de Caracteres.....	66

# 1. APRESENTAÇÃO

## 1.1. Introdução

Com o aumento contínuo do poder de processamento de computadores e da qualidade e velocidade das câmeras fotográficas e de vídeo, a um custo cada vez mais reduzido, os sistemas de visão computacional e processamento de imagem, áreas que demandam grande capacidade de processamento. Nas palavras de Klette (2014, p. vii, tradução nossa), “A Visão Computacional visa usar câmeras para analisar e entender cenas no mundo real. Essa disciplina estuda problemas metodológicos e algorítmicos assim como tópicos relacionados à implementação de soluções desenvolvidas”.

É possível citar uma gama de aplicações de detecção inteligente utilizando-se destes elementos, por exemplo: sistemas de detecção de cores, objetos, pedestres, reconhecimento de padrões, controle de qualidade de produtos, etc.

Todos esses sistemas que utilizam detecção inteligente podem ser denominados “Sistemas Inteligentes”. O presente trabalho se insere no âmbito dos denominados Sistemas Inteligentes de Tráfego (ITS – *Intelligent Traffic Systems*).

Os Sistemas Inteligentes de Tráfego se mostram cada vez mais necessários, uma vez que o mundo experimenta um *boom* de urbanização e com isso o aumento exponencial de carros particulares (ATIWADKAR et al., 2015). A implementação de Sistemas Inteligentes irá auxiliar a identificação de veículos que foram roubados, identificação automática em pedágios e fronteiras nacionais, além de aplicação de multas por excesso de velocidade e uso de faixas exclusivas, por exemplo.

Nesse cenário, os Sistemas de Detecção de Placas Veiculares emergem fortemente, tendo as pesquisas na área se intensificado muito nos últimos anos, com implementações e melhoramentos propostos por meio de diversas técnicas de processamento de imagens, processamento de sinais e uso de aprendizado de máquina (LAD; PATEL, 2015).

Cada técnica proposta na literatura possui certas limitações quanto às variações ambientais, de iluminação, do período do dia, da qualidade de captura da imagem, de sombreamento, além de outros fatores (VIDHYA; SUNDARI, 2014) e (SONAVANE; SONI; MAJHI, 2015).

## 1.2. Detecção e Reconhecimento de Placas Veiculares

A Detecção de Placas Veiculares se divide, basicamente, em duas etapas que são tratadas separadamente na literatura (MAKAOUI; GUENNOUN; GHOGHO, 2016):



- A detecção da posição em que se encontra a placa numa dada imagem, ou LPD (*License Plate Detection*) ou VLPL (*Vehicle License Plate Localization*); e
- O reconhecimento dos caracteres na placa veicular, ou LPR (*License Plate Recognition*) ou VLPCR (*Vehicle License Plate Character Recognition*).

Dentro da área de reconhecimento de padrões, cada uma destas duas etapas tem diversas subdivisões e técnicas de implementação, que possuem suas particularidades e resultados muito sensíveis a determinadas condições de iluminação, sombra, ângulo de captura da imagem, etc.

Apresentam-se a seguir, brevemente, ambas as etapas. Mais detalhes são fornecidos no Capítulo 2.

### 1.2.1. Detecção de Placa Veicular

O problema fundamental do LPD é o de determinar numa imagem onde estão localizadas as placas dos veículos, denominada de Região de Interesse (ROI - *Region of Interest*) (SONAVANE; SONI; MAJHI, 2015).

A complexidade se dá pelo fato de que pode haver numa imagem diversas regiões que se confundem com uma placa veicular, além de: o tamanho da placa variar muito; a localização não ser a mesma em diferentes tipos de veículos; as cores de fundo da placa variarem de acordo com o país e o tipo de veículo (particular, público etc.) (LAD; PATEL, 2015). Alguns exemplos estão ilustrados na Figura 1.

Figura 1 – Exemplos de Placas Veiculares



Fonte: próprio autor.

Diversas estratégias são utilizadas na literatura para detectar a posição da placa numa imagem. As principais são Detecção de Bordas (FARAJIAN; RAHIMI, 2014), Binarização (GILLY; RAIMOND, 2013), Morfologia Matemática (PANG, 2014) e técnicas de Aprendizado de Máquina (FARAJIAN; RAHIMI, 2014).

### 1.2.2. Segmentação e Reconhecimento de Caracteres

O LPR consiste em identificar os caracteres da placa que foi localizada. Para tanto, primeiro separam-se os diversos segmentos em unidades distintas, e cada qual é identificado isoladamente.

Utilizam-se, em geral, técnicas de processamento de imagens denominadas Reconhecimento Óptico de Caracteres ou OCR (*Optical Character Recognition*) para a última etapa do LPR.

As principais técnicas utilizadas são: Morfologia (Transformada *Top-Hat*) (MAKAOUI; GUENNOUN; GHOGHO, 2016) e Projeções para segmentação (BELVISI et al., 1999), bem como Aprendizado de Máquina (FARAJIAN; RAHIMI, 2014), *Template Matching* e outras técnicas de OCR para o reconhecimento dos caracteres (ATIWADKAR et al., 2015).

### 1.3. Justificativa

O presente trabalho busca ampliar a discussão sobre sistemas de identificação de placas veiculares brasileiras. O impacto social que sistemas de detecção e identificação de placas veiculares proporcionam se justifica no contexto das diversas aplicações-fim desses sistemas, dentre as quais as principais são:

- Cobrança automática de pedágio e estacionamento;
- Monitoramento e controle de tráfego;
- Identificação de veículos roubados;
- Segurança e aplicação da lei e de multas;
- Controle de fronteiras e aeroportos.

Dessa forma, notam-se as vantagens do uso de tais sistemas no aumento da segurança, eficiência e até mesmo soberania nacional, no que tange as aplicações fronteiriças, possibilitando um maior controle dos veículos que entram ou saem de tais perímetros, a um custo relativamente baixo.

Dadas as aplicações típicas desses sistemas, uma característica em comum é que se espera que tais sistemas atuem de maneira rápida (ATIWADKAR et al., 2015). Portanto, busca-se minimizar o tempo necessário para detecção e identificação de placas e, de igual forma, garantir precisão nos resultados.

---

## 1.4. Objetivos e Escopo

### 1.4.1. Objetivos

#### 1.4.1.1. Objetivo Geral

O objetivo geral deste trabalho é fazer a identificação de placas veiculares brasileiras em imagens estáticas, através da implementação de um sistema capaz de localizar a placa, segmentar e reconhecer os caracteres, retornando como saída uma variável do tipo *String*.

#### 1.4.1.2. Objetivos Específicos

- Desenvolver um algoritmo em plataforma de simulação computacional, composto de sub-rotinas, capazes de:
  - Identificar a posição de uma placa veicular na imagem;
  - Segmentar os caracteres para identificação individual;
  - Extrair os caracteres da imagem da placa.
- Desenvolver um método que forneça resposta rápida.

### 1.4.2. Escopo

Delimita-se o escopo deste trabalho na escolha e implementação de uma técnica de detecção e uma técnica de segmentação e reconhecimento de placa veicular em ambiente de simulação computacional, utilizando como entrada as imagens do banco de dados definido. Para tanto, assume-se que haverá apenas uma placa veicular a ser localizada por imagem. A abordagem implementada será comparada com outras metodologias, quando possível.

## 1.5. Estrutura do Trabalho

Este trabalho foi estruturado em 6 Capítulos, sendo a Apresentação do Tema o primeiro deles. No Capítulo 2 são discutidas as principais técnicas empregadas na literatura disponível a fim de tratar do problema de identificação de placas veiculares.

No Capítulo 3, é discutida a Metodologia empregada neste trabalho assim como os recursos utilizados e são detalhadas as abordagens propostas para simulação em ambiente computacional dos algoritmos de Localização, Segmentação e Reconhecimento de Caracteres.

---

O Capítulo 4 traz os resultados obtidos através dos testes dos algoritmos desenvolvidos nas imagens do banco de dados definido, bem como apresenta uma comparação entre os resultados de outras abordagens.

O Capítulo 5 traz as conclusões e propõe trabalhos futuros.

Finalmente, no Capítulo 6 são apresentadas as referências bibliográficas.

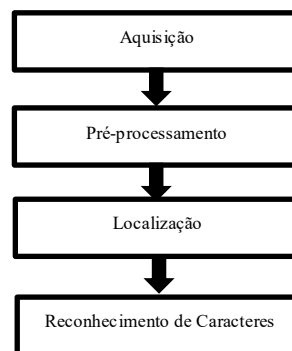
## 2. EMBASAMENTO TEÓRICO

### 2.1. Identificação de Placa Veicular em uma imagem

Todos os Sistemas de Identificação de Placas Veiculares são compostos basicamente de 4 etapas principais, que podem ser identificados na Figura 2 (SONAVANE; SONI; MAJHI, 2015): Aquisição/Captura, Pré-processamento, Localização da Placa, Reconhecimento de Caracteres.

Conforme exposto na Seção 1.2, a Localização da Placa ou LPD, consiste em identificar Regiões de Interesse na imagem onde a placa possa estar localizada. Já o Reconhecimento de Caracteres ou LPR, consiste em segmentar e identificar os caracteres da Região de Interesse encontrada.

Figura 2 – Etapas da Identificação de Placa Veicular



Fonte: Adaptado de (SONAVANE; SONI; MAJHI, 2015)

### 2.2. Aquisição

O processo de aquisição consiste em obter imagens através de câmeras fotográficas digitais ou câmeras de vídeo digitais. No Apêndice A é detalhada a definição de imagens digitais no domínio espacial e seu tratamento no domínio da frequência.

#### 2.2.1. Pré-processamento

De forma a se obterem resultados satisfatórios nos processos de localização da placa e reconhecimento de caracteres, além de menores tempos de execução e resposta, são necessárias técnicas de processamento de imagem que possam melhorar certas características da imagem antes da execução daqueles processos.

Encontram-se na literatura diversas técnicas de pré-processamento, tendo todas em comum os seguintes pontos (SONAVANE; SONI; MAJHI, 2015): Conversão, Binarização e Filtragem.

### 2.2.1.1. Conversão

As imagens obtidas no processo de aquisição em geral são imagens RGB de alta definição (SONAVANE; SONI; MAJHI, 2015). De modo a otimizar o tempo de processamento as imagens devem ser convertidas para uma imagem em escala de cinza.

As imagens RGB possuem 3 canais de cor, com 8-bits por canal, com total de 24-bits. Ao converterem-se as imagens para a escala de cinza, reduz-se para apenas um canal, que possui 8-bits de resolução, isto é, 256 intensidades (MAKAOUI; GUENNOUN; GHOGHO, 2016).

O valor da intensidade de cinza de um ponto  $(x, y)$  pode ser obtido através de uma média ponderada dos valores de vermelho, verde e azul do ponto, conforme Equação (2-1):

$$C = \frac{w_r R + w_g G + w_b B}{w_r + w_g + w_b} \quad (2-1)$$

onde:

$w_i$  representam os pesos atribuídos a cada canal;

C, R, G e B representam os valores inteiros de cinza, vermelho, verde e azul, respectivamente.

Um caso particular, é quando se adotam os pesos  $w_r = w_g = w_b = 1$ . Aplicando-se esses pesos na Equação (2-1) é obtida a Equação (2-2), de média simples (KLETTE, 2014):

$$C = \frac{R + G + B}{3} \quad (2-2)$$

### 2.2.1.2. Binarização

O processo de Binarização de uma imagem consiste em converter uma imagem em escala de cinza em uma imagem em preto e branco, conforme definido na Equação (2-3) (GONZALEZ, 2010):

$$G(x, y) = \begin{cases} 0, & \text{se } I(x, y) \leq T \\ 1, & \text{se } I(x, y) > T \end{cases} \quad (2-3)$$

Na Equação (2-3) T representa o limiar (*Threshold*). Seja, por exemplo,  $I(x, y)$  uma imagem composta de objetos claros sobre um plano escuro, dessa maneira os pixels do objeto e do fundo possuirão intensidades em dois grupos ditos dominantes (também denominados

modos). A forma natural de separar os objetos do fundo é determinar o limiar T que separa estes modos (GONZALEZ, 2010).

Há duas formas gerais de aplicar a limiarização. Quando T é constante e usado na imagem toda, o processo é chamado de *limiarização global*. Se o valor de T é variável ao longo da imagem, usa-se o termo *limiarização variável* (GONZALEZ, 2010).

Na Binarização de Otsu (OTSU, 1979), que é um tipo de limiarização global, o valor do limiar é ajustado de forma a maximizar a variância entre classes ( $\sigma_b^2$ ). Uma classe pode ser definida como um conjunto de objetos numa imagem que compartilham algumas características. Para o caso de duas classes, pode-se calcular  $\sigma_b^2$  conforme a Equação (2-4) (KLETTE, 2014):

$$\sigma_b^2 = P1(\mu_1 - \mu)^2 + P2(\mu_2 - \mu)^2 = P1P2(\mu_1 - \mu_2)^2 \quad (2-4)$$

onde:

P1 e P2 são as probabilidades das classes 1 e 2;

$\mu, \mu_i$  são a média global e a média dentro da classe  $i$ , respectivamente.

A Binarização Adaptativa de Bradley (BRADLEY; ROTH, 2007), que é um tipo de limiarização variável, consiste em utilizar a média das intensidades locais para definição de cada limiar local. Nessa técnica é possível reduzir-se os efeitos de sombreamento. Possui um tempo maior de processamento, mas em determinadas situações pode obter resultados mais precisos, em particular na presença de sombreamentos e padrões de interferência.

### 2.2.1.3. Filtragem

É possível utilizar as técnicas de processamento de imagens no domínio da frequência (vide Apêndice A.2 Imagens no domínio da Frequência) para aplicar filtros em uma imagem.

Técnicas de filtragem podem ser utilizadas para suavizar imagens e amenizar ruídos intrínsecos ao processo de captura de imagens digitais (filtros passa-baixas), como também para realçar detalhes em imagens (filtros passa-altas).

A filtragem pode, ainda, ser feita na representação espacial da imagem, porém neste caso são realizadas operações de convolução.

## 2.3. Localização da Placa

### 2.3.1. Detecção de Bordas

Uma borda em uma imagem pode ser definida como um ponto onde o brilho muda abruptamente (ATIWADKAR et al., 2015). Para realizar-se a detecção de bordas numa imagem, usam-se algumas técnicas matemáticas sendo as principais discutidas a seguir.

#### 2.3.1.1. Derivadas Espaciais

Para se detectar a variação abrupta do brilho, aproximam-se as derivadas espaciais de primeira ou de segunda ordem (nas direções de  $x$  e  $y$ ) da imagem  $I$ . As derivadas de primeira e segunda ordem estão representadas nas Equações (2-5) e (2-6), respectivamente (KLETTE, 2014).

$$\nabla = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} \quad (2-5)$$

$$\nabla^2 = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2-6)$$

#### 2.3.1.2. Sobel

Um dos mais utilizados detectores de bordas é o Operador Sobel, que consiste em operadores 3x3 que aproximam os gradientes em ambas as direções da imagem. Os operadores para as direções  $x$  e  $y$  são denominados  $I_x$  e  $I_y$  e estão representados a seguir nas Equações (2-7) e (2-8) (KLETTE, 2014).

$$I_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ 1 & 0 & 1 \end{bmatrix} \quad (2-7)$$

$$I_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2-8)$$

De forma a obter-se uma nova imagem com as bordas salientadas, faz-se a varredura dos operadores pela imagem  $I$ . Um exemplo da soma dos módulos dos resultados dos dois operadores  $I_x$  e  $I_y$  aplicados a uma imagem pode ser visto na Figura 3.



Figura 3 – Operador Sobel



Fonte: próprio autor

### 2.3.1.3. Laplaciano

O Operador Laplaciano é um operador isotrópico 2D similar ao Sobel, entretanto este usa apenas uma máscara para ambas as direções (FARAJIAN; RAHIMI, 2014).

O Operador Laplaciano é uma aproximação das derivadas de segunda ordem da imagem  $I$  (KLETTE, 2014). Uma aproximação comumente utilizada está expressa na Equação (2-9).

$$I_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2-9)$$

O Operador Laplaciano, representado por  $\Delta$ , é equivalente ao gradiente de segunda ordem da Equação (2-6) ( $\Delta = \nabla^2$ ) e pode ser aproximado por outras matrizes, conforme a derivação utilizada (KLETTE, 2014). Um resultado comparativo pode ser observado na Figura 4, onde o operador Laplaciano foi aplicado na mesma imagem do veículo, à esquerda na Figura 3.

Figura 4 – Laplaciano



Fonte: próprio autor

### 2.3.1.4. Haar Wavelet Discreto

A DFT (vide Apêndice A.2.1 Transformada Discreta de Fourier) é capaz de identificar as frequências que compõem uma imagem, mas sem explicitar sua posição. Para que se possa trabalhar com frequência e posição, pode-se utilizar Transformadas Discretas de Haar Wavelet (Haar DWT – *Discrete Wavelet Transform*).

Analogamente à DFT bidimensional, uma Haar DWT bidimensional calcula a representação de uma dada Imagem  $I$  em termos de funções bidimensionais discretas de formas de onda com diferentes comprimentos de onda. Os coeficientes gerados pela Haar DWT são puramente reais, mantendo-se as imagens geradas no domínio espacial (KLETTE, 2014).

Seja uma *wavelet* discreta  $\psi = [W_1, W_2, B]$ , onde  $W_1$  e  $W_2$  são duas regiões de cor branca e  $B$  uma região de cor preta, definida com um ponto de referência, conforme ilustrado na Figura 5.a. Ao manter a imagem  $I$  fixada e transportar a *wavelet* discreta para o ponto  $p(x, y)$ , temos a *wavelet*  $\psi_p$  em torno daquele ponto, pode-se descrever seu valor naquele ponto conforme visto na Equação (2-10) (KLETTE, 2014):

$$V(\psi_p) = \omega_1 \cdot S_{W_1} + \omega_2 \cdot S_{W_2} - \omega_3 \cdot S_B \quad (2-10)$$

onde:

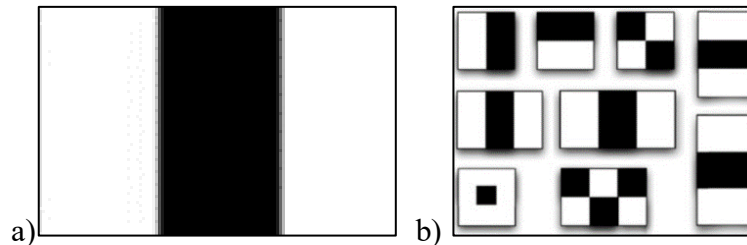
$\omega_i$  são pesos de proporcionalidade para cada região  $W_i$ .

Tal que  $S_{W_{1,2}}$  e  $S_B$  podem ser definidos conforme Equações (2-11) e (2-12) (KLETTE, 2014):

$$S_{W_i} = \sum_{p \in W_i} I(p) \quad (2-11)$$

$$S_B = \sum_{p \in B} I(p) \quad (2-12)$$

Figura 5 – Wavelet Bidimensional Discreta



Fonte: Adaptado de (KLETTE, 2014)

Dessa forma, ao se aplicarem diversas *wavelets* discretas de diversos comprimentos de onda (vide Figura 5.b) em torno dos pontos da imagem, obtém-se uma decomposição da imagem composta das quatro sub-bandas LL, HL, LH, HH (L – *low* ou baixa e H – *high* ou alta).

Figura 6 – Resultado da Haar DWT



Fonte: (PANG, 2014)

Decomposta a imagem nas quatro sub-bandas, se extraem as linhas das bordas verticais (HL) e horizontais (LH) do veículo, como se pode verificar na Figura 6. Suas vantagens são a velocidade e menor tamanho de saída da imagem, quando comparada com os resultados dos operadores Sobel e Laplaciano (PANG, 2014).

### 2.3.2. Morfologia Matemática

A morfologia matemática é utilizada no tratamento das formas encontradas na imagem binária ou em escala de cinza. A morfologia é basicamente composta pelas operações de dilatação e erosão. Com suas combinações, geram-se as operações de Fechamento, com dilatação seguida de erosão, expressa na Equação (2-13) e de Abertura, com erosão seguida de dilatação, expressa na Equação (2-14) (MAKAOUI; GUENNOUN; GHOGHO, 2016).

Para tanto, nas operações de erosão e dilatação, é feita a varredura da imagem original com um elemento estruturante que pode ser circular, retangular, diamante, etc.

$$f * e(x, y) = [(f \oplus e) \ominus e](x, y) \quad (2-13)$$

$$f \circ e(x, y) = [(f \ominus e) \oplus e](x, y) \quad (2-14)$$

onde:

- $\oplus$  é o operador dilatação
- $\ominus$  é o operador erosão.
- $e(x,y)$  é um elemento estruturante.

A Figura 7 ilustra, uma imagem original e os resultados da dilatação utilizando como elemento estruturante círculos de raios 5px, 10px e 15px, respectivamente.

Figura 7 – Dilatação



Fonte: (MAKAOUI; GUENNOUN; GHOGHO, 2016)

A Figura 8 ilustra, uma imagem original e os resultados da erosão utilizando como elemento estruturante círculos de raios 5px, 10px e 15px, respectivamente.

Figura 8 – Erosão

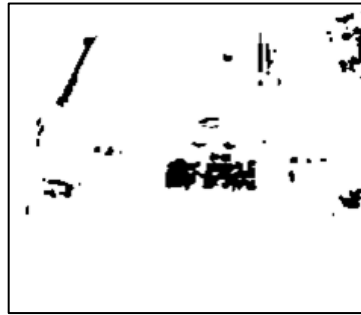


Fonte: (MAKAOUI; GUENNOUN; GHOGHO, 2016)

As técnicas de morfologia para localização da placa em uma imagem se baseiam no fato de que as variações do brilho dentro da região da placa são maiores que em outras regiões da imagem (ATIWADKAR et al., 2015). Esse número de variações abruptas da intensidade dos *pixels* dentro de uma região delimitada da imagem pode ser denominado como **Variância** (PANG, 2014). Uma alta variância representa muitas mudanças abruptas na intensidade dos *pixels* numa região pequena, por exemplo.

Após a extração das bordas da imagem, conforme discutido na Seção 2.3.1, agrupando-se as bordas que são próximas entre si dentro dessas regiões de alta variância, com um operador de dilatação, por exemplo, criam-se áreas dentro da imagem e o padrão com maior área é selecionado como provável área de interesse (FARAJIAN; RAHIMI, 2014). Um resultado está exemplificado na Figura 9. Nota-se claramente a região central (placa) com área maior do que os outros segmentos dispersos pela imagem, como a lateral e faróis do veículo.

Figura 9 – Morfologia



Fonte: Adaptado de (ATIWADKAR et al., 2015)

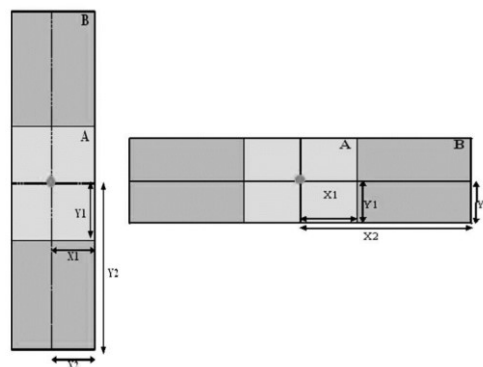
### 2.3.3. Técnicas baseadas em Histograma

O histograma (vide Apêndice A.1.2 Histograma) de uma imagem representa a taxa de ocorrência de cada uma das intensidades de cinza (ATIWADKAR et al., 2015). A principal forma de LPD baseada em histograma é a técnica de Janelas Concêntricas.

#### 2.3.3.1. Janelas concêntricas

Janelas concêntricas (SWC – *Sliding Concentric Windows*) é um método em que se criam duas janelas concêntricas que varrem a imagem, calcula-se o desvio padrão de cada pixel na direção de cada uma das janelas e, se a variação é abrupta dentro de alguma delas, define-se o ponto como sendo de borda vertical ou horizontal.

Figura 10 – Janelas Concêntricas



Fonte: Adaptado de (ATIWADKAR et al., 2015)

Verifica-se então a conectividade dessas regiões verticais e horizontais geradas e, na sequência, se são regiões de interesse (FARAJIAN; RAHIMI, 2014). É possível verificar um esquema das janelas utilizadas na Figura 10.

### 2.3.4. Aprendizado de Máquina

Sistemas de inteligência de máquina usam um paradigma diferente daquele de processamento/memória clássico. Os sistemas mais utilizados na literatura para sistemas de identificação de placas veiculares são as Redes Neurais Artificiais (FARAJIAN; RAHIMI, 2014).

## 2.4. Segmentação e Reconhecimento

Em geral, na etapa de Segmentação e Reconhecimento, a região de interesse contendo a placa passa pelos mesmos tratamentos de pré-processamento que a etapa de Localização, a saber, Conversão, Binarização e Filtragem (vide Seção 2.2.1). Em particular, na etapa de binarização, é primordial que os segmentos que são caracteres fiquem perfeitamente separados de forma que, na segmentação, seja possível extraí-los individualmente para reconhecimento.

### 2.4.1. Segmentação de caracteres

A segmentação de caracteres consiste em separar os caracteres de forma que possa ser executada sua identificação de forma individual. A técnica mais difundida na literatura consiste no uso da Projeção Horizontal e Vertical. Outras técnicas utilizadas são aquelas baseadas em morfologia (Transformada *Top-Hat*).

#### 2.4.1.1. Projeção Horizontal e Vertical

A técnica mais utilizada para segmentação de caracteres na ROI é a projeção nas direções vertical e horizontal. Um resultado desse processo pode ser verificado na Figura 11.

Figura 11 – Projeção Horizontal e Vertical



Fonte: (BELVISI et al., 1999)

A técnica consiste em varrer as colunas e linhas e identificar as regiões de maior densidade de pontos na imagem binária.

Após a segmentação, são aplicadas técnicas de morfologia e quantização para excluir segmentos de tamanhos muito pequenos, que não representam caracteres.

Além disso, são contabilizados os segmentos e são descartadas as regiões de interesse que contenham um número de segmentos muito maior ou muito menor do que o número típico de segmentos encontrado em uma placa veicular (GILLY; RAIMOND, 2013).

#### 2.4.1.2. Transformada Top-Hat

A transformada *Top-Hat* é composta de duas outras operações (MAKAOUI; GUENNOUN; GHOGHO, 2016): Erosão e Dilatação (vide Seção 2.3.2). Na Erosão diminui-se o tamanho de regiões conectadas, permanecendo apenas a estrutura geométrica, eliminando-se ainda pequenos detalhes. Na Dilatação é feita a operação inversa.

Há dois tipos de Transformadas *Top-Hat*: *Black Top-Hat* e *White Top-Hat*, abreviadas por THB e THW, respectivamente (MAKAOUI; GUENNOUN; GHOGHO, 2016).

Seja  $f(x, y)$  uma imagem em escala de cinza e  $e(i, j)$  um elemento estrutural (círculo, quadrado, retângulo etc.). Podem-se escrever as Equações (2-15) e (2-16) (MAKAOUI; GUENNOUN; GHOGHO, 2016):

$$THB(f) = (f * e - f)(x, y) \quad (2-15)$$

$$THW(f) = (f - f \circ e)(x, y) \quad (2-16)$$

onde:

$f * e$  é o Fechamento da Equação (2-13)

$f \circ e$  é a Abertura da Equação (2-14)

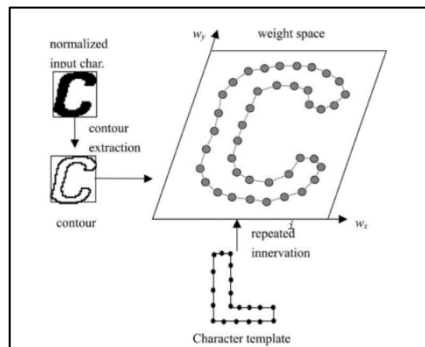
## 2.5. Reconhecimento dos Caracteres

A etapa de reconhecimento dos caracteres é onde de fato se identifica o texto contido na placa veicular. O Reconhecimento Óptico de Caracteres (OCR – *Optical Character Recognition*) é a técnica clássica implementada em sistemas de identificação de caracteres como, por exemplo, reconhecimento de texto em documentos digitalizados.

Algumas técnicas de OCR utilizam Morfologia (vide Seção 2.3.2) ao processar imagens binárias (ATIWADKAR et al., 2015).

A técnica de *Template Matching* consiste em comparar cada segmento identificado com *templates* de uma fonte já conhecida, obtendo-se a similaridade por alguma métrica definida (ATIWADKAR et al., 2015). Em alguns modelos isso é feito com a extração do contorno do segmento, conforme visto na Figura 12.

Figura 12 – Template Matching



Fonte: (ATIWADKAR et al., 2015)

Além destas técnicas, veem-se técnicas de aprendizado de máquina sendo utilizadas na literatura para a etapa de reconhecimento de caracteres (SONAVANE; SONI; MAJHI, 2015).



### 3. METODOLOGIA E IMPLEMENTAÇÃO

A metodologia proposta para o presente trabalho consistiu em uma revisão de literatura das técnicas já implementadas, que foram discutidas no Capítulo 2, a seguir foram feitos testes e a seleção de uma técnica de localização, uma de segmentação e uma de reconhecimento para implementação dos respectivos algoritmos em ambiente de simulação. Passou-se ao levantamento e seleção de banco de dados e, por fim, ao teste dos algoritmos nas imagens.

Os recursos básicos utilizados para a execução deste trabalho estão identificados a seguir:

- Material Bibliográfico.
  - Fonte: BC/UFES – acesso ao acervo IEEE, livros e outras revistas e periódicos especializados.
- Computador com o software MATLAB.
  - Fonte: Laboratórios LCEE/UFES e CISNE/UFES.
- Banco de Dados de imagens estáticas de veículos no trânsito e estacionados.
  - Veículos Brasileiros: próprio autor;
  - Veículos Estadunidenses: *UCSD/Calit2 Car License Plate, Make and Model Database* (DLAGNEKOV; BELONGIE, 2005).

O Projeto foi executado em quatro etapas separadas, de forma a poderem-se discutir os resultados parciais, dificuldades e possibilidades de melhorias em trabalhos futuros de maneira independente. As quatro etapas são, respectivamente: Aquisição, Localização, Segmentação e Reconhecimento. Cada uma delas é discutida a seguir em detalhes.

#### 3.1. Aquisição

##### 3.1.1. Banco de Dados de Placas Brasileiras

Este banco de dados foi criado pelo próprio autor, consistindo em imagens estáticas de veículos no trânsito e estacionados, fotografados na cidade de Vitória-ES, Brasil. Há imagens de veículos com diversos ângulos, tipos de iluminação, ruído, foco, etc. O Banco de Dados é composto de um *set* com 310 imagens RGB de 24-bits, em formato JPG, com dimensões de 1920 x 1080 *pixels*, ou 1440 x 1080 *pixels*, de acordo com a razão (*aspect ratio*) em que foram capturas, isto é 16:9 ou 4:3.

A fim de capturar as diversas possibilidades de variações em uma imagem digital de um veículo, foram produzidas fotografias nos períodos matutino, vespertino e noturno. Há

imagens com maior ou menor grau de ruído. Há imagens próximas e distantes dos veículos, frontais e traseiras.

Após a aquisição, as 310 imagens obtidas foram agrupadas para formar o banco de dados brasileiro, os números das placas foram digitados em um arquivo tipo CSV, que foi juntado à Base de Dados. Além dos caracteres de cada placa, foi definido manualmente a região de cada placa e salvo em arquivo tipo MAT.

Esses dois procedimentos são importantes para automatização do algoritmo e verificação dos resultados no Banco de Dados. A Figura 13 ilustra algumas imagens cujas placas foram recortadas manualmente.

Figura 13 – Posição das Placas Veiculares - Manual



Fonte: próprio autor

### 3.1.2. UCSD/Calit2 Car License Plate, Make and Model Database

Foi utilizado, ainda, para fins de comparação do algoritmo de Localização das Placas Veiculares, o Banco de Dados *UCSD/Calit2 Car License Plate, Make and Model Database* (DLAGNEKOV; BELONGIE, 2005). Os autores gentilmente permitiram sua utilização neste trabalho.

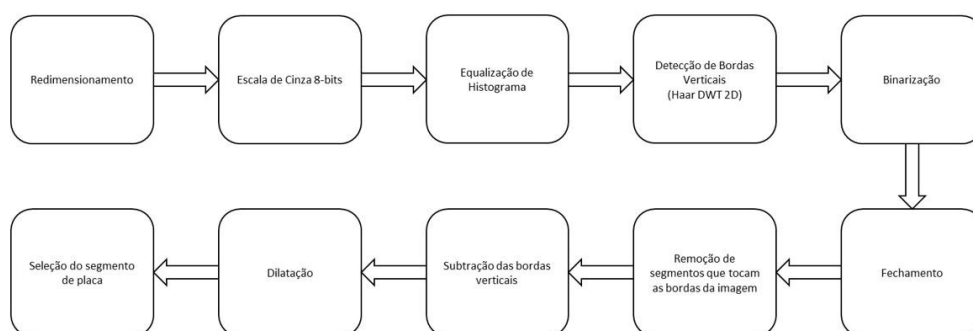
O Banco de Dados consiste em imagens estáticas de veículos no trânsito e estacionados, extraídos de uma série de vídeos feitos pelos autores Dlagnekov e Belongie, na *University of California, San Diego*. As imagens possuem formato RGB 24-bits, extensão JPG e dimensões 640 x 480 *pixels*. As imagens estão divididas em 3 *sets*, nominalmente: *Gilman*, *Regents* e *Stills*.

O *set Gilman* é composto por 1.290 imagens traseiras de veículos no trânsito, dentre as quais 18 não possuem placas visíveis. O *set Regents* é composto de 114 imagens recortadas da traseira de veículos no trânsito e o *set Stills* é composto de 291 imagens de veículos estacionados.

### 3.2. Localização

Para implementação da Localização de Placas Veiculares (VLPL ou LPL), foram utilizadas as técnicas de detecção de bordas, binarização, morfologia matemática e propriedades geométricas, conforme fluxograma da Figura 14.

Figura 14 – Fluxograma da Localização



Fonte: próprio autor

A função desenvolvida em ambiente MATLAB, denominada ‘*crop\_plate.m*’ (vide Apêndice B.1 Localização), tem a função de, através da entrada de uma imagem de veículo, identificar a localização aproximada de uma placa veicular, retornando um retângulo em forma de vetor 1x4, composto pelas coordenadas [x, y] do *pixel* correspondente ao canto superior esquerdo, seguido da largura w e altura h, isto é, [x y w h]. A seguir, discute-se em detalhes as etapas da função.

#### 3.2.1. Leitura, Redimensionamento e Conversão

A primeira etapa da localização consiste na leitura do arquivo do Banco de Dados e redimensionamento. A altura da imagem foi fixada em 540 *pixels*, sendo a largura redimensionada proporcionalmente.

De forma a se obterem menores tempos de processamento, optou-se por tratar as imagens em escala de cinza de 8-bits. A conversão entre RGB e escala de cinza utilizada foi aquela discutida em 2.2.1.1. A Equação (3-1) apresenta os pesos utilizados, que aproximam a percepção humana de luminosidade.

$$C = 0,2989R + 0,5870G + 0,1140B \quad (3-1)$$

onde:

C, R, G e B representam os valores de cinza, vermelho, verde e azul, respectivamente.

Na Figura 15 pode-se verificar a imagem RGB lida e o resultado após a conversão para escala de cinza.

Figura 15 – Imagem RGB lida e escala de cinza



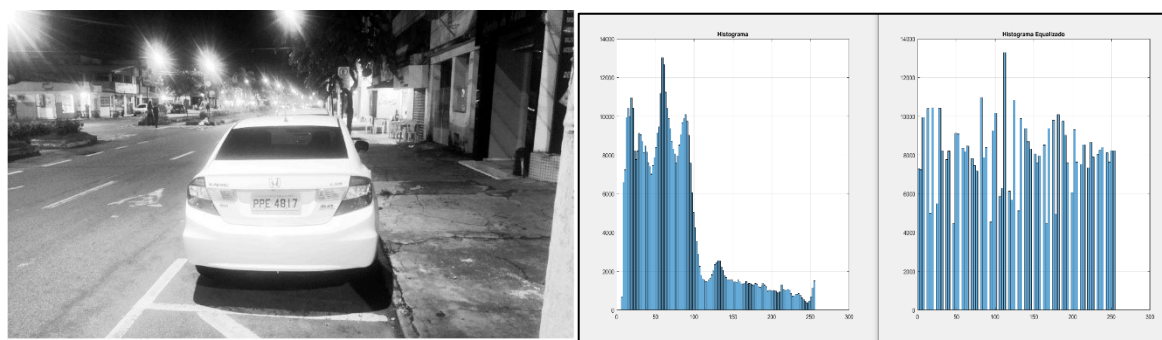
Fonte: próprio autor

### 3.2.2. Processamento inicial

De modo a produzirem-se melhores resultados, foi adotada a técnica de Equalização de Histograma, para aumentar o contraste dos elementos na imagem, ressaltando-se, inclusive, os caracteres presentes na placa.

Conforme Klette (2014), a equalização de histograma consiste em recalculer o valor da intensidade de cada pixel na imagem de forma a obter-se uma distribuição aproximadamente uniforme de todas as intensidades de cinza. Na Figura 16 ilustra-se o histograma da imagem lida antes e após a equalização.

Figura 16 – Histograma Equalizado



Fonte: próprio autor

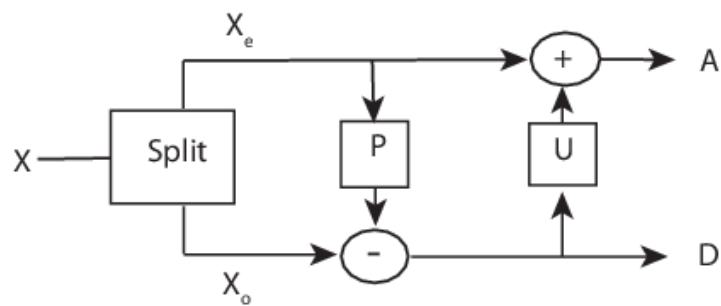
### 3.2.3. Detecção de Bordas Verticais

De forma a identificar a posição aproximada da placa na imagem, utilizou-se a propriedade de que a variância (vide Seção 2.3.2) nesta região é muito maior em relação ao restante da imagem (PANG, 2014). Isso nem sempre é verdade, uma vez que regiões que contenham grama, folhas e outros elementos com padrões de variação abrupta de brilho possuem também

uma grande variância de intensidade dos *pixels*, dificultando a precisão. De forma a isolar essa região de alta variância, foi utilizado o Haar Wavelet Bidimensional.

A implementação da Haar DWT no MATLAB é feita utilizando-se o método de *lifting* para construção de *wavelets*. Esse *design* é denominado *wavelets* de segunda geração (MATHWORKS, 2017). O sinal de entrada é **dividido** (*Split*) em componentes polifásicos a saber amostras pares ( $X_e$ ) e ímpares ( $X_o$ ). É feita uma **predição** (*Predict* ‘P’) do componente ímpar baseado em uma combinação linear dos componentes pares, e por fim uma **atualização** (*Update* ‘U’) dos componentes pares pela diferença das amostras obtidas na etapa de predição. O processo é ilustrado na Figura 17. Os vetores A e D são o resultado de uma etapa de *dual lifting* e são utilizadas para definir as *wavelets*.

Figura 17 – *Lifting* – *Split/Predict/Update*



Fonte: (MATHWORKS, 2017)

Como os caracteres na região da placa possuem muitas bordas verticais próximas, há uma janela de alta variância em torno da placa (PANG, 2014). O resultado da Haar DWT utilizada está ilustrado na Figura 18. Nota-se que fica destacada a região da placa, bem como as regiões com árvores/galhos/grama. A técnica Haar DWT foi escolhida em detrimento das outras discutidas na seção 2.3.1, porque apresentou menor tempo de processamento e detectou as bordas verticais, na maioria das imagens, com mais precisão que as outras testadas.

Figura 18 – Bordas Verticais – Haar Wavelet



Fonte: próprio autor

A sub-banda HL (*high-low* vide Seção 2.3.1.4) representa as bordas verticais da imagem. O Haar DWT retorna essa sub-banda com metade do tamanho da imagem original, por isso, esta é redimensionada de volta ao tamanho inicial.

#### 3.2.4. Binarização

Após a detecção das bordas verticais, é executado um processo de binarização, de forma a salientar as bordas detectadas, utilizando-se a binarização adaptativa. Isso foi importante porque muitas imagens do banco de dados possuem sombras na região da placa, o que causa artefatos indesejados no método de binarização de Otsu, devido a seu limiar global (OTSU, 1979).

No método adaptativo, um limiar é computado para cada região da imagem, de forma que os efeitos de sombreamento são reduzidos. Em particular, no método adaptativo utilizado, denominado Método de Bradley (BRADLEY; ROTH, 2007), a comparação é feita a nível de pixel, sendo cada pixel comparado com a média dentro de uma janela correspondente a 1/8 do tamanho da imagem. O resultado desse processo é ilustrado na Figura 19.

Figura 19 – Imagem Binarizada

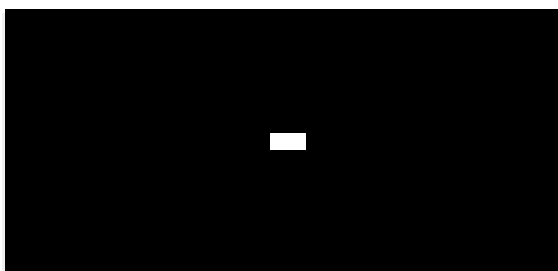


Fonte: próprio autor

#### 3.2.5. Morfologia Matemática

Após a extração das bordas verticais e após ressaltar as bordas pela binarização, a imagem passa pelo processo de morfologia, com uma dilatação seguida de erosão, utilizando-se como Elemento Estruturante um retângulo, ilustrado na Figura 20, com a mesma razão entre largura e altura das placas brasileiras, isto é 3, com dimensões de 4 x 12 *pixels*, que seja capaz apenas de conectar as linhas verticais dos caracteres da placa durante a dilatação, mas sem conectar a região da placa com outras partes próximas, como as laterais ou faróis, etc.

Figura 20 – Elemento Estruturante - Morfologia



Fonte: próprio autor

Dessa maneira, cria-se uma região bem definida na área dos caracteres que compõem a placa, e criam-se outras regiões grandes, que são a conexão de todos os elementos de fundo que podem ser considerados ruído para a aplicação aqui pretendida. Isso é importante, pois esses blocos grandes ou com formatos muito irregulares irão ser descartados a seguir. O resultado desse processo de Morfologia, denominado “Fechamento”, isto é, dilatação seguido de erosão (MAKAOUI; GUENNOUN; GHOGHO, 2016), está ilustrado na Figura 21.

Figura 21 – Morfologia - Fechamento



Fonte: próprio autor

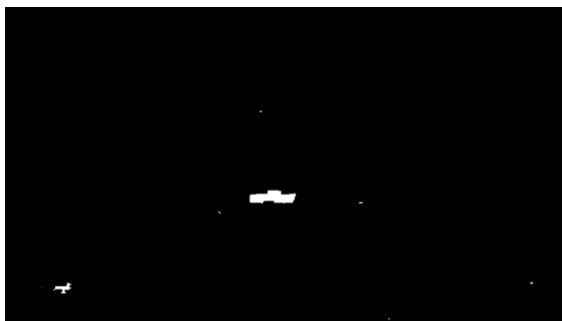
### 3.2.6. Elementos de Borda e Preenchimento

O próximo passo para eliminar os segmentos que não representam uma placa é excluir as regiões conectadas que tocam a borda da imagem. Em boa parte do *set* isso faz com que apenas a região da placa fique isolada, entretanto, se houver adesivos, modelo do veículo e outros elementos textuais, a metodologia consegue identificar tais regiões, que também se tornam segmentos. A questão da seleção de regiões candidatas será discutida na Seção 3.2.7.

Após a eliminação dos segmentos que tocam a borda da imagem, é feita a subtração das bordas verticais encontradas anteriormente (Figura 19). Isso faz com que a região da placa seja desconectada das regiões de fundo, nos casos em que isso possa ter ocorrido no processo de fechamento. Após a subtração é executada uma nova dilatação com um elemento retangular de  $2 \times 8$  *pixels* e, por fim é feito o preenchimento dos “buracos” dentro dos

segmentos restantes. Dessa maneira, a região da placa ficará mais sólida. A imagem abaixo ilustra o resultado após remoção dos elementos de borda e preenchimento.

Figura 22 – Imagem após remoção dos elementos de borda e preenchimento



Fonte: próprio autor

### 3.2.7. Segmentação e Seleção de Regiões Candidatas

Com os segmentos restantes após o processamento da imagem, resta definir qual deles representa uma placa veicular. Para tanto, todos os segmentos são etiquetados e, na sequência suas propriedades são extraídas.

No MATLAB esse processo é feito através das técnicas denominadas *run-length encoding* e *local table method*, utilizando-se o algoritmo proposto por Shapiro e Haralick (1992, p. 40-48). A técnica consiste em varrer as linhas de uma imagem binária gravando em uma tabela as coordenadas do pixel inicial e final dos segmentos de valor unitário conectados. Após varrer todas as linhas, são atribuídas etiquetas preliminares e gravadas as equivalências entre as etiquetas numa tabela local. As equivalências de classes são verificadas e as regiões 8-conectadas são atualizadas com a etiqueta de menor valor. Em outras palavras, são identificadas e enumeradas as regiões 8-conectadas que foram destacadas pela binarização da imagem. As propriedades de cada segmento são, então, calculadas. Na Figura 23, cada linha representa as propriedades de um dos segmentos da Figura 22.

Figura 23 – Propriedades dos segmentos

Fields	Area	Eccentricity	Solidity	Centroid	BoundingBox
1	1	0	1	[62 477]	[61.5000 476.5000 1 1]
2	207	0.8751	0.5656	[100.7295 477.5652]	[83.5000 468.5000 30 18]
3	8	0.9621	1	[364.5000 350]	[362.5000 347.5000 4 5]
4	1200	0.9703	0.8163	[454.2833 324.6625]	[415.5000 312.5000 78 22]
5	7	0.8562	1	[434.2857 179.4286]	[432.5000 178.5000 4 2]
6	12	0.9428	1	[603.5000 333.5000]	[600.5000 332.5000 6 2]
7	4	0	1	[651.5000 529.5000]	[650.5000 528.5000 2 2]
8	12	0.6614	1	[892.5000 469]	[890.5000 467.5000 4 3]
9					

Fonte: próprio autor

A área representa o somatório de *pixels* 1 no segmento. A Excentricidade representa o quão circular o segmento é, numa escala de 0 a 1, sendo 0 totalmente circular e 1 um segmento de



linha. A Solidez representa quão sólida é a região, isto é, quão preenchida está a região retangular que contém o segmento.

Uma placa veicular deverá satisfazer alguns critérios, verificados nos testes feitos, a saber:

- Em geral, o segmento de placa possuirá uma área grande após o processamento da imagem, em muitos casos a maior área;
- A excentricidade da região deverá ser próxima de 1,00. Nesta aplicação foi utilizado  $E > 0,8$ ;
- A região deverá ser aproximadamente sólida. Isso pode variar muito entre diferentes placas segmentadas, foi utilizado  $S > 0,5$  para o caso geral aqui;
- A razão entre largura e altura deverá ser maior ou igual a 2,5 e menor que 10.

O segmento escolhido como placa é aquele dentre os que satisfazem as propriedades elencadas e que possui a maior área. Na Figura 23 está selecionada a região que representa a placa para a imagem de exemplo utilizada. Os parâmetros estão compilados na Tabela 1.

Tabela 1 – Parâmetros de seleção da região de interesse

<b>Excentricidade</b>	$> 0,8$
<b>Solidez</b>	$> 0,5$
<b>Área Mínima</b>	500 <i>pixels</i>
<b>Razão</b>	$2,5 \leq R < 10$

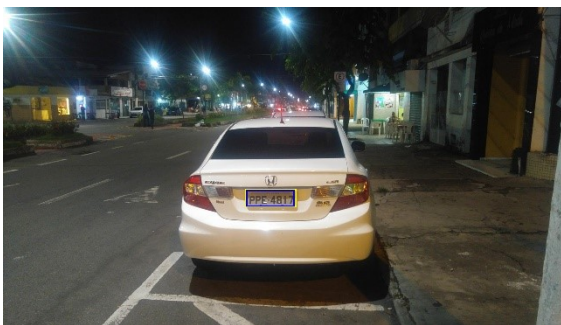
Fonte: próprio autor

O último passo na localização é ajustar o retângulo ao redor do segmento escolhido. Isso foi implementado mantendo-se a largura e ajustando a altura para satisfazer a razão de 3. Além disso, o retângulo é redimensionado para as dimensões da imagem original, que havia sido reduzida, conforme Equação 3.2.1.

Para verificar se a região selecionada é a correta, é feita a verificação da sobreposição<sup>1</sup> ou *overlap* (detalhes na Seção 3.4.1.2) entre as duas *bounding boxes* vistas na Figura 24, onde está ilustrado o resultado da Localização da placa na imagem (retângulo em azul), comparando-se com o recorte feito à mão (retângulo em amarelo). Se a sobreposição é maior que 60%, considera-se como uma identificação correta.

<sup>1</sup> No MATLAB isto é implementado com a função *bboxOverlapRatio(bboxA, bboxB)*.

Figura 24 – Resultado da Localização automática



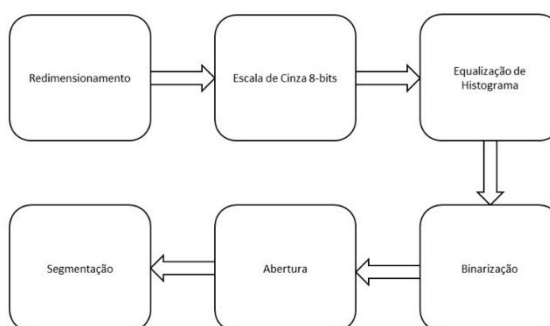
Fonte: próprio autor

Nota: Em amarelo o recorte manual e em azul o automático.

### 3.3. Segmentação

Para implementação da Segmentação dos caracteres das Placas Veiculares, foram utilizadas as técnicas de binarização, morfologia matemática, segmentação e propriedades geométricas, conforme fluxograma da Figura 25.

Figura 25 – Fluxograma da Segmentação



Fonte: próprio autor

A função desenvolvida em ambiente MATLAB denominada *'segment\_characters.m'* (vide Apêndice B.2 Segmentação), tem a função de, dada uma imagem de placa veicular já recortada da imagem original através da técnica apresentada na Seção 3.2, segmentar os caracteres componentes daquela, de forma a possibilitar a posterior identificação de qual letra ou número se trata. A seguir é apresentado o método implementado para segmentação de caracteres. O processo é, em alguns pontos análogo àquele utilizado para localizar a placa, possuindo, entretanto, muitas particularidades.

#### 3.3.1. Entrada, Redimensionamento e Conversão

A entrada do algoritmo de segmentação de caracteres é o recorte feito no arquivo da região onde se encontra a placa veicular. A imagem recortada é, então, redimensionada para maior

velocidade na execução do algoritmo, havendo naturalmente um *trade-off* entre velocidade de execução e qualidade da segmentação.

Um bom resultado intermediário foi conseguido fixando a largura da placa em 300 *pixels*, sendo a altura redimensionada proporcionalmente. A placa terá, portanto, um tamanho de aproximadamente 300 x100 *pixels*, dada a razão entre largura e altura do padrão brasileiro.

Análogo à Localização, de forma a se obterem menores tempos de processamento, optou-se por tratar as imagens em escala de cinza de 8-bits. A conversão entre RGB e escala de cinza utilizada foi a mesma, que está expressa na Equação (3-1).

Na Figura 26 pode-se verificar a placa em RGB e o resultado após a conversão para escala de cinza.

Figura 26 – Placa RGB e escala de cinza



Fonte: próprio autor

### 3.3.2. Processamento inicial

O primeiro passo do processamento de imagem para segmentação da placa é a Equalização de Histograma, conforme discutido na Seção 3.2.2. Com o aumento de contraste proporcionado, há uma melhor separação entre os caracteres e o fundo, melhorando o resultado da binarização, discutida a seguir.

O resultado da equalização do histograma está ilustrado na Figura 27. Nota-se também que o aumento de contraste ressalta o sombreamento da placa.

Figura 27 – Histograma Equalizado - Placa



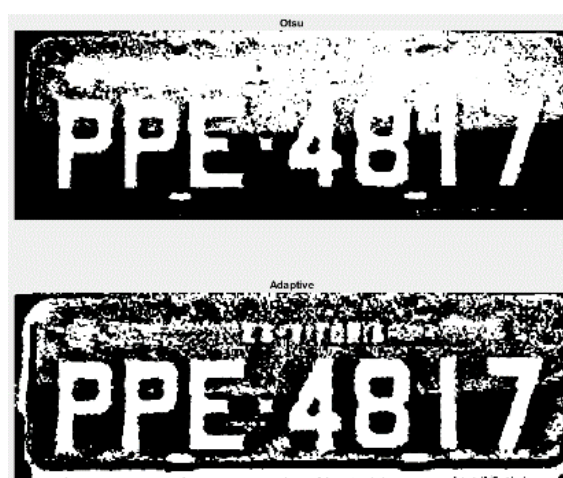
Fonte: próprio autor

### 3.3.3. Binarização

Após o ajuste no contraste, passa-se ao processo de binarização. Analogamente ao método de binarização utilizado para localização, na segmentação foi utilizado o método adaptativo. As sombras na região da placa que causam artefatos indesejados no método de binarização de Otsu (OTSU, 1979), podem ser verificados na Figura 28, na parte superior.

Comparativamente, no método adaptativo (BRADLEY; ROTH, 2007), os efeitos de sombreamento são reduzidos, conforme discutido na Seção 3.2.4. O resultado desse processo é ilustrado também na Figura 28, na parte inferior.

Figura 28 – Binarização de Otsu e Adaptativa



Fonte: próprio autor

### 3.3.4. Morfologia Matemática

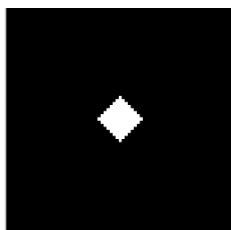
Após o processo de binarização, os caracteres ficam salientados, entretanto, na maioria dos casos há ruído e elementos menores que conflitam com as regiões de interesse. Dessa maneira, de forma a eliminar pequenos segmentos e, uma vez que os caracteres devem ser as regiões com maior área, é feito um tratamento com Morfologia Matemática, do tipo Abertura, que consiste em uma erosão seguida de dilatação (MAKAOUI; GUENNOUN; GHOGHO, 2016).

Diferente da Localização em que se utilizou um Fechamento, aqui a intenção é isolar o máximo possível os caracteres entre si e dos artefatos que, porventura ficaram a estes conectados no processo de binarização.

O Elemento Estruturante utilizado nesta abordagem foi um losango com largura de 14 *pixels*, ou seja,  $d = 7$  *pixels* (sendo  $d$  a distância entre o centro e a ponta do elemento), que está ilustrado na Figura 29.

Esse tamanho definido teve resultados interessantes para a maior parte das placas, uma vez que com as placas recortadas, apesar de haver rotações e ruído, o tamanho dos caracteres será aproximadamente o mesmo, dado o redimensionamento prévio feito.

Figura 29 – Elemento Estruturante - Placas



Fonte: próprio autor

O resultado da Abertura feita na imagem está ilustrado na Figura 30. Nota-se que os caracteres ficam, de fato, destacados e possuem as maiores áreas. Entretanto, dependendo do tipo de recorte da placa, podem restar outros segmentos grandes, como o segmento alongado na parte inferior da imagem. A seleção das regiões candidatas a ‘caractere’ será discutido a seguir.

Figura 30 – Morfologia - Abertura



Fonte: próprio autor

### 3.3.5. Segmentação e Seleção dos Caracteres

Dentre os segmentos restantes após o processamento da imagem, resta definir quais são os sete caracteres que correspondem à placa veicular que se quer reconhecer. Aqui, novamente análogo à Localização, todos os segmentos são etiquetados, com suas propriedades extraídas para testar quais são caracteres. É feita a segmentação da imagem com a mesma técnica expressa na Seção 3.2.7, e as propriedades dos segmentos são calculadas.

Um caractere de uma placa veicular brasileira deverá possuir uma razão aproximadamente entre 0,65 e 0,85 entre largura e altura, excluída a letra 'I' e o número '1' que possuem uma razão muito menor, de 0,16 aproximadamente (CONTRAN, 2007).

De toda sorte, nota-se que na Figura 30, os segmentos que são caracteres possuem as maiores áreas dentre aqueles que possuem uma razão entre largura e altura menor ou igual a 0,85. Na Figura 31 estão as propriedades dos caracteres da placa utilizada como exemplo.

Figura 31 – Propriedades dos caracteres

Fields	Area	Centroid	BoundingBox
1	8797	[584.5561 174.7005]	[539.5000 103.5000 90 145]
2	6994	[312.3380 174.2568]	[277.5000 102.5000 91 148]
3	6942	[105.0029 167.9334]	[65.5000 107.5000 91 151]
4	6823	[209.7530 165.7988]	[171.5000 104.5000 89 155]
5	6507	[481.6462 180.1947]	[435.5000 102.5000 84 154]
6	5624	[797.8160 159.4387]	[750.5000 101.5000 89 152]
7	3509	[689.1690 175.4563]	[673.5000 100.5000 31 148]
8			

Fonte: próprio autor

Dessa maneira, de forma a incluir eventuais distorções nessa razão, foi aplicado um filtro que seleciona apenas razões menores que 1,5. Isto significa que o caractere mesmo distorcido, não deverá possuir uma largura maior que uma vez e meia a altura, o que proporcionou bons resultados, excluindo particularmente segmentos horizontais.

As propriedades utilizadas na escolha dos caracteres estão compiladas na Tabela 2. O resultado da seleção dos caracteres está ilustrado na Figura 32.

Tabela 2 – Parâmetros de seleção de caracteres

<b>Área</b>	7 maiores
<b>Razão</b>	$R \leq 1,5$

Fonte: próprio autor

Após selecionados os sete caracteres, o algoritmo retorna uma matriz contendo os mesmos para a função principal, responsável por executar, finalmente, o reconhecimento, que será tratado na sequência.

Figura 32 – Resultado da Segmentação



Fonte: próprio autor

### 3.4. Reconhecimento Óptico de Caracteres

Após a segmentação de Caracteres passa-se à etapa de reconhecimento, sendo cada um dos sete caracteres reconhecido individualmente. A função desenvolvida em ambiente MATLAB denominada *'recognize\_character\_cmp.m'* (vide Apêndice B.3 Reconhecimento de Caracteres), tem a função de receber os caracteres segmentados, reconhecê-los, concatenar as variáveis *char* e retornar uma *String* com a placa completa identificada.

As placas veiculares de identificação brasileiras seguem, atualmente, o padrão definido na Resolução número 231, de 15 de março de 2007 (CONTRAN, 2007).

Figura 33 – Padrão das Placas Brasileiras



Fonte: (CONTRAN, 2007)

Nesse padrão os três primeiros caracteres são alfabéticos em caixa alta e os quatro últimos caracteres são numéricos, como visto na Figura 33. Existe uma fonte específica utilizada, denominada *Mandatory* (CONTRAN, 2007), que está ilustrada na Figura 34.

Figura 34 – Fonte dos Caracteres Brasileiros

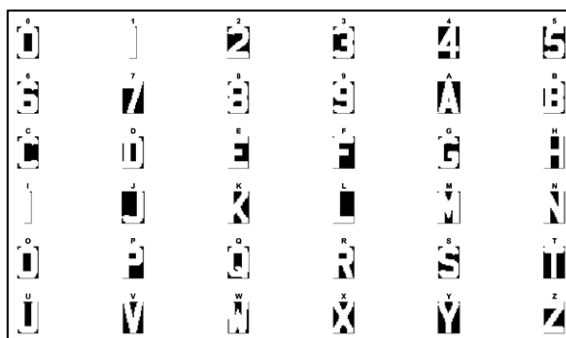


Fonte: (CONTRAN, 2007)

Como as placas possuem o padrão explicitado acima, foram abordadas duas metodologias de reconhecimento dos caracteres. A primeira delas foi o uso da técnica de comparação (*template matching*) dos caracteres com a fonte *Mandatory*, da Figura 34.

Para tanto, a fonte extraída do documento digital da Resolução 231 (CONTRAN, 2007) foi binarizada, segmentada e os caracteres foram salvos em uma matriz e indexados pelo caractere correspondente. Na Figura 35 estão os caracteres usados para comparação.

Figura 35 – Fonte Mandatory Segmentada



Fonte: próprio autor

A segunda metodologia foi o uso do *Software* Tesseract OCR<sup>2</sup>, que é um programa em linha de comandos de código aberto de reconhecimento de texto (SMITH, 2007).

Cada uma das metodologias é detalhada a seguir e uma comparação dos tempos de execução é feita no capítulo de resultados.

### 3.4.1. Template Matching

#### 3.4.1.1. Entrada e Redimensionamento

Os caracteres segmentados e binarizados são recebidos como uma matriz 7x1. Cada caractere é, então, redimensionado para uma altura de 128 *pixels*, redimensionando-se a largura proporcionalmente. Cada caractere é então reconhecido individualmente, através de sua comparação com os *templates* binarizados dos caracteres da fonte original (Figura 35).

Cada um dos três primeiros segmentos é comparado com os caracteres alfabéticos da fonte original, por sua vez, cada um dos quatro últimos segmentos é comparado com os caracteres numéricos.

#### 3.4.1.2. Similaridade

De forma a estimar o quão parecido um segmento  $A_i$  é de um caractere  $B_j$  da fonte original é calculada a razão de sobreposição entre as áreas como (SILVA; CIARELLI, 2016):

$$overlap(A_i, B_j) = \frac{area(A_i \cap B_j)}{area(A_i \cup B_j)} \quad (3-2)$$

<sup>2</sup> Disponível em [github.com/tesseract-ocr](https://github.com/tesseract-ocr).



onde:

$A_i$  é o  $i$ -ésimo segmento encontrado

$B_j$  é o  $j$ -ésimo caractere da fonte original

Nesse caso, é escolhido o caractere mais similar ao segmento, desde que satisfeita a propriedade de que o índice de similaridade obtido seja maior que 50%, caso contrário, o segmento é descartado como não sendo um caractere.

### 3.4.1.3. Conhecimentos *a priori*

De forma a tornar o algoritmo mais especializado para o caso tratado, é utilizado o conhecimento *a priori* de que as combinações de letras e números são predefinidas pelo CONTRAN para cada estado brasileiro (UNIFORCE, 2017).

Com esta informação, assumindo-se que todas as placas do banco de dados são do Espírito Santo, a etapa de reconhecimento dos caracteres leva em consideração apenas os valores que, de fato, formam combinações possíveis. Cada caractere alfabético é comparado apenas com os valores possíveis de serem assumidos em cada uma das 3 posições iniciais da placa, por exemplo, {M, O, P} para a primeira posição e assim por diante, conforme Tabela 3, onde estão todas as sequências alfanuméricas de emplacamento para o estado do Espírito Santo.

Tabela 3 – Sequências Alfanuméricas - ES

MOX 0001 a MTZ 9999	
OCV 0001 a ODT 9999	
OVE 0001 a OVF 9999	Espírito Santo (ES)
OVH 0001 a OVL 9999	
OYD 0001 a OYH 9999	
PPA 0001 a PPZ 9999	

Fonte: Adaptado de (UNIFORCE, 2017)

### 3.4.2. Tesseract OCR

O Tesseract OCR é um *software open source* de linha de comando, multi-idioma, de reconhecimento óptico de caracteres (SMITH, 2007).

É possível obtê-lo em seu repositório oficial em [github.com/tesseract-ocr](https://github.com/tesseract-ocr). A versão utilizada para testes neste trabalho foi a 3.05.01.

Análogo ao método anterior, aqui cada um dos três primeiros segmentos é comparado com os caracteres alfabéticos da fonte original, por sua vez, cada um dos quatro últimos segmentos é comparado com os caracteres numéricos.

Para se comparar cada um dos segmentos, primeiro este é salvo em disco com o nome 'char.jpg'. O Tesseract é executado via linha de comando com a seguinte sintaxe para o caso de letras:

```
!tesseract char.jpg char -c tessedit_char_whitelist=ABCDEFGHIJKLMNOPQRSTUVWXYZ -psm 10
```

Para o caso de números, a sintaxe é a seguinte:

```
!tesseract char.jpg char -c tessedit_char_whitelist=0123456789 -psm 10
```

Sendo limitado a reconhecer um caractere individualmente ('-psm 10') e limitado à faixa aceita para cada caso, isto é, letras em caixa alta ou números.

A saída do Tesseract é um arquivo de texto na mesma pasta da imagem de entrada, que é lido pelo algoritmo e concatenado à *String* da placa. Quando o Tesseract não identifica um caractere, o arquivo de retorno fica vazio.

Uma comparação das duas formas de reconhecimento utilizadas está no capítulo de resultados.

## 4. RESULTADOS E DISCUSSÃO

Foram executadas duas etapas de testes para determinarem-se os resultados da metodologia implementada. Os primeiros testes consistem no ciclo completo de reconhecimento das imagens do banco de dados brasileiro criado pelo próprio autor.

A segunda etapa de testes consistiu em executar o procedimento de localização da placa nas imagens do banco de dados *UCSD/Calit2 Car License Plate, Make and Model Database* (DLAGNEKOV; BELONGIE, 2005) para fins de comparação com os resultados encontrados na literatura.

### 4.1. Banco de Dados Brasileiro

#### 4.1.1. Localização

Primeiro foi executado o algoritmo de Localização da placa nas 310 imagens do banco de dados de veículos brasileiros. Dentre os segmentos que satisfazem as propriedades discutidas na Seção 3.2.7, foram selecionados até 5 retângulos que envolvem os primeiros 5 segmentos como possíveis regiões de interesse.

Utilizando apenas o primeiro retângulo selecionado como região de interesse, foi obtida uma taxa de identificação correta da região de interesse maior que 88% e utilizando até 5 retângulos selecionados foi obtida uma taxa maior que 90%, conforme Tabela 4.

Tabela 4 – Resultado da Localização - placas brasileiras

<b>Qtd Segmentos</b>	1	5
<b>Total</b>	310	310
<b>Acertos</b>	273	281
<b>Identificação de Placas</b>	88,06%	90,65%

Fonte: próprio autor

Utilizando a função denominada *timeit* do MATLAB, foi possível verificar que a Localização leva, em média, 100ms. Dessa forma o algoritmo seria capaz de alcançar a taxa de 10 quadros por segundo em uma aplicação implementada no próprio MATLAB.

Em geral, como exposto, o algoritmo teve bons resultados. A seguir estão expostos alguns dos resultados mais assertivos, apesar de elementos que poderiam interferir na correta detecção. Logo após, são discutidos alguns problemas encontrados durante o processo de localização.

Entre os bons resultados, podemos verificar, na Figura 36, à esquerda, que há pouco luz e a placa está muito desgastada e, à direita há muitos pontos de reflexão de luz, inclusive sobre a placa.

Figura 36 – Resultados da Localização automática (azul)



Fonte: próprio autor

Na Figura 37, à esquerda, há uma região de alta variância logo acima da placa e, à direita, a placa está inclinada e há diversos reflexos no vidro, que poderiam interferir.

Figura 37 – Resultados da Localização automática (azul)



Fonte: próprio autor

Na Figura 38, à esquerda a placa está extremamente desgastada e, à direita, há diversos reflexos no para-brisa e capô.

Figura 38 – Resultados da Localização automática (azul)



Fonte: próprio autor

Dentre os problemas que interferiram na correta detecção, pode-se notar, na Figura 39, à esquerda há pouca luz e a seleção foi menor que a placa, à direita há muita variação no para-choques, de forma que a região ficou estendida.

Figura 39 – Resultados da Localização automática (azul)



Fonte: próprio autor

Na Figura 40, à esquerda, o algoritmo selecionou uma placa em segundo plano e, à direita, a região selecionada incluiu um adesivo que estava próximo à placa.

Figura 40 – Resultados da Localização automática (azul)



Fonte: próprio autor

Na Figura 41, à esquerda, o algoritmo selecionou outro texto (marca do veículo) e, à direita, a região selecionada foi de reflexos de galhos no porta-malas.

Figura 41 – Resultados da Localização automática (azul)



Fonte: próprio autor

#### 4.1.2. Segmentação e Reconhecimento

Após executada a localização, utilizando-se dos recortes de placa definidos manualmente, foi executado o algoritmo de Segmentação e Reconhecimento de caracteres nas 310 imagens do banco de dados de veículos brasileiros.

##### 4.1.2.1. Segmentação

Para verificar a Segmentação, apenas as respostas do algoritmo que retornavam com exatos sete segmentos que satisfaziam as propriedades elencadas na Tabela 2 foram contabilizadas como placas segmentadas corretamente, sendo as outras descartadas. Por exemplo, se o processo de Segmentação de Caracteres retornasse 6 segmentos, significa que, com certeza, um caractere não foi corretamente selecionado, sendo essa placa descartada. Isso ocorre, em geral, quando um caractere possui falhas, desgastes ou elementos próximos que se conectam a ele na binarização, como furos, parafusos, etc.

Foi possível obter-se uma taxa de identificação correta dos segmentos maior que 96%, como pode ser visto na Tabela 5.

Tabela 5 – Resultado da Segmentação - placas brasileiras

<b>Total</b>	<b>2.170</b>
<b>Acertos</b>	2.100
<b>Identificação de Segmentos</b>	96,77%

Fonte: próprio autor

Utilizando a função denominada *timeit* do MATLAB, foi possível verificar que a Segmentação leva, em média, 110ms. Dessa forma o algoritmo seria capaz de alcançar uma taxa de cerca de 9 placas por segundo em uma aplicação implementada no MATLAB.

##### 4.1.2.2. Reconhecimento

Para verificar o Reconhecimento, em ambos os casos, os caracteres recortados eram enviados à função de reconhecimento e verificados um a um em suas respectivas posições originais na *String* da placa original, segundo os dados de placas constantes do banco de dados, anteriormente tabelados à mão.



### 4.1.2.3. Template Matching

Com a utilização da técnica de comparação de *Templates* foi possível obter-se uma taxa de correta classificação de caracteres individuais maior que 87%, como pode ser visto na Tabela 6. Mais de 47% de placas inteiras foram identificadas corretamente, dentre aquelas segmentadas.

Tabela 6 – Resultado do Reconhecimento – Template Matching

	<b>Caracteres</b>	<b>Placas</b>
<b>Total</b>	2.100	300
<b>Acertos</b>	1.830	141
<b>Classificação de Caracteres</b>	87,14%	47%

Fonte: próprio autor

Importante ressaltar que, com a utilização do conhecimento *a priori* (vide Seção 3.4.1.3), assumindo-se que as placas são todas do estado do Espírito Santo (o que é verdade para a maior parte do banco de dados), foi possível melhorar consideravelmente o resultado do reconhecimento de caracteres, passando-se a uma taxa de acerto de 89% e de placas inteiras, passando-se a 62%, conforme pode ser verificado na Tabela 7.

Tabela 7 – Resultado do Reconhecimento – Conhecimento *a priori*

	<b>Caracteres</b>	<b>Placas</b>
<b>Total</b>	2.100	300
<b>Acertos</b>	1.889	186
<b>Classificação de Caracteres</b>	89,95%	62,00%

Fonte: próprio autor

Utilizando, novamente, a função denominada *timeit* do MATLAB, verificou-se que o Reconhecimento por *Template Matching* leva, em média, 150ms. Dessa forma o algoritmo seria capaz de alcançar uma taxa de cerca de 7 placas por segundo em uma aplicação utilizando o próprio MATLAB.

De forma a verificar como o algoritmo de reconhecimento se comporta em relação a cada um dos 36 caracteres individualmente, foi criada a Matriz de Confusão (REVISTABW, 2015), constante à Tabela 8, onde cada coluna representa o caractere esperado e cada linha os reconhecimentos obtidos. É possível, assim, verificarem-se os erros de classificação.





Utilizando, novamente, a função denominada *timeit* do MATLAB, verificou-se que o Reconhecimento com o Tesseract OCR leva em média 2,5s.

Isso se dá, principalmente, devido ao uso do disco para fazer com que o software acesse o caractere segmentado. Dessa forma, esse tipo de metodologia não seria viável para uma aplicação que necessita resposta rápida.

Comparativamente, pôde-se verificar que a taxa de acerto da metodologia *Template Matching* foi maior que aquela do Tesseract OCR, em parte devido aos problemas deste último quanto à fonte específica. Não obstante, entende-se que a metodologia para reconhecimento de caracteres implementada teve bons resultados para a aplicação específica a que se propõem.

#### 4.1.3. Resultado da Primeira Etapa

Com o uso da metodologia implementada, executando o ciclo completo com conhecimento *a priori*, foi possível obter as taxas totais de acerto expressos na Tabela 10. Vale ressaltar que com os recortes manuais a taxa de acerto da segmentação foi maior (vide Tabela 5). A taxa de reconhecimento também diminui (vide Tabela 7), uma vez que todas as placas não segmentadas corretamente são descartadas.

Tabela 10 – Resultados do Ciclo Completo

	<b>Localização</b>	<b>Segmentação</b>	<b>Reconhecimento</b>
<b>Taxa de Acerto</b>	90,65%	73,87%	43,04%

Fonte: próprio autor

Quanto ao tempo de execução, novamente, foi utilizada a função denominada *timeit* do MATLAB, verificando-se que o ciclo completo de reconhecimento leva, em média, 300ms. Dessa forma o algoritmo seria capaz de alcançar uma taxa de cerca de 3,5 placas por segundo em uma aplicação utilizando o próprio MATLAB.

#### 4.2. UCSD/Calit2 Car License Plate, Make and Model Database

Foi utilizado o mesmo método de seleção das imagens do banco de dados aplicado por Makaoui et al. (2016), isto é, foram criados 2 grupos com as imagens no banco de dados *UCSD/Calit2 Car License Plate*. O grupo 1 consiste nas 291 imagens do *set Stills*, que são veículos estacionados. O grupo 2 é composto das 300 primeiras imagens do *set Gilman*, que são veículos no trânsito. Entretanto, dentre estas 300 imagens selecionadas, 6 de fato não possuíam placas visíveis, fazendo o total deste *set* igual a 294.

O resultado comparativo da Localização está expresso na Tabela 11. Como se verifica, utilizando a metodologia proposta para o banco de dados de placas brasileiras no banco de dados de DLAGNEKOV e BELONGIE (2005), alterando-se apenas a restrição da razão entre largura e altura para  $1,5 \leq R < 10$ , devido à diferente proporção no padrão de placas estadunidense, obtiveram-se bons resultados, da mesma ordem dos obtidos na Seção 4.1.

Tabela 11 – Resultado Localização - UCSD/Calit2 - Comparativo

	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Total</b>
<b>Total</b>	291	294	585
<b>Acertos</b>	237	233	470
<b>Taxa de Acerto</b>	81,44%	79,25%	80,34%
<b>Pang (2014)</b>	-	98%	98%
<b>Makaoui et al. (2016)</b>	98%	97%	97,5%

Fonte: próprio autor

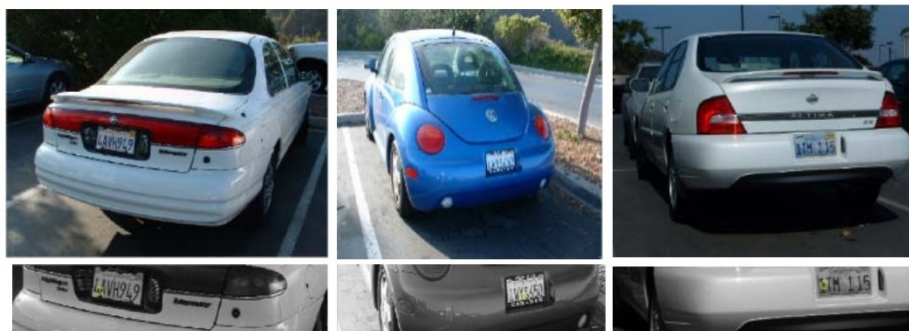
Comparando-se com os resultados da literatura, entretanto, o resultado aqui alcançado ficou aquém daqueles já obtidos por Pang (2014) e Makaoui et al. (2016), que também utilizaram o mesmo banco de dados, estando seus melhores resultados expressos também na Tabela 11. Nesse ponto, entretanto, é válido ressaltar duas características dos resultados prévios.

A primeira se refere ao trabalho de Pang (2014), que utilizou apenas o *set Gilman*, onde afirma-se ter sido obtida a taxa de acerto de 78%, contendo-se todos os caracteres da placa e 98% contendo-se parcialmente a placa, utilizando para isso 3 retângulos com o maior valor da métrica denominada *F-score*, entre os 10 primeiros pré-selecionados.

Comparativamente, dos 79,25% obtidos no presente trabalho para o *set Gilman*, cerca de 2% cobria apenas parcialmente a região da placa, com outros 77,25% cobrindo todos os caracteres, utilizando até 5 retângulos pré-selecionados.

A segunda se refere ao trabalho de Makaoui et al. (2016), que afirma ter obtido 98% e 97% de acerto nos *sets Stills* e *Gilman*, respectivamente. No presente trabalho obteve-se 81,44% e 79,25% naqueles respectivos *sets*. Entretanto, conforme verifica-se na Figura 42 e na Figura 43, comparativamente, no presente trabalho a região da placa foi mais precisamente delimitada.

Figura 42 – Resultados da Localização no trabalho de Makaoui et al. (2016)



Fonte: (MAKAOUI; GUENNOUN; GHOGHO, 2016)

Figura 43 – Resultados da Localização automática com o método proposto (em azul)



Fonte: próprio autor

Dessa forma, entende-se que, apesar de ter tido taxas de acerto menores, o presente trabalho obteve resultados satisfatórios. Não obstante, haja vista as taxas de acerto apresentadas na literatura, há margem para melhorias em implementações futuras.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como objetivo fazer a identificação de placas veiculares e o reconhecimento de seus caracteres em imagens digitais, em particular das placas no padrão brasileiro, através do uso de técnicas de processamento de imagens e de reconhecimento de caracteres. O método proposto foi testado em duas bases de dados: uma brasileira e uma estadunidense e, com base nos resultados apresentados no capítulo anterior, o objetivo geral deste trabalho foi alcançado satisfatoriamente.

Quanto aos objetivos específicos, foi alcançado adequadamente o objetivo de desenvolver a simulação computacional, com sub-rotinas capazes de: identificar a posição de uma placa veicular na imagem; segmentar os caracteres para identificação individual; e extrair os caracteres da imagem da placa. O tempo de resposta das sub-rotinas foi da ordem esperada para o propósito a que se destina a aplicação de detecção de placas. O método se mostrou capaz de fornecer resposta da ordem de milissegundos quando executado no MATLAB. Em aplicações práticas, a metodologia deverá ser implementada em *hardware* dedicado com otimização do *software*.

No banco de dados de imagens estáticas estadunidense, utilizado como base de comparação do resultado da etapa de Localização, foram obtidos resultados alinhados à expectativa. A comparação dos resultados com aqueles já obtidos na literatura possibilitou verificar que foram de ordem de grandeza próximos e que, apesar de menores, foram mais precisos, conforme discutido no capítulo anterior. Entretanto, dados os resultados da literatura, existe possibilidade de aprofundamento e melhoria do algoritmo para aumento das taxas de acerto.

### 5.1. Trabalhos Futuros

O objetivo de longo prazo deste sistema de identificação de placas veiculares brasileiras é ser embarcado junto a uma câmera de videomonitoramento para as aplicações descritas no item 1.3. Como trabalhos futuros podem ser elencados:

- Uso de técnicas de aprendizado de máquina, como arquiteturas LeNet-5 (LECUN et al., 1998) e YOLO (REDMON; FARHADI, 2016);
- Estudo e *design* de detecção de placas de motocicletas;
- Adequação do algoritmo para entrada de vídeo digital;
- Implementação e teste em hardware embarcado dedicado;

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ATIWADKAR, A. et al. Vehicle License Plate Detection: A Survey. **International Research Journal of Engineering and Technology (IRJET)**, v. 02 i. 08, p. 354-360, Novembro de 2015. ISSN 2395-0056.

BELVISI, R. et al. **Um Sistema de Reconhecimento Automático de Placas de Automóveis**. Anais do II ENIA. Rio de Janeiro: Sociedade Brasileira de Computação. 1999. p. 537-539.

BRADLEY, D.; ROTH, G. Adapting Thresholding Using the Integral Image. **Journal of Graphics Tools**, v. 12 n.02, p. 13-21, 2007.

CONTRAN. RESOLUÇÃO 231 DE 15 DE MARÇO DE 2007. **DEPARTAMENTO NACIONAL DE TRÂNSITO.**, 2007. Disponível em: <[http://www.denatran.gov.br/download/Resolucoes/RESOLUCAO\\_231.pdf](http://www.denatran.gov.br/download/Resolucoes/RESOLUCAO_231.pdf)>. Acesso em: outubro de 2017.

DLAGNEKOV, L.; BELONGIE, S. **UCSD/Calit2 Car License Plate, Make and Model Database**, 2005. Disponível em: <[http://vision.ucsd.edu/car\\_data.html](http://vision.ucsd.edu/car_data.html)>. Acesso em: 13 de junho de 2017.

FARAJIAN, N.; RAHIMI, M. Algorithms for licenseplate detection: A survey. **International Congress on Technology, Communication and Knowledge**, Mashhad, Irã, n. 1, Novembro de 2014. ISSN 978-1-4799-8021-5/14.

GILLY, D.; RAIMOND, K. A Survey on License Plate Recognition Systems. **International Journal of Computer Applications**, v. 61 n. 6, Janeiro 2013. ISSN 0975-8887.

GONZALEZ, R. C. **Processamento digital de imagens**. 3. ed. ed. São Paulo: Pearson Prentice Hall, 2010.

KLETTE, R. **Concise Computer Vision: An Introduction into Theory and Algorithms**. London: Springer-Verlag, 2014.

LAD, A.; PATEL, D. A Survey on License Plate Recognition System. **International Journal of Advance Research and Innovative Ideas in Education**, v. 1 i. 2, p. 53-57, 2015. ISSN 2395-4396.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, november 1998.

MAKAOUI, K.; GUENNOUN, Z.; GHOGHO, M. **Improved license plate localization**. International Conference on Electrical and Information Technologies, 2nd. Tangier, Morocco: IEEE. Maio 2016.

MATHWORKS. Lifting Method for Constructing Wavelets. **The MathWorks, Inc.**, 2017. Disponível em: <<https://www.mathworks.com/help/wavelet/ug/lifting-method-for-constructing-wavelets.html>>. Acesso em: dez 2017.

OTSU, N. A threshold selection method from grey-level. **IEEE Transactions on Systems, Man, and Cybernetics**, 1979. 62-66.

PANG, J. Variance Window Based Car License Plate. **Journal of Computer and Communications**, v. 2, p. 61-69, 2014. Disponível em: <<http://dx.doi.org/10.4236/jcc.2014.29009>>.

REDMON, J.; FARHADI, A. YOLO9000: Better, Faster, Stronger. **arXiv.org**, 2016. Disponível em: <<https://arxiv.org/pdf/1612.08242.pdf>>. Acesso em: dezembro de 2017.

REVISTABW. Matriz de Confusão. **Revista Brasileira de Web**, 2015. Disponível em: <<https://www.revistabw.com.br/revistabw/matriz-de-confusao/>>. Acesso em: dezembro de 2017.

SHAPIRO, L. G.; HARALICK, R. M. **Computer and Robot Vision**. Boston: Addison-Wesley, v. I, 1992.

SILVA, B. L. S. D.; CIARELLI, P. M. Edge Detection and Confidence Map Applied to Identify Textual Elements in Images. **Departamento de Ciência da Computação - UFMG - Laboratório de Banco de Dados**, 2016. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wvc/2016/048.pdf>>. Acesso em: dezembro de 2017.

SMITH, R. An Overview of the Tesseract OCR Engine. **Ninth International Conference on Document Analysis and Recognition**, p. 629-633, 2007. ISSN 0-7695-2822-8/07.

SONAVANE, K.; SONI, B.; MAJHI, U. Survey on Automatic Number Plate Recognition (ANR). **International Journal of Computer Applications**, v. 125 n. 6, Setembro de 2015. ISSN 0975-8887.

UNIFORCE. Placas de carros: Combinação por Estado de A a Z. **SeuCarro.net Classificados**, 2017. Disponível em: <<https://seucarro.net/placas-de-carros-por-estado/>>. Acesso em: dezembro 2017.

---

VIDHYA, N.; SUNDARI, S. G. A Survey on Localization and recognition of license plate number. **Proceedings of the IRF International Conference**, Chennai, India, v. 14, p. 59-62, Setembro de 2014. ISSN 978-93-84209-55-1.

## APÊNDICE A

### A.1 Imagens no Domínio Espacial

Uma imagem digital é basicamente construída pelo processo de amostragem de dados analógicos de uma câmera, no domínio espacial. Uma imagem então pode ser definida como uma matriz de pontos  $(x,y,u)$ , tal que  $(x,y)$  é uma posição e  $u$  é a cor amostrada naquele ponto. Formalmente, uma imagem  $I$  é definida em um *set* retangular  $\Omega$  tal que (KLETTE, 2014):

$$\Omega = \{(x,y) : 1 \leq x \leq \text{Colunas} \wedge 1 \leq y \leq \text{Linhas}\} \subset \mathbb{Z}^2 \quad (6-1)$$

Assume-se um sistema de coordenadas onde  $X$  representa o eixo horizontal,  $Y$  o eixo vertical, e a origem das coordenadas no canto superior esquerdo (KLETTE, 2014).

#### A.1.1 Imagens Binárias e Escalares

Uma imagem binária ou em preto e branco possui apenas dois níveis de intensidade, qual sejam  $u = \{0,1\}$ , representando branco e preto, respectivamente (KLETTE, 2014).

Já nas imagens escalares, os valores que podem ser assumidos por  $u$  são descritos pelo conjunto de valores inteiros  $u = \{0, 1, 2, \dots, 2^a - 1\}$ , neste caso  $a$  representando o número de *bits* utilizado na quantização do canal de cor. Aqui,  $0$  representa a cor preta e  $2^a - 1$  a cor branca. Todos os outros valores de intensidade estão interpolados entre estes extremos (KLETTE, 2014).

Pode-se ainda utilizar mais de um canal de cor, nesse caso  $u$  é um vetor de  $n$  valores inteiros ( $u \in \mathbb{Z}^n$ ), onde  $n$  é a dimensão do espaço de cor. Tipicamente, utiliza-se em imagens coloridas o sistema RGB (*red, green, blue* – vermelho, verde, azul), e a cor é definida como um vetor de  $n=3$  valores inteiros, representando respectivamente as intensidades de vermelho, verde e azul deste ponto  $(x,y)$ .

#### A.1.2 Histograma

Um histograma representa as frequências tabuladas de cada intensidade de uma cor em uma imagem. Ou seja, para cada intensidade soma-se o número de pontos na imagem em que aquela intensidade ocorre, gerando-se um gráfico em barras (KLETTE, 2014). Os histogramas são feitos independentemente para cada canal em uma imagem colorida.



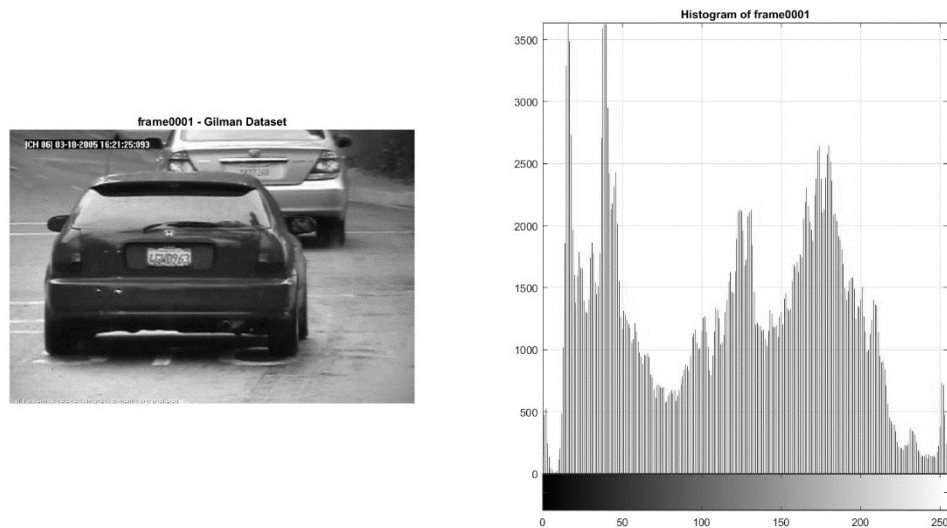
Matematicamente, o histograma  $H_I$  pode ser definido para  $\forall u \in \{0, 1, \dots, I_{max}\}$ , onde  $I_{max} = 2^n - 1$  é a intensidade máxima e  $n$  a quantidade de bits utilizada na quantização, conforme se vê na Equação abaixo (KLETTE, 2014):

$$H_I(u) = |\{(x, y) \in \Omega : I(x, y) = u\}| \quad (6-2)$$

onde  $|\cdot|$  é a cardinalidade de um set, ou seja, o somatório de todas as ocorrências de  $I(x, y) = u$  em  $\Omega$ .

Na Figura 44, pode-se verificar como se distribuem os níveis de cinza no histograma da foto de um veículo.

Figura 44 – Histograma de uma imagem



Fonte: Próprio Autor.

Imagem do veículo extraída do set 'Gilman' (DLAGNEKOV; BELONGIE, 2005).

## A.2 Imagens no domínio da Frequência

### A.2.1 Transformada Discreta de Fourier

A Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) bidimensional mapeia uma imagem  $I$  de  $N_{lin}$  linhas e  $N_{col}$  colunas em uma imagem  $\mathbf{I}$  no plano complexo, denominado domínio da frequência. A DFT bidimensional é dada por (KLETTE, 2014):

$$\mathbf{I}(u, v) = DFT\{I(x, y)\} = \frac{1}{N_{col}N_{lin}} \sum_{x=0}^{N_{col}-1} \sum_{y=0}^{N_{lin}-1} I(x, y) \cdot e^{-j2\pi(\frac{xu}{N_{col}} + \frac{yu}{N_{lin}})} \quad (6-3)$$

onde  $j = \sqrt{-1}$  é o número imaginário dos números complexos,  $u = \{0, 1, \dots, N_{col} - 1\}$  e  $v = \{0, 1, \dots, N_{lin} - 1\}$  são as frequências obtidas na transformada e  $e = 2,71828 \dots$  é o número de Euler.

É possível verificar que a DFT é uma soma de funções seno e cosseno no plano complexo (KLETTE, 2014). Seja  $\alpha = [0, 2\pi)$  um ângulo em radianos. Então pela identidade de Euler<sup>3</sup>:

$$e^{j\alpha} = \cos(\alpha) + j\sin(\alpha) \quad (6-4)$$

### A.2.2 Transformada Inversa Discreta de Fourier

De forma a transformar uma imagem  $\mathbf{I}$  do domínio da frequência de volta para o domínio espacial, utiliza-se a Transformada DFT Inversa (KLETTE, 2014), dada por:

$$I(x, y) = DFT^{-1}\{\mathbf{I}(u, v)\} = \sum_{u=0}^{N_{col}-1} \sum_{v=0}^{N_{lin}-1} \mathbf{I}(u, v) \cdot e^{j2\pi(\frac{xu}{N_{col}} + \frac{yu}{N_{lin}})} \quad (6-5)$$

---

<sup>3</sup> Leonhard Paul Euler (1707-1783) foi um matemático e físico suíço.



```
%% Plate Candidates
[labels, n] = bwlabel(im_filled);
regions = regionprops(labels, 'all');

cropped = 0; %workaround not finding anything

if size(regions)
    bbox = vertcat(regions.BoundingBox);
    w = bbox(:,3);
    h = bbox(:,4);
    aspectRatio = w./h;

    % Filter indices for filtering out segs
    filterIdx = aspectRatio' < 2.5;
    filterIdx = filterIdx | aspectRatio' > 10;
    filterIdx = filterIdx | [regions.Eccentricity] < .8 ;
    filterIdx = filterIdx | [regions.Solidity] < 0.5;
    filterIdx = filterIdx | [regions.Area] < 500;

    % Remove regions
    regions(filterIdx) = [];

    % Sort biggest areas
    [~,index] = sortrows([regions.Area].'); regions = regions(index(end:-1:1));
clear index;

if size(regions)
    R = 3;
    cropped = zeros(size(regions,1), 4);
    for i = 1:size(regions,1)
        region = regions(i);
        Rectangle = region.BoundingBox;
        Width = fix(Rectangle(3));
        Height = fix(Rectangle(4));
        Ratio = Width/Height;
        Rectangle(3) = min(region.MajorAxisLength, Rectangle(3));
        % fix Ratio
        Rectangle(4) = Rectangle(3)/R;
        Rectangle(2) = Rectangle(2) - (Rectangle(4) - Rectangle(3)/Ratio)/2;
        % scale back to original
        cropped(i, :) = Rectangle*Sc;
    end
end
end
end
```

## B.2 Segmentação

```
% UFES - UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
% DEL - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
% PROJETO DE GRADUAÇÃO
% DAVID WILKERSON KÜSTER - 2017

function segchar = segment_characters(frame_original, idx)

%% Preprocess
frame_gray = rgb2gray(frame_original);
frame_original = imresize(frame_gray, [300 NaN]);

% Histogram equilization
frame_hist = histeq(frame_original);

% Binarization
frame_otsu = imbinarize(255-frame_hist);
frame_bin = imbinarize(255-frame_hist, 'adaptive', 'Sensitivity', 0.55);
% figure, subplot(2,1,1); imshow(frame_otsu); title('Otsu'); subplot(2,1,2);
imshow(frame_bin); title('Adaptive');

%% Clear small segs
frame_bin = bwareaopen(frame_bin, 2000);

%% Opening - Morphology
SE = strel('diamond', 7);
frame_erode = imerode(frame_bin, SE);
frame_dilate = imdilate(frame_erode, SE);

% figure, imshow(frame_dilate);

%% Select segments
[labels, n] = bwlabel(frame_dilate);
regions = regionprops(labels, 'all');

% Compute aspect ratio
bbox = vertcat(regions.BoundingBox);
w = bbox(:, 3);
h = bbox(:, 4);
aspectRatio = w./h;

% Filter indices for aspect ratio bigger than 1,5
filterIdx = aspectRatio > 1.5;

% Remove segs
regions(filterIdx) = [];

%% Sort and select 7 biggest areas
[~, index] = sortrows([regions.Area].'); regions = regions(index(end:-1:1)); clear
index;
regions(8:size(regions)) = [];
segchar = regions;

try
    minArea = fix(regions(7).Area);
    bbox = vertcat(regions(:).BoundingBox);

    %% Determine area in original pic to crop out 7 biggest areas
    bbox = sort(bbox, 1);
    xmin = bbox(1, 1);
    ymin = bbox(1, 2);
    wmax = norm(bbox(7, 1)+bbox(7, 3)-bbox(1, 1));
    hmax = norm(bbox(7, 2)+bbox(7, 4)-bbox(1, 2));
```

```
% crop around central characters
frame_crop = imcrop(frame_dilate, [xmin ymin wmax hmax]);

%% clear image based on 7th smallest char area
frame_crop = bwareaopen(frame_crop, minArea);

%% Opening Operator to improve Characters
SE = strel('disk',1);
frame_crop = imerode(frame_crop,SE);
frame_crop = imdilate(frame_crop,SE);

%% Selects segments left
[labels, n] = bwlabel(frame_crop);
regions = regionprops(labels);

%% Generates output
k = size(regions,1);
segchar = cell(k,1);
% figure,
for i = 1:k
    regionImage = imcrop(frame_crop, regions(i).BoundingBox);
    % segchar{i,1} = padarray(regionImage,[10 10]); %for Tesseract OCR only
    segchar{i,1} = regionImage;
    % subplot(1,k,i);
    % imshow(segchar{i,1});
end

%% Plot segments' boxes
%box = insertShape(255*frame_crop, 'Rectangle',vertcat(regions(:).BoundingBox),
'LineWidth', 3);
% figure,imshow(box)
% imwrite(box, (strcat('\BR_DB\Segmentation\(',num2str(idx),')'.jpg')));
catch
end
end
```

### B.3 Reconhecimento de Caracteres

```

% UFES - UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
% DEL - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
% PROJETO DE GRADUAÇÃO
% DAVID WILKERSON KÜSTER - 2017

function char = recognize_character_cmp(segchar)

%% Read reference font
REF = cell(36,1);

%% Load a priori knowledge base
[idx i1 i2 i3] = apriori();

% figure,
for i=1:size(idx,2)
    REF{i}=imbinarize(imread(strcat('.',BR_DB\ref_font\',idx(i),'.jpg')));
    % subplot(6,6,i);imshow(REF{i});title(num2str(idx(i)));
end

%% Recognition
char = '';
sz = size(segchar,1);
thresh = 0.5;

% figure,
for i=1:sz
    I = segchar{i};
    % subplot(1,sz,i);imshow(I);
    if i<4
        if i==1
            img = i1;
        elseif i==2
            img = i2;
        elseif i==3
            img = i3;
        end
    else
        img = 1:10;
    end
    c = 0;
    overlapMax = 0;
    % figure,
    for img = img
        ref = REF{img};
        [w h] = size(I);
        ref = imresize(ref, [w h]);
        overlap = abs(sum(I&ref))/abs(sum(I|ref));
        if overlap > overlapMax && overlap > thresh
            c = idx(img);
            overlapMax = overlap;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if c
    char = strcat(char, c);
else
    char = strcat(char, '?');
end
end
end

```