

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTOS ACADÊMICOS DE ELETRÔNICA E  
INFORMÁTICA

ANDRÉ LUIZ COSTANTINO BOTTA

**ANÁLISE COMPARATIVA DE MÉTODOS DE  
SEGMENTAÇÃO E TÉCNICAS DE APRENDIZADO DE  
MÁQUINA COM APLICAÇÃO NO RECONHECIMENTO  
AUTOMÁTICO DE PLACAS DE IDENTIFICAÇÃO DE  
VEÍCULOS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2015

ANDRÉ LUIZ COSTANTINO BOTTA

**ANÁLISE COMPARATIVA DE MÉTODOS DE  
SEGMENTAÇÃO E TÉCNICAS DE APRENDIZADO DE  
MÁQUINA COM APLICAÇÃO NO RECONHECIMENTO  
AUTOMÁTICO DE PLACAS DE IDENTIFICAÇÃO DE  
VEÍCULOS**

Trabalho de Conclusão do Curso de Engenharia de Computação apresentado aos Departamentos Acadêmicos de Eletrônica e Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Engenheiro em Computação”.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Leyza Baldo Dorini

Orientador: Prof. Dr. Rodrigo Minetto

**CURITIBA**

**2015**

*Aos meus pais, Márcia e Aparicio,  
por sempre terem me apoiado.*

## AGRADECIMENTOS

Gostaria de agradecer especialmente aos meus orientadores, Prof. Dr. Rodrigo Minetto e Prof<sup>a</sup>. Dr<sup>a</sup> Leyza Baldo Dorini, que, até mesmo antes da ideia do projeto estar clara e definida, sempre se mostraram dispostos a me ajudar e guiar.

## RESUMO

BOTTA, André Luiz Costantino. ANÁLISE COMPARATIVA DE MÉTODOS DE SEGMENTAÇÃO E TÉCNICAS DE APRENDIZADO DE MÁQUINA COM APLICAÇÃO NO RECONHECIMENTO AUTOMÁTICO DE PLACAS DE IDENTIFICAÇÃO DE VEÍCULOS. 72 f. Trabalho de Conclusão de Curso – Departamentos Acadêmicos de Eletrônica e Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Este trabalho trata do problema de Reconhecimento Automático de Placas de Veículos (do inglês, *Automatic Number Plate Recognition* - ANPR), o qual envolve o uso de diferentes técnicas, incluindo Reconhecimento Óptico de Caracteres (do inglês, *Optical Character Recognition* - OCR) e aprendizado de máquina, para realizar o processo de reconhecimento dos caracteres em uma placa de identificação de veículo a partir de uma imagem. Como este processo está sob condições reais, existem dificuldades no tocante aos diversos tipos de ruídos que podem estar presentes na aquisição das imagens, tais como iluminação não uniforme, oclusão de caracteres devido à algum tipo de objeto fixado à placa de identificação do veículo e posicionamento dos caracteres, entre outros. Dado que o método de segmentação aplicado tem influência direta na atenuação do efeito negativo destes ruídos, um dos objetivos deste trabalho foi determinar qual destes métodos otimiza o desempenho tanto do OCR desenvolvido quanto do módulo de OCR do mecanismo Tesseract — um software considerado estado da arte nesta área. Por fim, utilizando a precisão e a revocação como métodos de avaliação, analisou-se qual técnica de aprendizado de máquina obtém os melhores resultados no reconhecimento de caracteres e comparou-se estes resultados com o Tesseract, a fim de verificar-se o desempenho da abordagem proposta.

**Palavras-chave:** Máquina de Vetor Suporte, Perceptron Multi-Camadas, Classificador Bayes, K-Vizinhos Mais Próximos, Aprendizado de Máquina, Reconhecimento Automático de Placa de Veículos, Reconhecimento Óptico de Caracteres.

## ABSTRACT

BOTTA, André Luiz Costantino. COMPARATIVE ANALYSIS OF SEGMENTATION METHODS AND MACHINE LEARNING TECHNIQUES WITH APPLICATION IN AUTOMATIC NUMBER PLATE RECOGNITION. 72 f. Trabalho de Conclusão de Curso – Departamentos Acadêmicos de Eletrônica e Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

The present study deals with the problem of Automatic Number Plate Recognition (ANPR), which involves the use of different techniques, including Optical Character Recognition (OCR) and machine learning, to perform the process of character recognition of a vehicle license plate in an image. As this process is done under real conditions, there are difficulties with regard to various types of noise that may be present in the acquisition of images, such as irregular lighting, occlusion of characters due to some kind of object attached to the vehicle license plate and positioning of the characters, among others. Since the applied segmentation method has a direct influence in the attenuation of the negative effects of these noise, one of the objectives of this study is to evaluate which of these methods optimizes the performance of both the OCR developed and the Tesseract OCR engine module — a software considered to be the state of the art in this area. Lastly, using the precision and recall values as evaluation measures, we analyze which machine learning technique obtain the best results in character recognition and compare these results with Tesseract in order to verify the performance of the proposed approach.

**Keywords:** Support Vector Machine, Multilayer Perceptron, Bayes classifier, K-Nearest Neighbors, Machine Learning, Automatic Number Plate Recognition, Optical Character Recognition.

## LISTA DE FIGURAS

FIGURA 1	–	Etapas de um sistema de ANPR. ....	12
FIGURA 2	–	Quatro tipos diferentes de imagem e seus respectivos histogramas. .	17
FIGURA 3	–	Histograma que pode ser particionado com: (a) um único valor de limiar; (b) múltiplos valores de limiares. ....	19
FIGURA 4	–	Exemplo para o cálculo da precisão e revocação. ....	22
FIGURA 5	–	Rede neural do tipo <i>perceptron</i> . ....	27
FIGURA 6	–	<i>Perceptron</i> multi-camadas. ....	28
FIGURA 7	–	Modelo SVM. ....	29
FIGURA 8	–	Hiperplano ótimo gerado pela SVM para a separação linear. ....	30
FIGURA 9	–	Exemplo de um classificador naïve de Bayes. ....	30
FIGURA 10	–	Diagrama de blocos representando os módulos do trabalho. ....	33
FIGURA 11	–	Imagem não selecionada devido à baixa nitidez em relação à motocicleta. ....	34
FIGURA 12	–	Imagem em condições aceitáveis de uso. ....	35
FIGURA 13	–	Imagem em ótimas condições de uso. ....	35
FIGURA 14	–	Imagem de exemplo analisada para a extração de informações quanto à posição da placa e dos caracteres da mesma. ....	37
FIGURA 15	–	Exemplos dos resultados obtidos por cada método de segmentação: (a) imagem original em níveis de cinza; (b) Otsu; (c) Niblack; (d) Sauvola; (e) Wolf e (f) <i>SnooperText</i> . ....	42
FIGURA 16	–	Imagens de exemplo para a extração dos momentos de Hu. ....	45
FIGURA 17	–	Análise gráfica dos resultados do Cenário 3: Tesseract alterado em relação ao Tesseract completo em função da precisão. ....	58
FIGURA 18	–	Análise gráfica dos resultados do Cenário 3: Tesseract alterado em relação ao Tesseract completo em função da revocação. ....	59
FIGURA 19	–	Comparação entre o método de segmentação de Otsu implementado neste trabalho e o do Tesseract. ....	60

## LISTA DE TABELAS

TABELA 1	–	Configuração utilizada para os métodos de segmentação. ....	43
TABELA 2	–	Resultado da extração dos momentos de Hu das Figuras 16 (a) – 16 (c). ....	45
TABELA 3	–	Disposição dos caracteres utilizados para o treinamento das técnicas de aprendizado de máquina. ....	46
TABELA 4	–	Configuração utilizada para as técnicas de aprendizado de máquina. ....	47
TABELA 5	–	Disposição dos caracteres utilizados para a comparação entre os métodos de segmentação e as técnicas de aprendizado de máquina. ....	48
TABELA 6	–	Compilação dos melhores resultados em relação à precisão para todos os caracteres. ....	50
TABELA 7	–	Compilação dos melhores resultados em relação à precisão para o conjunto de caracteres alfabéticos. ....	51
TABELA 8	–	Compilação dos melhores resultados em relação à precisão para o conjunto de caracteres numéricos. ....	52
TABELA 9	–	Resultados para o Cenário 1: Tesseract completo em relação à abordagem proposta. ....	54
TABELA 10	–	Resultados para o Cenário 2: Tesseract alterado em relação à abordagem proposta. ....	55
TABELA 11	–	Resultados para o Cenário 3: Tesseract alterado em relação ao Tesseract completo. ....	56
TABELA 12	–	Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Otsu. ....	67
TABELA 13	–	Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Niblack. ....	68
TABELA 14	–	Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Sauvola. ....	69
TABELA 15	–	Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Wolf. ....	70
TABELA 16	–	Resultado de cada técnica de aprendizado de máquina para o método de segmentação <i>SnooperText</i> . ....	71
TABELA 17	–	Resultados do Tesseract para todos os métodos de segmentação. ....	72



## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 OBJETIVOS	13
1.2 ESTADO-DA-ARTE	14
1.3 ESTRUTURA DO TRABALHO	14
<b>2 LEVANTAMENTO BIBLIOGRÁFICO</b>	<b>16</b>
2.1 CONCEITOS BÁSICOS	16
2.1.1 Histograma	16
2.1.2 Limiarização	18
2.1.3 Descritor de imagem	18
2.1.4 Reconhecimento óptico de caracteres	20
2.1.5 Método de avaliação	21
2.1.6 Tesseract-OCR	23
2.2 SEGMENTAÇÃO DE IMAGEM	23
2.2.1 Método de Otsu	23
2.2.2 Método de Niblack	24
2.2.3 Método de Sauvola	25
2.2.4 Método de Wolf	25
2.2.5 Método SnooperText	26
2.3 APRENDIZADO DE MÁQUINA	26
2.3.1 Redes neurais artificiais	27
2.3.2 Máquina de vetor suporte	27
2.3.3 Classificador naïve Bayes	29
2.3.4 K-vizinhos mais próximos	31
2.3.5 Gradient boosted trees	31
<b>3 DESENVOLVIMENTO</b>	<b>33</b>
3.1 AQUISIÇÃO DE IMAGEM	34
3.2 EXTRAÇÃO DO POSICIONAMENTO DA PLACA DE IDENTIFICAÇÃO DO VEÍCULO E DOS SEUS RESPECTIVOS CARACTERES	36
3.3 MÓDULO DE SEGMENTAÇÃO	37
3.3.1 Codificação do método de segmentação de Otsu	37
3.3.2 Codificação dos métodos de segmentação de Niblack, Sauvola e Wolf	39
3.3.3 Codificação do método de segmentação <i>SnooperText</i>	41
3.3.4 Análise dos resultados dos métodos de segmentação	42
3.4 MÓDULO DE OCR	43
3.4.1 Codificação dos momentos de Hu	43
3.4.2 Treinamento das técnicas de aprendizado de máquina	46
<b>4 ANÁLISE DE DESEMPENHO E RESULTADOS</b>	<b>48</b>
4.1 COMPARAÇÃO COM O TESSERACT	49
4.1.1 Resultados do Cenário 1: Tesseract completo em relação à abordagem proposta	52
4.1.2 Resultados do Cenário 2: Tesseract alterado em relação à abordagem proposta	53
4.1.3 Resultados do Cenário 3: Tesseract alterado em relação ao Tesseract completo	53

<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>61</b>
5.1	PROBLEMAS ENCONTRADOS	61
5.2	TRABALHOS FUTUROS	62
	<b>REFERÊNCIAS</b>	<b>63</b>
	<b>Apêndice A – TABELAS DE RESULTADOS</b>	<b>66</b>

## 1 INTRODUÇÃO

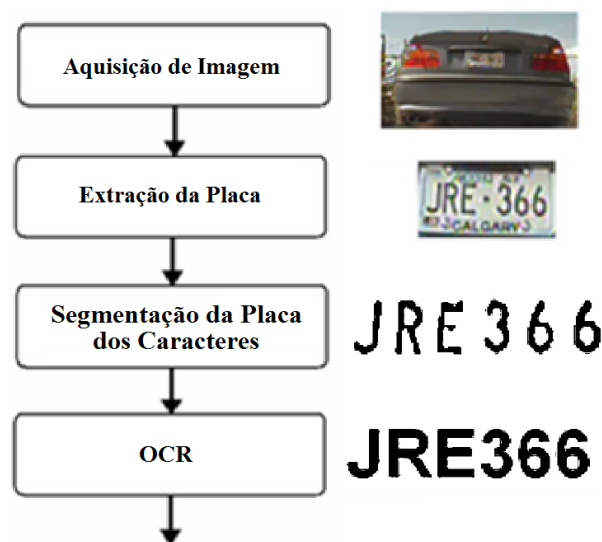
Dos cinco sentidos que os seres humanos utilizam para perceber o ambiente à sua volta, a visão é considerada a mais poderosa, dado que aproximadamente 99% da atividade do cérebro humano está envolvida no processamento de imagens no córtex visual (DOUGHERTY, 2009). Embora possa parecer natural e simples para o ser humano, atividades desempenhadas pelo sistema visual, tais como reconhecer rostos e letras, constituem uma enorme dificuldade quando precisam ser replicadas por sistemas computacionais (DOUGHERTY, 2009). A área de pesquisa ligada ao desenvolvimento de sistemas de *hardware* e *software* que buscam assemelhar-se à visão humana é denominada visão computacional. Ela emprega sistemas de aquisição de imagens e algoritmos sofisticados para desempenhar, de forma ainda primária, ações efetuadas pelo sistema de visão (MOESLUND, 2012).

O enfoque deste trabalho é o Reconhecimento Automático de Placas de Veículos (do inglês, *Automatic Number Plate Recognition* - ANPR), um sub-ramo de pesquisa dentro da área de visão computacional. O ANPR envolve o uso de métodos de segmentação e de Reconhecimento Óptico de Caracteres (do inglês, *Optical Character Recognition* - OCR) para reconhecer os caracteres contidos na placa de identificação de um veículo (BAGGIO, 2012). Para realizar o processo de OCR, pode-se aplicar descritores de imagem (preferencialmente invariantes à translação, rotação e mudança de escala) para se extrair um vetor com dados que a representam.

Após esse estágio, técnicas de aprendizado de máquina, tais como Perceptron Multi-Camadas (do inglês, *Multilayer Perceptron* - MLP), Máquina de Vetores Suporte (do inglês, *Support Vector Machine* - SVM), Classificador Naïve Bayes (do inglês, *Naïve Bayes Classifier* - NBC), K-Vizinhos Mais Próximos (do inglês, *K-Nearest Neighbors* - KNN) e *Gradient Boosted Trees* - GBT; possibilitam a transformação desses dados em informação de alto nível por meio da extração de regras ou padrões assimilados (BRADSKI; KAEHLER, 2008) (WINDER; BROWN, 2007).

De forma geral, um sistema completo de ANPR pode ser resumido em quatro

módulos, como mostra a Figura 1. É possível observar que o primeiro módulo consiste na aquisição da imagem, seja em tempo real por uma câmera de vídeo ou a partir de um vídeo armazenado em disco. Em seguida, realiza-se a extração do posicionamento da placa, visando identificar a região de interesse na qual a segmentação e o OCR serão aplicados. O terceiro módulo é o de segmentação da placa dos caracteres, extremamente significativo para o sucesso do OCR. Seu propósito é o de dividir a imagem em objeto de interesse e fundo, ou seja, diferenciar os caracteres do fundo e de ruídos externos. Neste trabalho, utiliza-se os métodos de segmentação de Otsu, Niblack, Sauvola, Wolf e *SnooperText* para este fim. Por último, o módulo de OCR realiza o reconhecimento dos caracteres.



**Figura 1:** Etapas de um sistema de ANPR.  
Fonte: Adaptado de (DU, 2013).

O propósito deste trabalho é o de desenvolver e avaliar os módulos de segmentação e OCR de um sistema de ANPR. O desenvolvimento do módulo de segmentação compreende a codificação dos métodos de segmentação de Otsu, Niblack, Sauvola, Wolf e *SnooperText* (descritos na Seção 2.2). Para o módulo de OCR, foi implementado o descritor de imagem conhecido como momentos de Hu (Seção 2.1.3) e utilizou-se as técnicas de aprendizado de máquina MLP, SVM, NBC, KNN e GBT (descritas na Seção 2.3), fornecidas pela biblioteca de visão computacional OpenCV (BRADSKI; KAEHLER, 2008).

A avaliação destes módulos envolve determinar qual método de segmentação e técnica de aprendizado de máquina obtém o melhor desempenho, em relação à precisão e revocação, e comparar esses resultados com o software Tesseract (SMITH, 2007), descrito na Seção 2.1.6, uma das ferramentas mais avançadas de OCR. Também avalia-se o grau

de eficiência do módulo de segmentação desenvolvido ao integrá-lo ao módulo de OCR do Tesseract. Com isso, é possível definir se o módulo de segmentação implementado neste trabalho otimiza o desempenho daquele.

Dentre os desafios que um sistema ANPR encontra em situações reais e que serão consideradas neste trabalho, pode-se citar: (a) tamanho variado da placa de identificação em razão da distância da câmera ou do fator de *zoom* da mesma; (b) oclusão das informações presentes na placa de identificação; (c) inclinação da placa de acordo com o ângulo de visualização da imagem e (d) diferenças de iluminação que podem incidir no processo de aquisição de imagem devido ao ambiente ou em consequência das luzes de faróis.

Uma vez que as aplicações de sistemas de ANPR incluem desde uma simples ferramenta auxiliar ao gerenciamento de estacionamentos até o complexo controle de segurança de uma área extremamente restrita como, por exemplo, zonas militares ou prédios governamentais (QADRI; ASIF, 2009), é interessante notar que o OCR destes sistemas apresenta, atualmente, taxas de acerto de apenas 10% em média, para aplicações reais (KLUZNER, 2011). Desta forma, a análise comparativa proposta neste trabalho gerará avanços quanto aos mecanismos de ANPR ao se avaliar quais métodos de segmentação e técnicas de aprendizado de máquina implicam em melhores taxas de precisão e revocação para os mecanismos de OCR.

## 1.1 OBJETIVOS

Os objetivos do trabalho são:

- Objetivo geral: Desenvolver os módulos de segmentação e OCR de um sistema ANPR e analisar o desempenho desses módulos, assim como comparar os resultados da abordagem proposta em relação ao software Tesseract.
- Objetivos específicos:
  - Analisar comparativamente os módulos de segmentação e OCR desenvolvidos, isto é, avaliar todas as 25 combinações possíveis entre os métodos de segmentação e as técnicas de aprendizado de máquina;
  - Substituir o módulo de segmentação do Tesseract pelo módulo de segmentação proposto, a fim de avaliar quantitativamente a melhora de desempenho do Tesseract;

- Implementar um descritor de imagem que seja invariante às transformações de translação, rotação, mudança de escala e espelhamento;
- Criar uma base de dados de placas de identificação de veículos e dos caracteres contidos nestas.

## 1.2 ESTADO-DA-ARTE

A história de pesquisa e desenvolvimento sobre OCR pode ser rastreada até o começo da década de 1950, quando cientistas tentaram capturar caracteres e texto de imagens por meios mecânicos e ópticos mediante a rotação de discos e fotomultiplicadores utilizando um *scanner* com tubo de raios catódicos seguido por uma sequência de fotocélulas (CHERIET, 2007). Esse processo, além de lento, só podia ser realizado para uma linha de texto por vez. Com a invenção do *scanner* de mesa, pode-se estender o procedimento para toda a página. Com o avanço tecnológico, essa tarefa tornou-se mais precisa e veloz. Nas décadas de 1960 e 1970, novas aplicações de OCR chegaram ao mercado de varejo, bancos, hospitais, correios, publicadores de jornais e muitas outras indústrias. Paralelamente à estes avanços em desenvolvimento de hardware, uma intensa pesquisa sobre OCR estava se formando, tanto em âmbito acadêmico quanto industrial (CHERIET, 2007). Para diminuir a taxa de erros das técnicas de OCR na época, padrões de tinta, papel e fonte foram criados (CHERIET, 2007).

Ainda na década de 1970, a Seção de Desenvolvimento da Polícia Científica do Reino Unido desenvolveu o que hoje é conhecido como ANPR. Entretanto, foi década de 2000 que o interesse nesta área ganhou força, grande parte em razão do aperfeiçoamento das câmeras digitais e o aumento em capacidade computacional (QADRI; ASIF, 2009).

Desenvolvido pela HP entre os anos de 1984 e 1994, o Tesseract é um mecanismo de OCR tido como o estado da arte da área atualmente (SMITH, 2007) (KAE, 2010). Em 1995, após o projeto já estar finalizado, o Tesseract participou de uma competição anual de acurácia de OCR, onde provou sua importância contra os mecanismos da época. Em 2005, a HP tornou o Tesseract *open source* (SMITH, 2007).

## 1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta, brevemente, os conceitos básicos para a compreensão do restante do trabalho, descrevendo a base teórica que sustenta o projeto em termos de terminologia, abordagens, métodos

e técnicas. O Capítulo 3 expõe como realizou-se o desenvolvimento do trabalho a fim de cumprir-se os objetivos do mesmo. Em seguida, no Capítulo 4, discute-se os resultados obtidos e suas implicações. Por fim, o Capítulo 5 apresenta um panorama geral do trabalho, assim como limitações do mesmo e sugestões de trabalhos futuros.

## 2 LEVANTAMENTO BIBLIOGRÁFICO

Este capítulo aborda os conceitos necessários para compreender os métodos de segmentação e técnicas de aprendizado de máquina empregados neste projeto. Além disso, são apresentados conceitos básicos pertinentes ao trabalho.

### 2.1 CONCEITOS BÁSICOS

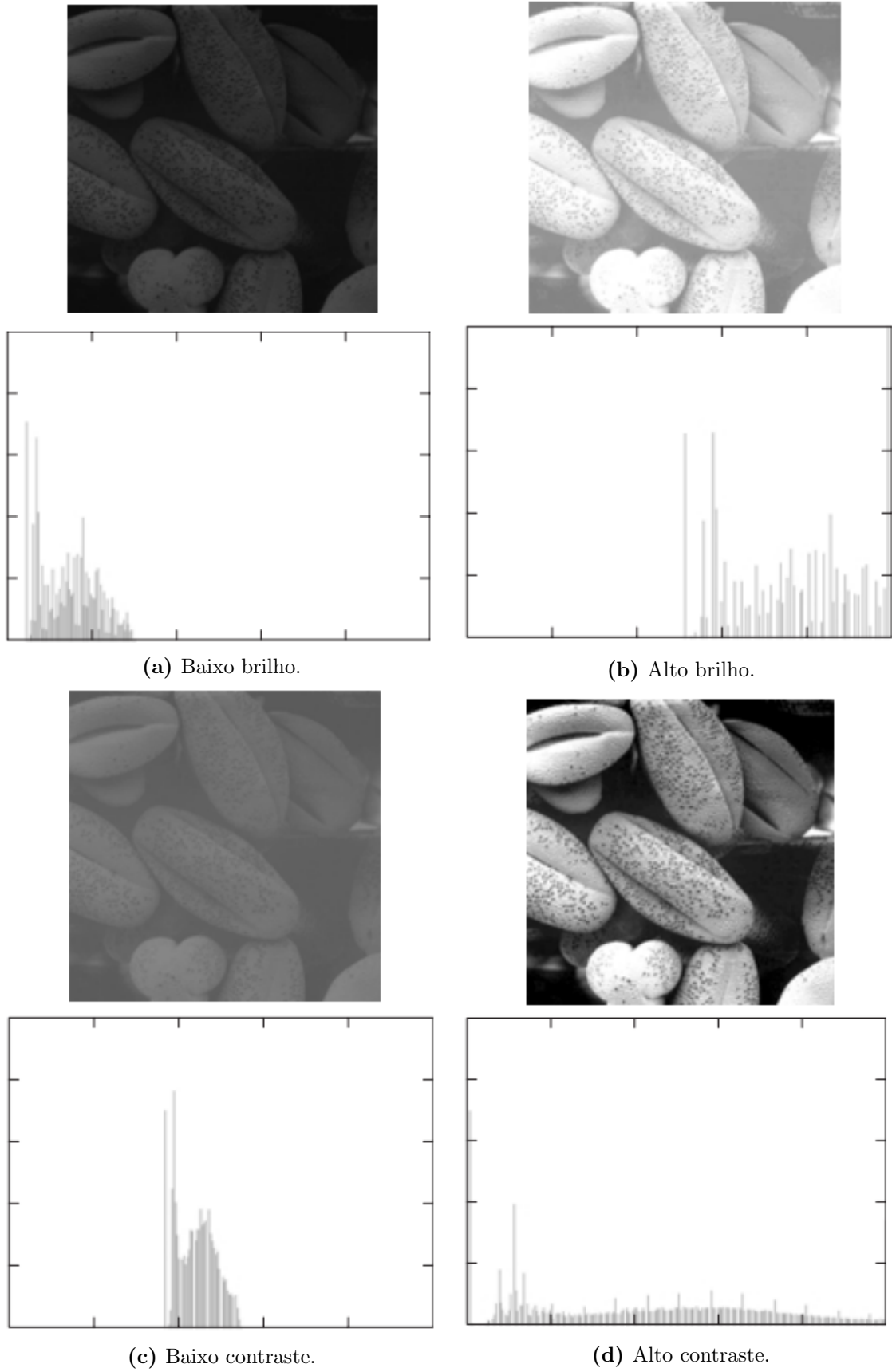
Esta seção define conceitos relevantes de processamento digital de imagens, realiza uma breve discussão sobre OCR e o método de avaliação, baseado em precisão e revocação, além de, resumidamente, apresentar o *software* Tesseract.

#### 2.1.1 HISTOGRAMA

Dada uma imagem  $f(x,y)$  com valores de intensidade no intervalo  $[0, L-1]$ , sendo  $L$  o número de níveis de cinza da imagem, define-se histograma como uma função discreta, na forma  $h(r_k) = n_k$ , em que  $r_k$  é o  $k$ -ésimo valor de intensidade e  $n_k$  é o número de *pixels* com intensidade  $r_k$ .

Uma operação comum aplicada a histogramas é a sua normalização. Uma das possibilidades consiste em dividir cada componente pelo total de *pixels* na imagem. Com isso, um histograma normalizado é dado por  $p(r_k) = n_k/NM$ , para  $k=0, 1, 2, \dots, L-1$ , em que  $M$  é o número de linhas e  $N$  é o número de colunas da imagem. Assim sendo,  $p(r_k)$  pode ser definido como a probabilidade de ocorrência de um nível de cinza,  $r_k$ , arbitrário. O resultado do somatório dos componentes normalizados é igual à 1 (GONZALEZ; WOODS, 2007). A Figura 2 mostra a mesma imagem com quatro diferentes características de histograma: baixo brilho, alto brilho, baixo contraste e alto contraste.





**Figura 2:** Quatro tipos diferentes de imagem e seus respectivos histogramas.  
 Fonte: Adaptado de (GONZALEZ; WOODS, 2007).

### 2.1.2 LIMIAZIZAÇÃO

O conceito básico de limiarização é que, dado um histograma de uma imagem  $f(x,y)$ , onde o objeto de interesse e o fundo tenham valores de intensidade claramente distribuídos em dois grupos, pode-se extrair o objeto do fundo ao selecionar um valor de limiar,  $T$ , que separe estes grupos. Deste modo, qualquer *pixel*  $(x,y)$  na imagem que satisfaça  $f(x,y) > T$  é chamado de *pixel* de objeto. Caso contrário, o ponto é definido como fundo. A Figura 3 (a) apresenta o histograma de uma imagem que pode ser limiarizada aplicando-se este conceito básico. Na Figura 3 (b) observa-se um histograma um pouco mais complexo, sendo necessária a utilização de dois valores de limiar,  $T_1$  e  $T_2$ . Nesse caso, o ponto  $(x,y)$  pode ser classificado em três classes distintas, dependendo das condições  $f(x,y) \leq T_1$ ,  $T_1 < f(x,y) \leq T_2$  e  $f(x,y) > T_2$ .

O valor de limiarização pode ser definido por meio de uma função,  $f_T$ , como apresentado na Equação 2.1 (GONZALEZ; WOODS, 2007):

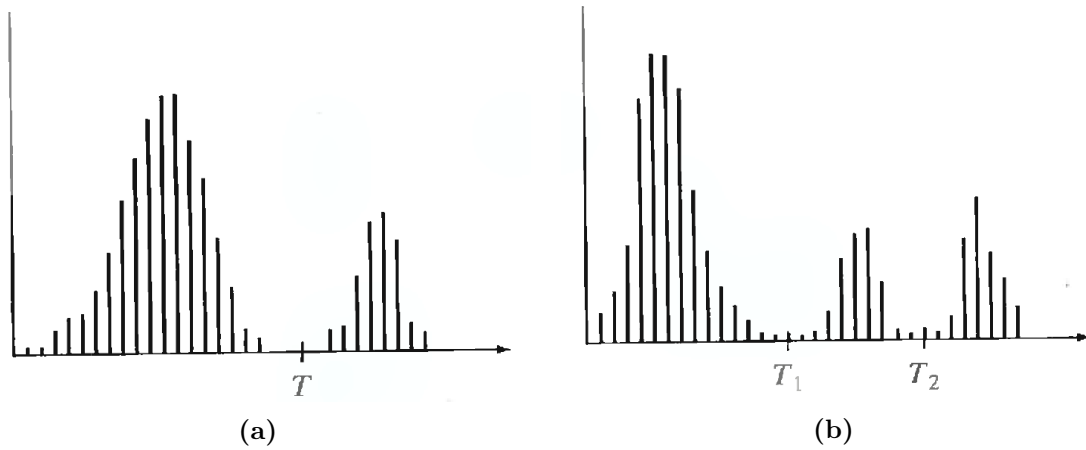
$$T = f_T[x,y,p(x,y),f(x,y)], \quad (2.1)$$

em que  $f(x,y)$  é o valor de nível de cinza no ponto  $(x,y)$  e  $p(x,y)$  caracteriza alguma propriedade local do ponto como, por exemplo, a média local de níveis de cinza da vizinhança centralizada em  $(x,y)$ . Se  $f_T$  não avaliar  $p(x,y)$ , a limiarização é dita como global, do contrário, ela é chamada de local. Assim sendo, a imagem limiarizada,  $g(x,y)$ , é definida na Equação 2.2 (GONZALEZ; WOODS, 2007):

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) > T, \\ 0, & \text{caso contrário.} \end{cases} \quad (2.2)$$

### 2.1.3 DESCRITOR DE IMAGEM

Uma região de uma imagem pode ser representada de diferentes formas como, por exemplo, em função de suas bordas. Contudo, essas representações tipicamente não permitem a extração de informações úteis para processamento e análise. Para extrair informação da imagem, pode-se utilizar descritores que, a partir de uma representação, sintetizam informações para uso computacional (GONZALEZ; WOODS, 2007). Este trabalho utiliza o descritor conhecido como momentos de Hu. Momento, em processamento digital de imagens, refere-se à informação geométrica de uma imagem (SHUTLER, 2015).



**Figura 3:** Histograma que pode ser particionado com: (a) um único valor de limiar; (b) múltiplos valores de limiares.

Fonte: (GONZALEZ; WOODS, 2007)

Para uma imagem digital  $f(x,y)$  de dimensões  $M \times N$ , seu momento 2-D de ordem  $(p+q)$  é definido como mostra a Equação 2.3:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p \times y^q \times f(x,y), \quad (2.3)$$

em que  $p = 0, 1, 2, \dots$  e  $q = 0, 1, 2, \dots$  são inteiros. Define-se o correspondente momento central como exposto na Equação 2.4:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x-\bar{x})^p \times (y-\bar{y})^q \times f(x,y), \quad (2.4)$$

para  $p = 0, 1, 2, \dots$  e  $q = 0, 1, 2, \dots$ , em que  $\bar{x}$  e  $\bar{y}$  são, respectivamente, definidos como apresentado nas Equações 2.5 e 2.6:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad (2.5)$$

$$\bar{y} = \frac{m_{01}}{m_{00}}. \quad (2.6)$$

O momento central normalizado, denotado por  $\eta_{pq}$ , é definido na Equação 2.7:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}}, \quad (2.7)$$

em que  $\gamma = \frac{p+q}{2} + 1$ , para  $p+q = 2, 3, \dots$

As Equações 2.8 à 2.14 mostram o conjunto de sete momentos invariantes à translação, rotação, mudança de escala e espelhamento, denominados como momentos de Hu, que são derivados do segundo e terceiro momentos (GONZALEZ; WOODS, 2007):

$$\phi_1 = \eta_{20} + \eta_{02}, \quad (2.8)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4 \times \eta_{11}^2, \quad (2.9)$$

$$\phi_3 = (\eta_{30} - 3 \times \eta_{12})^2 + (3 \times \eta_{21} - \eta_{03})^2, \quad (2.10)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \quad (2.11)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12}) \times (\eta_{30} + \eta_{12}) \times [(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) \times (\eta_{21} + \eta_{03}) \times \\ & [3 \times (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \end{aligned} \quad (2.12)$$

$$\begin{aligned} \phi_6 = & (\eta_{20} - \eta_{02}) \times [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4 \times \eta_{11} \times (\eta_{30} + \eta_{12}) \times (\eta_{21} + \eta_{03}), \end{aligned} \quad (2.13)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03}) \times (\eta_{30} + \eta_{12}) \times [(\eta_{30} - \eta_{12})^2 \\ & - 3 \times (\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30}) \times (\eta_{21} + \eta_{03}) \times \\ & [3 \times (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]. \end{aligned} \quad (2.14)$$

No contexto deste trabalho, tal descritor é utilizado para a extração de informações que caracterizam cada caractere. Essas informações serão utilizadas pelas técnicas de aprendizado de máquina discutidas na Seção 2.3, para realizar o OCR, apresentado na próxima subseção.

#### 2.1.4 RECONHECIMENTO ÓPTICO DE CARACTERES

OCR pode ser definido como a análise de caracteres em imagens digitais e a posterior tradução do texto encontrado para o formato ASCII, o qual possibilita que o

texto digitalizado seja armazenado e manipulado eletronicamente de forma mais eficiente (BEITZEL, 2003). Atualmente, as técnicas de OCR estão sendo aplicadas em vários tipos de projetos, incluindo as áreas (BOROVIKOV, 2014):

- OCR adaptativo: seu objetivo é um robusto manuseio de uma vasta gama de imagens de documentos impressos abordando: (a) Segmentação automática de documentos; (b) Reconhecimento de notação matemática;
- Reconhecimento de escrita à mão: é uma tecnologia de OCR que precisa ser extremamente robusta e adaptativa. Embora amadurecida, a área continua ativa para pesquisas já que muitos de seus problemas são solucionados apenas em condições especiais, tais como: (a) Reconhecimento de escrita à mão em cheques; (b) Leitores de envelopes postais; (c) OCR em dispositivos portáteis e de mão;
- Aprimoramento digital de imagem: envolve a escolha e aplicação de filtros apropriados à uma imagem com o propósito de ajudar o mecanismo de OCR a conceber melhores resultados;
- OCR em multimídia: é uma adaptação das técnicas de OCR para mídias que não sejam documentos impressos, como fotos, vídeos e a internet.

Além de dificuldades como ruído, segmentação imprecisa e baixa resolução, que influenciam negativamente na taxa de acerto dos mecanismos de OCR, há ainda fatores como a própria representação de certos caracteres que criam “confusões”. Por exemplo, uma tarefa árdua para um OCR pode ser diferenciar a letra “O” do numeral “0” (BOROVIKOV, 2014).

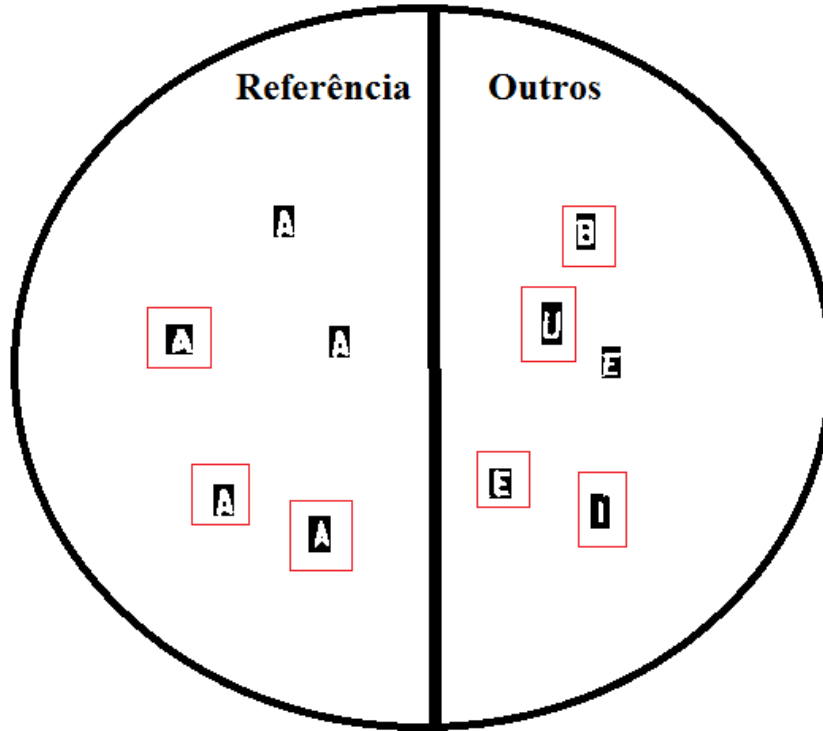
### 2.1.5 MÉTODO DE AVALIAÇÃO

Para avaliar a acurácia de um método OCR, as duas medidas mais comuns são: precisão, cuja representação matemática pode ser vista na Equação 2.15, e revocação, expressa na Equação 2.16. Tanto a precisão quanto a revocação estão no intervalo [0.0, 1.0] (BOROVIKOV, 2014):

$$\text{precisão} = \frac{\text{número de itens corretos}}{\text{número de itens na saída do OCR}}, \quad (2.15)$$

$$\text{revocação} = \frac{\text{número de itens corretos}}{\text{número de itens na referência}}. \quad (2.16)$$

A Figura 4 apresenta um exemplo de como calcula-se a precisão e a revocação. Foram utilizadas algumas das imagens de caracteres reais sendo analisadas neste projeto. Como pode ser observado, os caracteres já estão segmentados.



**Figura 4:** Exemplo para o cálculo da precisão e revocação.  
Fonte: Autoria própria.

Suponha que deseja-se calcular a precisão e revocação em relação aos caracteres “A”, os quais compõem a referência, com um total de cinco itens. Uma das técnicas de aprendizado de máquina utilizada no projeto foi aplicada após o processo de extração e apresentação do descritor de cada uma das 10 imagens. O OCR reconhece os caracteres destacados (retângulo em volta do caractere) como “A”. Com isso, têm-se três itens corretos e quatro itens incorretos, totalizando sete itens na saída do OCR. Logo, a precisão e a revocação são, respectivamente, as apresentadas nas Equações 2.17 e 2.18:

$$precisão = \frac{\text{número de itens corretos}}{\text{número de itens na saída do OCR}} = \frac{3}{3+4} = \frac{3}{7} = 0,43, \quad (2.17)$$

$$revocação = \frac{\text{número de itens corretos}}{\text{número de itens na referência}} = \frac{3}{5} = 0,6. \quad (2.18)$$

### 2.1.6 TESSERACT-OCR

Considerado o estado da arte da área de OCR, o Tesseract é o resultado de dez anos, de 1984 à 1994, de pesquisa e desenvolvimento realizados pela empresa HP. Sua elaboração iniciou-se nos laboratórios da HP em Bristol, como fruto de um projeto de doutorado cujo objetivo era adicionar funcionalidades aos *scanners* de mesa da HP (SMITH, 2007). A eficiência do Tesseract foi provada em 1995 quando participou da quarta competição anual de acurácia de OCR da Universidade de Nevada e apresentou os melhores resultados em relação aos outros participantes. Após participar desta competição, o desenvolvimento do Tesseract voltou a ser sigiloso até que, em 2005, a HP o tornou *open source*. O Tesseract é composto por diversos módulos, tais como: OCR, segmentação, classificação, dicionário, entre outros. Seu módulo de segmentação emprega o método de Otsu (SMITH, 2007) (KAE, 2010).

O Tesseract será utilizado neste projeto para comparação de resultados da abordagem proposta, detalhada no Capítulo 3, assim como será avaliada a possibilidade de otimização do Tesseract por meio da troca de seu módulo de segmentação pelo módulo de segmentação proposto neste trabalho, como discutido na Seção 4.1.

## 2.2 SEGMENTAÇÃO DE IMAGEM

Segmentar uma imagem consiste em subdividi-la em suas partes ou objetos constituintes. Idealmente, esse processo deve parar assim que os objetos de interesse tiverem sido isolados (GONZALEZ; WOODS, 2007). As subseções a seguir abordam os métodos de segmentação explorados neste trabalho.

### 2.2.1 MÉTODO DE OTSU

Desenvolvido por Otsu, este é um método automático de segmentação caracterizado por ser não-supervisionado e não-paramétrico (OTSU, 1979). O método original proposto baseia-se no histograma unidimensional e somente nos níveis de cinza da imagem, não considerando informação espacial entre os *pixels*. Desta forma, embora possa não ser adequado para imagens com alto grau de ruído, métodos derivados de duas ou três dimensões foram desenvolvidos a fim de melhorar seu desempenho (CHEN, 2014). O método clássico pode ser resumido em seis passos (GONZALEZ; WOODS, 2007):

1. Computar as componentes normalizadas do histograma da imagem, para  $i = 0, 1, 2, \dots, 0$ ,

$L-1$ , em que  $L$  é o número de níveis de cinza encontrado na imagem. Cada componente pode ser denotado como  $p_i$ .

2. Calcular as somas cumulativas,  $P(k)$ , para  $k=0, 1, 2, \dots, L-1$ , como mostra a Equação 2.19:

$$P(k) = \sum_{i=0}^k p_i. \quad (2.19)$$

3. Em seguida, com base na Equação 2.20, calcular as médias cumulativas,  $m(k)$ , para  $k=0, 1, 2, \dots, L-1$ :

$$m(k) = \sum_{i=0}^k (i \times p_i). \quad (2.20)$$

4. Computar a média global de intensidade  $m_G$ , empregando a Equação 2.21:

$$m_G = \sum_{i=0}^{L-1} (i \times p_i). \quad (2.21)$$

5. Através da Equação 2.22, calcular a variância entre-classes  $\sigma_B^2(k)$ , para  $k=0, 1, 2, \dots, L-1$ :

$$\sigma_B^2(k) = \frac{[m_G \times P(k) - m(k)]^2}{P(k) \times [1 - P(k)]}. \quad (2.22)$$

6. Obter, por meio da Equação 2.23, o valor de  $k^*$ , sendo este o valor de  $k$  para ao qual  $\sigma_B^2(k)$  é máximo. Caso este valor não seja único,  $k^*$  pode ser computado a partir da média aritmética dos valores de  $k$  correspondentes aos vários máximos detectados:

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k). \quad (2.23)$$

Assim, a imagem  $g(x,y)$  limiarizada é dada pela Equação 2.24:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) > k^*, \\ 0, & \text{se } f(x,y) \leq k^*. \end{cases} \quad (2.24)$$

## 2.2.2 MÉTODO DE NIBLACK

O algoritmo de Niblack utiliza uma janela deslizante para calcular o limiar desejado,  $T$ , sendo, portanto, um método de segmentação local. Seu cálculo baseia-se na



média local,  $m$ , e no desvio padrão,  $s$ , de todos os *pixels* da janela, sendo  $p_i$  o nível de cinza na  $i$ -ésima posição desta. A Equação 2.25 rege este comportamento:

$$\begin{aligned}
 T_{niblack} &= m + k \times s \\
 T_{niblack} &= m + k \sqrt{\frac{1}{NP} \sum (p_i - m)^2} \\
 &= m + k \sqrt{\frac{\sum p_i^2}{NP} - m^2},
 \end{aligned} \tag{2.25}$$

em que  $NP$  é o total de *pixels* na janela e  $k$  é um valor definido pelo usuário que recebe apenas números negativos (SAUVOLA; PIEIKAINEN, 2000) (KHURSHID, 2009).

### 2.2.3 MÉTODO DE SAUVOLA

Proposto por Sauvola e Pieikainen (2000), este método de segmentação realiza uma mudança no algoritmo de Niblack, aplicando uma taxa dinâmica de desvio padrão,  $R$ . A média local é utilizada para multiplicar o termo  $R$ , além de se fixar o valor  $k$ , como apresentado na Equação 2.26, em que  $m$  e  $s$  são, respectivamente, a média local e o desvio padrão da janela deslizante sendo considerada para calcular o limiar. Sauvola e Pieikainen (2000) sugerem que para otimizar os resultados de segmentação  $R$  seja igual à 128 e  $k$  assumo o valor de 0.5:

$$T = m \times \left[ 1 + k \times \left( \frac{s}{R} - 1 \right) \right]. \tag{2.26}$$

### 2.2.4 MÉTODO DE WOLF

Wolf, Jolion e Chassaing (2002) propuseram alterações ao algoritmo de Sauvola que conduziram ao método de segmentação da Equação 2.27, em que  $M$  é o menor nível de cinza encontrado na imagem e  $\alpha = 1 - \frac{s}{R}$ , sendo que  $R$  é o maior valor dos desvios padrões de todas as janelas analisadas, ou seja,  $R = \max(s)$ . O objetivo deste método é normalizar o contraste e o nível médio de cinza presentes em uma imagem, fatores que podem influenciar negativamente o método de Sauvola:

$$T = m - k \times \alpha (m - M). \tag{2.27}$$

### 2.2.5 MÉTODO SNOOPERTEXT

Baseado no método de *Toggle Mapping*, que consiste em um operador morfológico para o aprimoramento do contraste local e da limiarização, Minetto, Thome, Cord, Leite e Stolfi (2014) propuseram o método denominado **SnooperText**, aplicando um elemento estruturante à toda a imagem e realizando os seguintes passos:

1. Computa-se uma imagem que representa a erosão em níveis de cinza da imagem original,  $\mathbb{I}$ , compreendendo o valor mínimo da vizinhança, e, de forma análoga, uma imagem de dilatação resultante do valor máximo encontrado em cada vizinhança, denominadas como  $\mathbb{B}$  e  $\mathbb{F}$ , respectivamente;
2. Cada *pixel*  $p$  na imagem  $\mathbb{I}$  é delimitado à um conjunto ternário de valores  $\mathbb{D}(p) \in \{0, 1, 2\}$  (fundo, indeterminado e região de interesse, respectivamente), com base na Equação 2.28, em que  $c_{min}$  e  $c_{med}$  são dois valores de limiar definidos pelo usuário:

$$\mathbb{D}(p) = \begin{cases} 0, & \text{se } \frac{|\mathbb{I}(p) - \mathbb{B}(p)|}{|\mathbb{F}(p) - \mathbb{B}(p)|} < c_{med}, \\ 1, & \text{se } |\mathbb{F}(p) - \mathbb{B}(p)| < c_{min}, \\ 2, & \text{se } \frac{|\mathbb{I}(p) - \mathbb{B}(p)|}{|\mathbb{F}(p) - \mathbb{B}(p)|} \geq c_{med}. \end{cases} \quad (2.28)$$

### 2.3 APRENDIZADO DE MÁQUINA

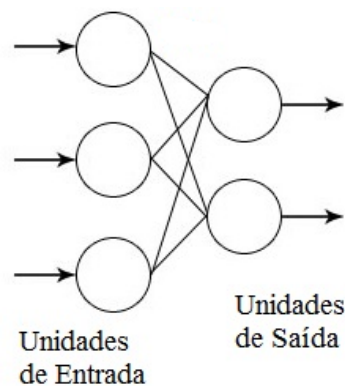
O campo de aprendizado de máquina baseia-se na ideia que os algoritmos e sistemas podem aperfeiçoar seu desempenho com o tempo. Com os avanços em hardware e software, uma variedade de aplicações está emergindo. Uma das áreas de aplicação é a de visão computacional, uma vez que esta apresenta problemas desafiadores e ricos ambientes para o avanço do estado da arte do aprendizado de máquina, permitindo-se o desenvolvimento de algoritmos robustos e flexíveis para sistemas reais de visão computacional (SEBE, 2006).

De forma geral, as técnicas de aprendizado de máquina fornecem um meio de generalizar, verificar e modificar hipóteses, a partir de uma etapa denominada treinamento (SEBE, 2006). Um dos métodos para realizar esse processo é definido como aprendizagem supervisionada, onde o objetivo principal é aprender a mapear um conjunto de padrões de entrada para um conjunto de padrões de saída, sendo estes fornecidos por um supervisor (ALPAYDIN, 2004).

As Subseções 2.3.1 a 2.3.5 apresentam as técnicas de aprendizado de máquina utilizadas neste trabalho. O seu processo de treinamento consiste em apresentar o descritor de cada imagem que contém um caractere e associar qual caractere é representado. Esses são os padrões de entrada e saída, respectivamente.

### 2.3.1 REDES NEURAIS ARTIFICIAIS

Um neurônio pode ser definido como uma célula cerebral capaz de coletar, processar e disseminar sinais elétricos. Contudo, acredita-se que a grande capacidade cognitiva advém não dos neurônios isoladamente, mas sim das redes que formam. Por essa razão, trabalhos iniciais na área de inteligência artificial tentaram criar essas redes neurais artificiais. Na Figura 5, apresenta-se o tipo de rede definida como *perceptron* de camada única (RUSSELL; NORVIG, 2013).

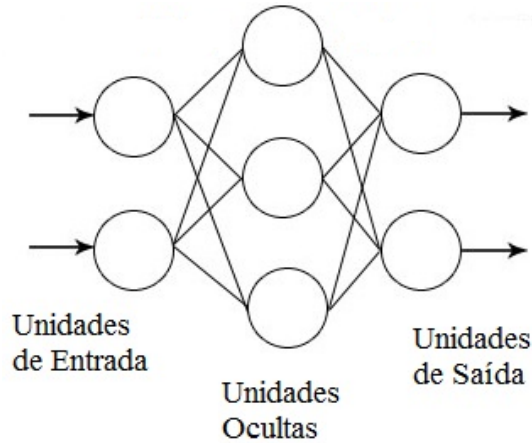


**Figura 5:** Rede neural do tipo *perceptron*.  
Fonte: Adaptado de (NEUROPH, 2014).

Uma extensão da rede *perceptron* é a *perceptron* multi-camadas, que inclui camadas intermediárias (ocultas) a fim de aumentar o espaço de hipóteses que esta pode representar, como exemplificado na Figura 6. Cada unidade oculta pode ser considerada como uma rede *perceptron* que representa uma função de limiar intermediária no espaço de entrada. A saída pode ser visualizada como uma combinação linear de limiares intermediários de várias dessas funções.

### 2.3.2 MÁQUINA DE VETOR SUPORTE

Atualmente, a SVM é uma das técnicas de aprendizagem supervisionada mais populares. Algumas das características que a tornam um ótimo classificador são (RUSSELL; NORVIG, 2013): (a) tende a generalizar com precisão graças à construção de



**Figura 6:** *Perceptron* multi-camadas.  
Fonte: Adaptado de (NEUROPH, 2014).

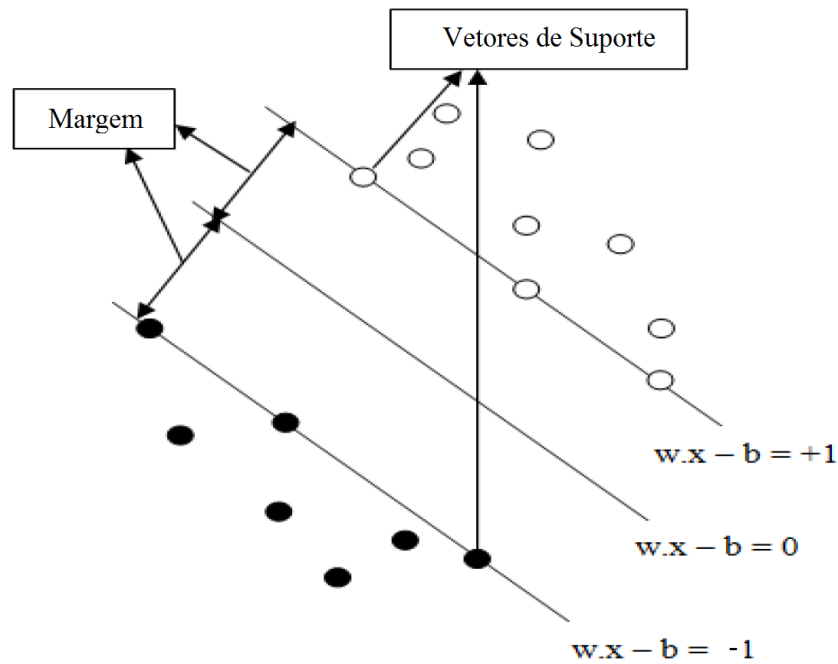
um separador de margem máxima; (b) é mais resistente ao problema de superadaptação em consequência de ser um método não paramétrico, ou seja, ela mantém exemplos de treinamento.

A modelagem da SVM pode ser analisada com base no exemplo ilustrado na Figura 7, na qual é possível identificar duas classes distintas (representadas pelos círculos) e cujo objetivo é separá-las. O modelo consiste em três segmentos de hiperplanos. O segmento  $w \cdot x - b = 0$  é definido como margem de separação ou hiperplano marginal. As retas  $w \cdot x - b = 1$  e  $w \cdot x - b = -1$  são as retas de cada um dos lados do hiperplano marginal. Esses três segmentos juntos constituem um hiperplano que separa as classes, e os padrões que estão sobre os segmentos de retas das extremidades são denominados de vetores de suporte. Quanto maior o valor da margem ou da distância perpendicular, melhor será o desempenho de classificação da SVM (PRADHAN, 2012).

Existem dois tipos diferentes de padrões separáveis: linear e não-linear. Para os padrões lineares, a SVM tentará construir um hiperplano ótimo que permita a separação linear destes padrões, o qual é selecionado de tal maneira que maximize a distância deste para com os padrões mais próximos, com isso, maximizando sua margem, como mostra a Figura 8. A Equação 2.29 define a representação matemática do hiperplano:

$$aX + bY = C. \quad (2.29)$$

Para os padrões não-lineares, é necessário mapeá-los para um espaço de dimensões maior por meio de uma função de *kernel*,  $\Phi(x)$ , como mostra a Equação 2.30:



**Figura 7:** Modelo SVM.  
 Fonte: Adaptado de (PRADHAN, 2012).

$$x \rightarrow \Phi(x). \quad (2.30)$$

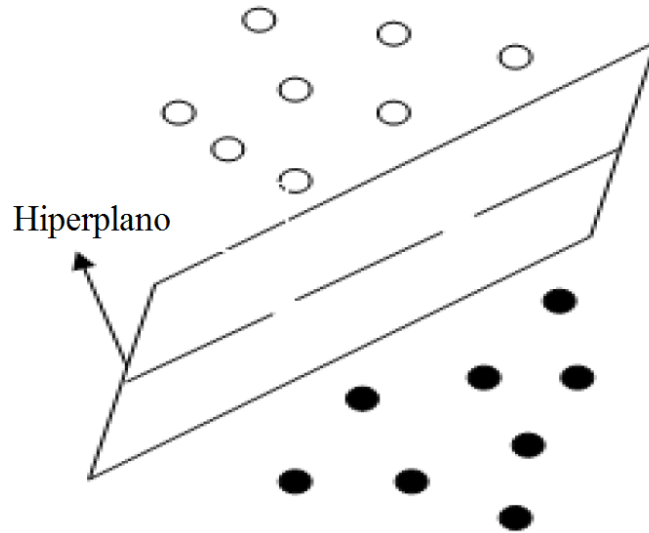
Os tipos de funções de *kernel* incluem: Linear, Polinomial (do inglês, *Poly*), Funções de Base Radial (do inglês, *Radial Basis Function* - RBF) e Sigmoide (do inglês, *Sigmoid*). Existe um parâmetro denominado de complexidade e abreviado como  $\mathbf{C}$ , que é a soma das distâncias de todos os pontos que estão do lado errado do hiperplano. Em outras palavras, ele define a quantidade de erros que podem ser ignorados durante o processo de classificação (PRADHAN, 2012). As configurações utilizadas neste trabalho serão apresentadas no próximo capítulo.

### 2.3.3 CLASSIFICADOR NAÏVE BAYES

Um classificador Bayes é definido por uma função  $f_b(E)$ , como mostra a Equação 2.31:

$$f_b(E) = \frac{p(C = +|E)}{p(C = -|E)} \geq 1, \quad (2.31)$$

em que  $E$  é um exemplo de padrão de entrada para o classificador e representa uma tupla de valores de atributos na forma  $(x_1, x_2, \dots, x_n)$ , sendo que  $x_i$  é o valor do atributo  $X_i$ . Para



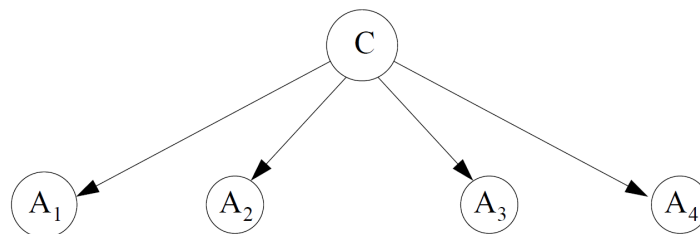
**Figura 8:** Hiperplano ótimo gerado pela SVM para a separação linear.  
Fonte: Adaptado de (PRADHAN, 2012).

cada padrão de entrada existe uma classe correspondente,  $C$ , cujo valor,  $c$ , pode assumir apenas dois valores distintos neste exemplo:  $+$  e  $-$ . Logo, a probabilidade do exemplo  $E$  ser da classe  $c$  é dada pela Equação 2.32 (ZHANG, 2004):

$$p(c|E) = \frac{p(E|c) \times p(c)}{p(E)}. \quad (2.32)$$

em que  $p(c|E)$  é a probabilidade da classe  $C$  ser  $c$  dado o padrão de entrada  $E$ .

Se todos os valores dos atributos do exemplo de padrão de entrada,  $E$ , forem independentes dado a classe, têm-se um classificador naïve Bayes. A Figura 9 apresenta um exemplo deste tipo de classificador, onde é possível observar que todos os atributos  $A_i$ , para  $i = \{1, 2, 3, 4\}$ , não possuem nenhuma outra dependência a não ser o nó da classe,  $C$  (ZHANG, 2004).



**Figura 9:** Exemplo de um classificador naïve de Bayes.  
Fonte: (ZHANG, 2004).

### 2.3.4 K-VIZINHOS MAIS PRÓXIMOS

A regra do Vizinho Mais Próximo (do inglês, *Nearest Neighbor* - NN) associa à dados não conhecidos a classe do seu vizinho mais próximo. Cover e Hart (1967) propuseram a utilização de um valor  $K$  para especificar o número de vizinhos que seriam considerados para definir a classe de uma amostra. Esta técnica ficou conhecida como K-Vizinhos Mais Próximos (do inglês, *K-Nearest Neighbors* - KNN) (BHATIA; VANDANA, 2010).

O KNN é baseado em uma função de distância que verifica as diferenças (ou similaridades) entre duas instâncias, sendo esta função definida na Equação 2.33:

$$d(x, y) = \sqrt{\prod_{i=1}^n (a_i(x) - a_i(y))^2}, \quad (2.33)$$

em que  $a_i(x)$  denota o valor do  $i$ -ésimo atributo  $A_i$  de  $x$ . A classe e o valor deste atributo são definidos, respectivamente, como  $C$  e  $c$  (JIANG, 2007). Assim, dada uma instância,  $x$ , o KNN atribui a classe mais comum dos  $K$  vizinhos mais próximos a  $x$ , como mostra a Equação 2.34:

$$c(x) = \arg \max_{c \in C} \sum_{i=1}^K \delta(c, c(y_i)), \quad (2.34)$$

em que  $y_1, y_2, \dots, y_K$  são os  $K$  vizinhos mais próximos de  $x$ ,  $K$  é o número de vizinhos definido, e  $\delta(c, c(y_i)) = 1$  se  $c = c(y_i)$ , ou 0 do contrário (JIANG, 2007).

O KNN apresenta vantagens como: aprendizado simples e fácil, treinamento rápido e robustez a ruído de treinamento. Como desvantagens pode-se citar: seus resultados são influenciados pelo valor de  $K$ , complexidade computacional e, em alguns casos, sua execução pode ser lenta (BHATIA; VANDANA, 2010).

### 2.3.5 GRADIENT BOOSTED TREES

O *Gradient Boosted Trees* (GBT) é um modelo de conjuntos de árvores de regressão simples construídos de forma gulosa (FRIEDMAN, 2000) (TEAM, 2015). O método de criação destas árvores se dá por meio de um processo iterativo similar à otimização numérica pelo método do gradiente descendente. Sejam  $y$  e  $x$ , respectivamente, o padrão de entrada e a classe deste padrão. A função de perda de desvio (*Deviance Loss*) utilizada para o problema de classificação proposto no presente trabalho é definida na Equação 2.35:

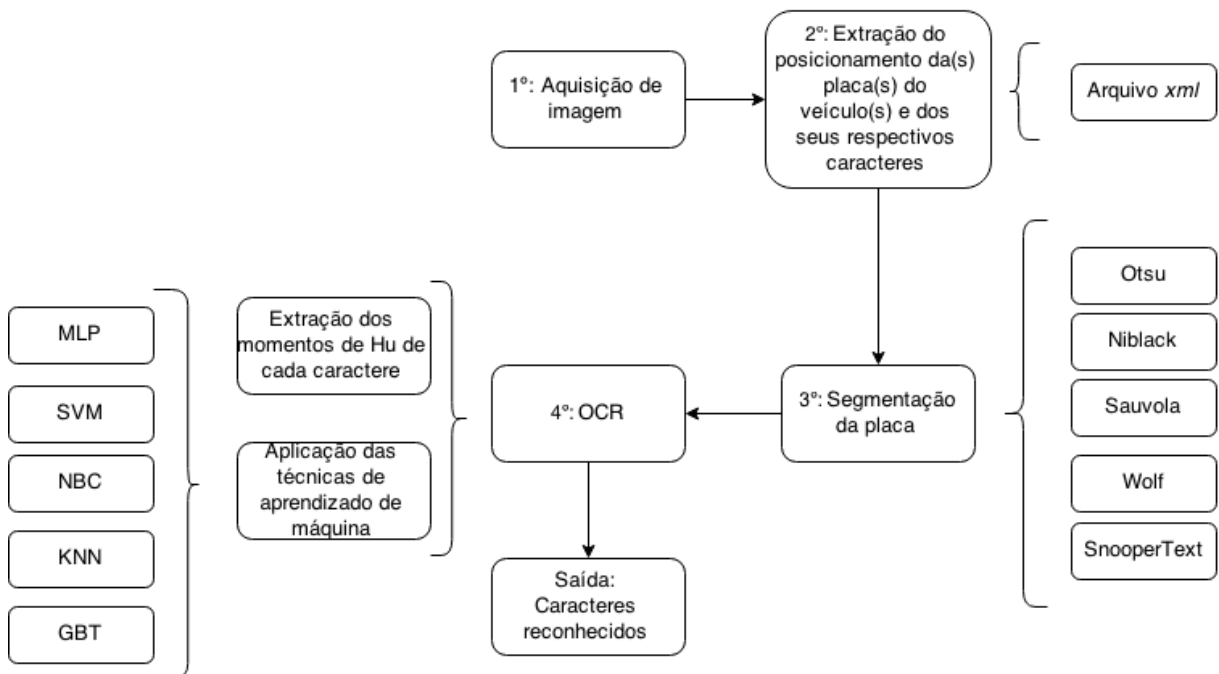
$$L(y, f_1(x), \dots, f_k(x)) = - \sum_{k=0}^K (y = k) \ln p_k(x), \quad (2.35)$$

em que  $L$  é a função de perda em relação ao padrão de entrada  $y$ ,  $f_i(x)$  para  $i = \{1, \dots, k\}$  é uma função de estimativa que melhor mapeia  $x$  para  $y$  e  $p_k(x) = \frac{\exp f_k(x)}{\sum_{i=1}^K \exp f_i(x)}$  é a probabilidade estimada para  $y = k$ .



### 3 DESENVOLVIMENTO

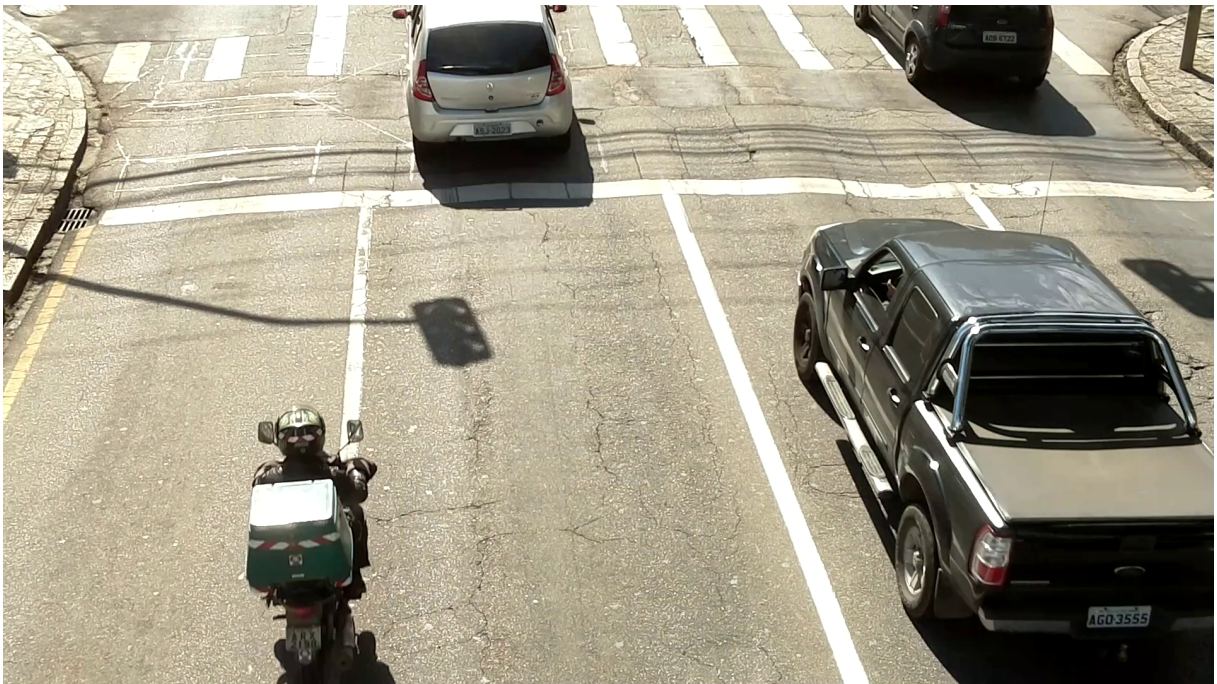
Este capítulo aborda todas as etapas realizadas para cumprir os objetivos do trabalho, o qual pode ser dividido em quatro módulos, representados pelo diagrama da Figura 10. O primeiro consiste na aquisição de imagens e é discutido na Seção 3.1. O módulo referente à extração do posicionamento da(s) placa(s) do veículo(s) e dos seus respectivos caracteres é abordado na Seção 3.2, a qual explica como criou-se a base de dados do trabalho. Em seguida discute-se, na Seção 3.3, o terceiro módulo: segmentação da placa, o qual consiste da codificação dos métodos apresentados na Seção 2.2. O quarto e último módulo, OCR, tem seu desenvolvimento apresentado na Seção 3.4, com base nas técnicas de aprendizado de máquinas discutidas na Seção 2.3.



**Figura 10:** Diagrama de blocos representando os módulos do trabalho.  
Fonte: Autoria própria.

### 3.1 AQUISIÇÃO DE IMAGEM

Neste módulo utilizou-se vídeos provenientes de filmagens reais do tráfego urbano, realizadas como parte de um projeto de mestrado cujo foco é estimar a velocidade dos carros transitando (LUVIZON, 2014). Os mesmos vídeos foram utilizados visando-se avaliar a possibilidade de criação de um sistema que unificasse funções de estimativa de velocidade e ANPR. A partir da análise de cerca de 50 mil imagens (quadros sequenciais dos vídeos utilizados), foram selecionadas 160 que continham placas de identificação diferentes e que apresentavam nitidez suficiente para a diferenciação dos caracteres. As Figuras 11, 12 e 13 apresentam três diferentes situações: (a) na primeira (Figura 11) percebe-se que a placa de identificação da motocicleta não está em condições de uso, pois apresenta baixa nitidez (“borrada”); (b) a próxima situação (Figura 13) mostra que, embora a nitidez da imagem não seja total, é possível diferenciar todos os caracteres contidos na placa da motocicleta; (c) a última situação (Figura 13) apresenta uma placa que está em ótima condição de uso, ao apresentar um nível alto de nitidez.



**Figura 11:** Imagem não selecionada devido à baixa nitidez em relação à motocicleta.  
Fonte: Autoria própria.



**Figura 12:** Imagem em condições aceitáveis de uso.  
Fonte: Autoria própria.



**Figura 13:** Imagem em ótimas condições de uso.  
Fonte: Autoria própria.

### 3.2 EXTRAÇÃO DO POSICIONAMENTO DA PLACA DE IDENTIFICAÇÃO DO VEÍCULO E DOS SEUS RESPECTIVOS CARACTERES

Para reduzir o escopo do trabalho, optou-se por não automatizar o processo de extração da placa de identificação dos veículos, o qual apresenta um série de dificuldades, como: (a) a imagem pode conter mais de uma placa, como na Figura 11; (b) padrões de texto podem ser encontrados na imagem sem que estes sejam de uma placa de identificação; (c) algumas placas de identificação podem ser personalizadas; entre outros. Assim sendo, este processo foi feito manualmente com as 160 imagens selecionadas. Com isso, criou-se uma base de dados com informações sobre a placa e seus caracteres. Estas informações estão inseridas em um arquivo *xml* na forma disposta no Algoritmo 3.1, que representa o exemplo da Figura 14.

**Algoritmo 3.1:** Exemplo de informações de uma placa presente no arquivo *xml*.

Fonte: A autoria própria.

---

```

<image>
<imageName>AHS5667</imageName>
<resolution x="1920" y="1080" />
<plate xmin="1590.0" ymin="938.0" xmax="1691.0" ymax="968.0"
  text="AHS5667" rotation="0.0"/>
<taggedRectangles>
  <taggedRectangle xmin="1591.0" ymin="940.0" xmax="1605.0" ymax="968.0"
    text="A" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1605.0" ymin="940.0" xmax="1623.0" ymax="966.0"
    text="H" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1623.0" ymin="942.0" xmax="1637.0" ymax="963.0"
    text="S" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1637.0" ymin="941.0" xmax="1653.0" ymax="964.0"
    text="5" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1654.0" ymin="940.0" xmax="1666.0" ymax="964.0"
    text="6" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1667.0" ymin="942.0" xmax="1678.0" ymax="964.0"
    text="6" rotation="0.0" offset="0.0" />
  <taggedRectangle xmin="1677.0" ymin="940.0" xmax="1690.0" ymax="963.0"
    text="7" rotation="0.0" offset="0.0" />
</taggedRectangles>
</image>

```

---

Como pode ser observado, estão presentes informações tais como nome, dimensões e coordenadas espaciais da imagem, bem como o texto, tanto da placa inteira quanto dos caracteres isoladamente.



**Figura 14:** Imagem de exemplo analisada para a extração de informações quanto à posição da placa e dos caracteres da mesma.

Fonte: Autoria própria.

### 3.3 MÓDULO DE SEGMENTAÇÃO

O módulo de segmentação engloba os métodos apresentados na Seção 2.2, os quais foram codificados conforme descrito nas Subseções 3.3.1 - 3.3.3, na linguagem de programação C++.

#### 3.3.1 CODIFICAÇÃO DO MÉTODO DE SEGMENTAÇÃO DE OTSU

A codificação do método de segmentação de Otsu baseia-se nos seis passos discutidos na Subseção 2.2.1. A Equação 2.19 é codificada como mostra o Algoritmo 3.2.

---

**Algoritmo 3.2:** Algoritmo para o cálculo das somas cumulativas.

---

```
double cumulative_sums [ SIZE_MAX_HISTOGRAM ] = { 0.0 };
for ( int k = 0; k < max_gray_value; k++ )
```

```

{
    for ( int i = 0; i < k; i++ )
    {
        cumulative_sums [ k ] += normalized_histogram [ i ];
    }
}

```

---

De forma análoga, as Equações 2.20 e 2.21 são codificadas como exposto nos Algoritmos 3.3 e 3.4.

**Algoritmo 3.3:** Algoritmo para o cálculo das médias cumulativas.

---

```

double cumulative_means [ SIZE_MAX_HISTOGRAM ] = { 0.0 };
for ( int k = 0; k < max_gray_value; k++ )
{
    for ( int i = 0; i < k; i++ )
    {
        cumulative_means [ k ] += ( i * normalized_histogram [ i ] );
    }
}

```

---

**Algoritmo 3.4:** Algoritmo para o cálculo da média global de intensidade.

---

```

double global_mean = 0.0;
for ( int i = 0; i < max_gray_value; i++ )
{
    global_mean += ( i * normalized_histogram [ i ] );
}

```

---

Os dois últimos passos para determinar o valor ideal para  $k$  são computados segundo apresentado no Algoritmo 3.5, que corresponde às Equações 2.22 e 2.23.

**Algoritmo 3.5:** Algoritmo para determinar o valor  $k$  ideal.

---

```

for ( int k = 0; k < max_gray_value; k++ )
{
    double numerator = pow ( global_mean * cumulative_sums [ k ] -
        cumulative_means [ k ] , 2 );
    double denominator = cumulative_sums [ k ] * ( 1 - cumulative_sums [ k ] );

    if ( denominator == 0 )

```

```

        continue;

    between_class_variance [ k ] = numerator / denominator;

    otsu_thresh [ between_class_variance [ k ] ].push_back ( k );
    if ( between_class_variance [ k ] > max_between_class_variance )
    {
        max_between_class_variance = between_class_variance [ k ];
    }
}
double auxSum = 0.0;
for ( int i = 0; i < otsu_thresh [ max_between_class_variance ].size (); i++ )
{
    auxSum += otsu_thresh [ max_between_class_variance ].at ( i );
}
k_star = auxSum / otsu_thresh [ max_between_class_variance ].size ();

```

---

### 3.3.2 CODIFICAÇÃO DOS MÉTODOS DE SEGMENTAÇÃO DE NIBLACK, SAUVOLA E WOLF

Os métodos de segmentação de Niblack, Sauvola e Wolf utilizam propriedades comuns à imagem, que correspondem às matrizes de média local,  $m$ , e/ou desvio padrão,  $s$ , referente ao tamanho da janela. O Algoritmo 3.6 apresenta a resolução para se obter tais matrizes.

---

**Algoritmo 3.6:** Algoritmo para obter as matrizes de média local e desvio padrão.

---

```

for ( int i = window_size_aux; i < nrows - window_size_aux; i++ )
{
    for ( int j = window_size_aux; j < ncols - window_size_aux; j++ )
    {
        double m = 0.0;
        double sum_Pi = 0.0;
        double B = 0.0;
        double s = 0.0;

        for ( int y = i - window_size_aux; y <= i + window_size_aux; y++ )
        {
            for ( int x = j - window_size_aux; x <= j + window_size_aux;

```

```

        x++ )
    {
        m += dataSrc [ y * ncols + x ];
        sum_Pi += pow ( dataSrc [ y * ncols + x ], 2 );
    }
}

if ( dataSrc [ i * ncols + j ] < min_gray_value )
{
    min_gray_value = dataSrc [ i * ncols + j ];
}

m /= (windowSize * windowSize);

B = sum_Pi / (windowSize * windowSize) - pow ( m, 2 );
s = sqrt (B);

if ( s > max_s_value )
{
    max_s_value = s;
}

mean_matrix [ i ][ j ] = m;
standard_desv_matrix [ i ][ j ] = s;
}
}

```

---

Os Algoritmos 3.7, 3.8 e 3.9 representam, respectivamente, as Equações 2.25, 2.26 e 2.27.

---

**Algoritmo 3.7:** Algoritmo que representa a equação 2.25.

---

```

threshold_matrix [ i ][ j ] = mean_matrix [ i ][ j ] + k *
    standard_desv_matrix [ i ][ j ];

```

---



---

**Algoritmo 3.8:** Algoritmo que representa a equação 2.26.

---

```

threshold_matrix [ i ][ j ] =
    mean_matrix [ i ][ j ] * ( 1 + k *
        ( standard_desv_matrix [ i ][ j ] / R - 1 ) );

```



---

**Algoritmo 3.9:** Algoritmo que representa a equação 2.27.

---

```
double alpha = 1 - standard_desv_matrix [ i ] [ j ] / max_s_value;
threshold_matrix [ i ] [ j ] = mean_matrix [ i ] [ j ] - k * alpha *
( mean_matrix [ i ] [ j ] - min_gray_value );
```

---

### 3.3.3 CODIFICAÇÃO DO MÉTODO DE SEGMENTAÇÃO *SNOOPERTEXT*

A codificação do método *SnooperText*, apresentado na Seção 2.2.5, aplica a Equação 2.28 para inferir a imagem limiarizada. Tal equação é implementada pelo Algoritmo 3.10.

---

**Algoritmo 3.10:** Algoritmo que representa a equação 2.28.

---

```
for ( int p = 0; p < size_data; p++ )
{
    if ( abs ( F [ p ] - B [ p ] ) < contrast )
    {
        *dataDest++ = INDETERMINATE_THRESHOLD;
    }
    else
    {
        if ( ( F [ p ] - dataSrc [ p ] ) < percentage * ( dataSrc [ p ] - B [ p ] ) / 100 )
        {
            *dataDest++ = BLACK_THRESHOLD;
        }
        else
        {
            *dataDest++ = WHITE_THRESHOLD;
        }
    }
}
}
```

---

### 3.3.4 ANÁLISE DOS RESULTADOS DOS MÉTODOS DE SEGMENTAÇÃO

A Figura 15 apresenta o resultado de cada método de segmentação para oito diferentes imagens utilizadas no trabalho. É possível observar as variações citadas no Capítulo 1, como inclinação, iluminação e tamanho variado da placa de identificação. Percebe-se que estes fatores influenciam cada método de segmentação de forma diferente, tendo pouca influência em alguns casos, enquanto que em outros o caractere torna-se irreconhecível em consequência deles. Por exemplo, para a placa de identificação “AXT-8491” presente na Figura 15, percebe-se, especialmente para o método de segmentação *SnooperText*, que o caractere “4” está totalmente irreconhecível.



**Figura 15:** Exemplos dos resultados obtidos por cada método de segmentação: (a) imagem original em níveis de cinza; (b) Otsu; (c) Niblack; (d) Sauvola; (e) Wolf e (f) *SnooperText*.

Estes resultados foram obtidos para a configuração exposta na Tabela 1. O método de Otsu, por ser um método global, não necessita de nenhum tipo de parâmetro de configuração. Para os métodos de Niblack, Sauvola e Wolf utilizou-se o mesmo tamanho de janela. Ainda para o método de Sauvola, após realizar-se alguns testes com o algoritmo, foram observados melhores resultados utilizando o valor de  $k = 0,05$  e não o sugerido por Sauvola e Pieikainen (2000), como explicado na Subseção 2.2.3. Para os outros dois métodos, de Niblack e Wolf, utilizou-se os valores de parâmetros propostos pelos próprios autores. O método *SnooperText* teve melhor desempenho quando reduziu-se o tamanho da janela de 15, o mesmo utilizado para os outros métodos, para 11. Seus parâmetros também foram definidos com base nas sugestões dos próprios autores.

**Tabela 1:** Configuração utilizada para os métodos de segmentação.

Otsu	-
Niblack	Tamanho da janela: 15 $k$ : -0,2
Sauvola	Tamanho da janela: 15 $R$ : 128 $k$ : 0,05
Wolf	Tamanho da janela: 15 $k$ : 0,1
<i>SnooperText</i>	Tamanho da janela: 11 $c_{min}$ : 34,00 $c_{med}$ : 89,00

### 3.4 MÓDULO DE OCR

Esta seção apresenta o desenvolvimento do módulo de OCR do trabalho, que abrange a codificação dos momentos de Hu, descritor aplicado para cada imagem de caractere, e de como utilizou-se as técnicas de aprendizado de máquina (discutidas na Seção 2.3) fornecidas pela biblioteca OpenCV.

#### 3.4.1 CODIFICAÇÃO DOS MOMENTOS DE HU

A codificação dos momentos de Hu segue as equações mostradas na Seção 2.1.3. A Equação 2.3 é codificada como mostra o Algoritmo 3.11.

**Algoritmo 3.11:** Algoritmo para obter o momento 2-D de ordem  $(p + q)$  de uma imagem digital.

---

```

double m = 0;
for ( int y = 0; y < nrows; y++ )
{
    for ( int x = 0; x < ncols; x++ )
    {
        m += pow ( x, p ) * pow ( y, q ) * data [ y * ncols + x ];
    }
}
return m;

```

---

O Algoritmo 3.12 refere-se à Equação 2.4, em que  $x_{ref}$  e  $y_{ref}$  são variáveis equivalentes à  $\bar{x}$  e  $\bar{y}$  desta mesma equação.

---

**Algoritmo 3.12:** Algoritmo para obter o momento central de ordem  $(p + q)$  de uma imagem.

---

```
double m = 0;
for ( int y = 0; y < nrows; y++ )
{
    for ( int x = 0; x < ncols; x++ )
    {
        m += pow(x - x_ref, p)*pow(y - y_ref, q)*data[ y * ncols + x ];
    }
}
return m;
```

---

Para se obter o momento central normalizado, definido na Equação 2.7, realizou-se as etapas mostradas no Algoritmo 3.13.

---

**Algoritmo 3.13:** Momentos centrais normalizados.

---

```
n20 = u20 / pow ( u00, ( 2 + 0 ) / 2.0 + 1 );
n11 = u11 / pow ( u00, ( 1 + 1 ) / 2.0 + 1 );
n02 = u02 / pow ( u00, ( 0 + 2 ) / 2.0 + 1 );
n30 = u30 / pow ( u00, ( 3 + 0 ) / 2.0 + 1 );
n21 = u21 / pow ( u00, ( 2 + 1 ) / 2.0 + 1 );
n12 = u12 / pow ( u00, ( 1 + 2 ) / 2.0 + 1 );
n03 = u03 / pow ( u00, ( 0 + 3 ) / 2.0 + 1 );
```

---

Por fim, os sete momentos invariantes de Hu são determinados por meio do Algoritmo 3.14, que implementa as Equações 2.8 à 2.14.

---

**Algoritmo 3.14:** Momentos invariantes de Hu.

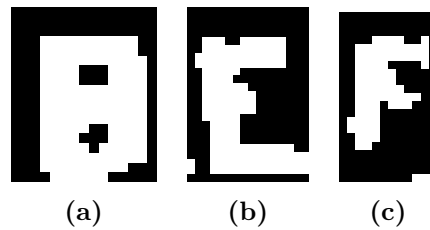
---

```
hu_moments [ 0 ] = n20 + n02;
hu_moments [ 1 ] = pow ( n20 - n02, 2 ) + 4 * pow ( n11, 2 );
hu_moments [ 2 ] = pow ( n30 - 3 * n12, 2 ) + pow ( 3 * n21 - n03, 2 );
hu_moments [ 3 ] = pow ( n30 + n12, 2 ) + pow ( n21 + n03, 2 );
hu_moments [ 4 ] = (n30 - 3*n12) * (n30+n12) * (pow(n30+n12, 2)
                    -3*pow (n21 + n03, 2)) + (3*n21 - n03) * (n21 + n03) *
                    (3 * pow ( n30 + n12, 2) - pow ( n21 + n03, 2 ) );
hu_moments [ 5 ] = (n20 - n02) * (pow ( n30 + n12, 2) - pow(n21 + n03, 2))
```

$$\begin{aligned}
& + 4 * n11 * ( n30 + n12 ) * ( n21 + n03 ); \\
\text{hu\_moments [ 6 ]} & = ( 3 * n21 - n03 ) * ( n30 + n12 ) * ( \text{pow} ( n30 + n12, 2 ) \\
& - 3 * \text{pow} ( n21 + n03, 2 ) ) + ( 3 * n12 - n30 ) * ( n21 + n03 ) * \\
& ( 3 * \text{pow} ( n30 + n12, 2 ) - \text{pow} ( n21 + n03, 2 ) );
\end{aligned}$$


---

Como exemplo, para as Figuras 16 (a), 16 (b) e 16 (c), segmentadas com o método de Otsu, os momentos de Hu são aqueles mostrados na Tabela 2. Cada conjunto de imagens representando o mesmo caractere irá gerar um conjunto de momentos de Hu similares. A união dos conjuntos dos momentos de Hu dos caracteres de “A” a “Z” e/ou de “0” a “9” será utilizada no treinamento das técnicas de aprendizado de máquina, como discutido na próxima subseção.



**Figura 16:** Imagens de exemplo para a extração dos momentos de Hu.

**Tabela 2:** Resultado da extração dos momentos de Hu das Figuras 16 (a) – 16 (c).

Momento	Figura 16 (a)	Figura 16 (b)	Figura 16 (c)
1	0.000758917	0.00124543	0.00126715
2	4.21E-008	3.92E-007	3.80E-007
3	6.35E-012	1.60E-010	4.63E-010
4	1.28E-013	5.26E-011	5.53E-010
5	-7.75E-026	-8.61E-022	2.52E-019
6	-2.54E-017	-2.34E-014	1.09E-013
7	8.63E-026	4.74E-021	1.23E-019

Percebe-se que os momentos de Hu da Figura 16 (a), representando um caractere “B”, apresentam uma notável diferença em relação aos momentos de Hu da Figura 16 (b), cujo caractere representado é um “E”. Contudo, entre os momentos de Hu das Figuras 16 (b) e 16 (c), que representam os caracteres “E” e “F”, respectivamente, há uma diferença menor entre os resultados. Essa observação demonstra que caracteres semelhantes apresentam momentos de Hu cujos valores são mais próximos do que caracteres com grandes variações de forma entre si.

### 3.4.2 TREINAMENTO DAS TÉCNICAS DE APRENDIZADO DE MÁQUINA

Para realizar-se o treinamento das técnicas de aprendizado de máquina foram utilizadas 100 imagens contendo uma placa de automóvel cada, totalizando 700 caracteres, como apresenta a Tabela 3. Dado o padrão de caracteres nas placas de identificação de veículos do Estado do Paraná, cuja série inicial é AAA-0001 e série final é BEZ-9999 (DETRAN-SC, 2012), o caractere “A” é aquele com maior probabilidade de ser encontrado, fato que explica o alto número de caracteres “A” mostrado nesta tabela. Também em razão do padrão de três letras e quatro números das placas, observa-se que numerais possuem uma quantidade mais expressiva quando comparados aos caracteres alfabéticos. O pior dos casos, em termos quantitativos, tanto na etapa de treinamento quanto na análise de desempenho, foi o caractere “K”, possuindo três padrões para o treinamento e apenas um padrão para a análise de desempenho. Dada essa particularidade, o nível de revocação deste padrão, discutido no Capítulo 4, apresenta valores extremos como 0% ou 100%.

**Tabela 3:** Disposição dos caracteres utilizados para o treinamento das técnicas de aprendizado de máquina.

Caractere	Quantidade	Caractere	Quantidade	Caractere	Quantidade
0	42	C	7	O	6
1	38	D	9	P	5
2	38	E	5	Q	6
3	39	F	7	R	13
4	29	G	8	S	10
5	34	H	6	T	12
6	46	I	9	U	10
7	46	J	8	V	10
8	45	K	3	W	9
9	43	L	7	X	14
A	87	M	11	Y	11
B	14	N	5	Z	8

A integração entre os métodos de segmentação apresentados na Seção 2.2 e as técnicas de aprendizado de máquina discutidas na Seção 2.3 é feita de forma que para cada método de segmentação avalia-se o desempenho de todas as técnicas de aprendizado de máquina. Desta forma, existem 25 combinações possíveis de métodos de segmentação e técnicas de aprendizado de máquina para serem avaliadas.

O procedimento para realizar o treinamento de cada técnica de aprendizado de máquina consiste de: (a) segmentar todas as imagens de caracteres; (b) extrair os momentos de Hu das imagens segmentadas; (c) processar os padrões de entrada e saída, ou

seja, realizar o mapeamento dos momentos de  $H_u$  para a respectiva classe (o caractere ao qual o conjunto de momentos refere-se) e (d) apresentar os padrões de entrada e saída para as técnicas de aprendizado de máquina.

A Tabela 4 apresenta a configuração aplicada às técnicas de aprendizado de máquinas da Seção 2.3. Cada um destes parâmetros é discutido em mais detalhes na Seção 2.3.

**Tabela 4:** Configuração utilizada para as técnicas de aprendizado de máquina.

MLP	Número de camadas ocultas: 1 Função de ativação: Gaussiana Número de neurônios por camada oculta : 15
SVM	Função de <i>kernel</i> : RBF $C$ : 3
NBC	–
KNN	$K$ : 5
GBT	Função de perda: <i>Deviance</i> (De desvio)

## 4 ANÁLISE DE DESEMPENHO E RESULTADOS

Para avaliar o desempenho dos métodos de segmentação e técnicas de aprendizado de máquina desenvolvidos neste trabalho, foram utilizadas 60 imagens com uma placa de automóvel cada, totalizando 420 caracteres, seguindo a disposição exposta na Tabela 5.

**Tabela 5:** Disposição dos caracteres utilizados para a comparação entre os métodos de segmentação e as técnicas de aprendizado de máquina.

Caractere	Quantidade	Caractere	Quantidade	Caractere	Quantidade
0	22	C	5	O	3
1	30	D	9	P	3
2	32	E	11	Q	7
3	26	F	2	R	6
4	21	G	4	S	6
5	24	H	2	T	5
6	25	I	5	U	3
7	18	J	4	V	6
8	17	K	1	W	7
9	25	L	8	X	9
A	46	M	5	Y	2
B	15	N	5	Z	1

Como explicado na Subseção 3.4.2, os caracteres que possuem apenas um padrão para ser apresentado, como é o caso dos caracteres “K” e “Z”, podem apresentar resultados extremos de precisão e, especialmente, de revocação, como 0% ou 100% em função desta particularidade.

A Tabela 6 apresenta quais métodos de segmentação e técnica de aprendizado de máquina obtiveram o melhor resultado em relação à precisão, assim como a respectiva revocação. Esta tabela é uma compilação dos melhores resultados das Tabelas 12 à 16 do Apêndice A. A coluna “Quantidade” informa os mesmos valores apresentados na Tabela 5 e foi inserida exclusivamente nesta tabela com o propósito de facilitar o entendimento dos resultados de precisão e revocação. A análise da Tabela 6 mostra que o método de



segmentação de Wolf e a técnica de aprendizado de máquina GBT estão presentes em, respectivamente, 11 e 13 dos resultados, sendo que em seis vezes aparecem em conjunto. A partir dessa observação, conclui-se que estes geraram os melhores resultados. Apesar disto, observa-se vários caracteres com um baixíssimo nível de precisão: “C” (cinco caracteres), “H” (dois caracteres), “K” (um caractere) e “R” (seis caracteres). A precisão avalia tanto o número de itens corretos quanto incorretos (Subseção 2.1.5), estes resultados são consequência de um alto número de caracteres reconhecidos de forma errada. Observa-se também que existem três caracteres, F, G e Z, que obtiveram 0% de revocação, ou seja, nenhum deles foi reconhecido. Como explicado no começo deste capítulo, este resultado pode ser consequência do baixo número de amostras. Contudo, há ainda a possibilidade do problema explicado na Subseção 2.1.4, que refere-se à dificuldade inerente de um OCR diferenciar alguns caracteres alfabéticos de numerais. Visando determinar se estes casos poderiam ser reduzidos, realizou-se novamente o processo de treinamento e avaliação considerando dois conjuntos distintos de caracteres: alfabéticos, de “A” a “Z”, e numéricos, de “0” a “9”. Os resultados são apresentados nas Tabelas 7 e 8.

A partir destas tabelas, percebe-se que houve redução de três para dois os casos de caracteres com 0% de revocação, uma vez que ao menos um caractere “G” foi reconhecido. Esta avaliação sugere que o caractere “G” possivelmente estava sendo confundido com o caractere “6”. Da Tabela 7, observa-se que o método de segmentação de Wolf e a técnica de aprendizado de máquina GBT obtiveram novamente a maior frequência de melhores resultados. Para o conjunto numérico avaliado na Tabela 8, a técnica de aprendizado de máquina KNN superou o GBT em termos de número de vezes com melhor precisão. Apesar disto, a avaliação conjunta das Tabelas 7 e 8 continua mostrando que o método de segmentação de Wolf e a técnica de aprendizado de máquina GBT são os que geram melhores resultados. De forma geral, a precisão da maioria dos caracteres aumentou, como é o caso dos caracteres “C”, “H”, “K” e “R” citados na discussão da Tabela 6.

#### 4.1 COMPARAÇÃO COM O TESSERACT

A comparação com os resultados do método estado da arte denominado Tesseract visa determinar o nível de eficácia da abordagem proposta, discutida em detalhes no Capítulo 3. Verifica-se também a possibilidade de se otimizar os resultados deste método ao integrá-lo ao módulo de segmentação proposto neste trabalho. Assim sendo, deste ponto em diante do texto, será utilizada a seguinte terminologia para diferenciar o Tesseract em função de seus módulos:

**Tabela 6:** Compilação dos melhores resultados em relação à precisão para todos os caracteres.

Caractere	Quantidade	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
0	22	13,85	40,91	OTSU	KNN
1	30	20,97	43,33	OTSU	KNN
2	32	13,04	18,75	SAUVOLA	KNN
3	26	23,08	11,54	WOLF	GBT
4	21	100,00	4,76	WOLF	SVM
5	24	100,00	4,17	WOLF	MLP
6	25	16,67	8,00	OTSU	GBT
7	18	26,67	22,22	OTSU	GBT
8	17	25,00	5,88	SAUVOLA	GBT
9	25	14,29	8,00	NIBLACK	MLP
A	46	44,44	8,70	OTSU	SVM
B	15	12,50	6,67	WOLF	GBT
C	5	2,94	20,00	SNOOPERTEXT	SVM
D	9	15,38	22,22	WOLF	NBC
E	11	100,00	9,09	SAUVOLA	KNN
F	2	0,00	0,00	–	–
G	4	0,00	0,00	–	–
H	2	1,04	50,00	SNOOPERTEXT	NBC
I	5	9,09	20,00	NIBLACK	MLP
J	4	16,67	25,00	OTSU	GBT
K	1	1,89	100,00	NIBLACK	NBC
L	8	25,00	12,50	OTSU	SVM
M	5	100,00	20,00	WOLF	MLP
N	5	25,00	20,00	WOLF	GBT
O	3	25,00	33,33	NIBLACK	GBT
P	3	9,09	33,33	WOLF	NBC
Q	7	25,00	14,29	WOLF	GBT
R	6	2,00	16,67	SAUVOLA	SVM
S	6	7,69	16,67	OTSU	GBT
T	5	16,67	20,00	SNOOPERTEXT	GBT
U	3	8,33	33,33	WOLF	GBT
V	6	16,67	16,67	WOLF	GBT
W	7	9,09	14,29	NIBLACK	MLP
X	9	14,29	11,11	NIBLACK	MLP
Y	2	10,00	50,00	NIBLACK	MLP
Z	1	0,00	0,00	–	–

**Tabela 7:** Compilação dos melhores resultados em relação à precisão para o conjunto de caracteres alfabéticos.

Caractere	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
A	60,00	13,04	WOLF	NBC
B	30,00	40,00	SNOOPERTEXT	KNN
C	33,33	20,00	WOLF	GBT
D	40,00	22,22	SAUVOLA	NBC
E	100,00	18,18	NIBLACK	KNN
F	0,00	0,00	–	–
G	16,67	25,00	OTSU	MLP
H	100,00	50,00	WOLF	GBT
I	33,33	20,00	OTSU	GBT
J	12,50	25,00	OTSU	GBT
K	4,35	100,00	NIBLACK	NBC
L	14,29	25,00	OTSU	NBC
M	33,33	20,00	OTSU	KNN
N	50,00	20,00	WOLF	GBT
O	50,00	33,33	WOLF	GBT
P	25,00	33,33	WOLF	NBC
Q	66,67	28,57	WOLF	GBT
R	10,00	16,67	WOLF	GBT
S	8,33	16,67	OTSU	GBT
T	14,29	20,00	WOLF	GBT
U	16,67	33,33	SNOOPERTEXT	GBT
V	25,00	16,67	NIBLACK	NBC
W	10,00	14,29	NIBLACK	MLP
X	20,00	11,11	OTSU	GBT
Y	5,88	50,00	NIBLACK	MLP
Z	0,00	0,00	–	–

**Tabela 8:** Compilação dos melhores resultados em relação à precisão para o conjunto de caracteres numéricos.

Caractere	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
0	20,00	22,73	WOLF	GBT
1	40,00	53,33	OTSU	GBT
2	25,00	15,62	SAUVOLA	KNN
3	37,50	34,62	WOLF	MLP
4	75,00	14,29	NIBLACK	SVM
5	28,57	16,67	WOLF	KNN
6	33,33	8,00	WOLF	SVM
7	20,00	11,11	OTSU	KNN
8	18,75	17,65	OTSU	KNN
9	23,53	16,00	NIBLACK	KNN

- Tesseract completo: diz respeito ao Tesseract sem nenhum tipo de alteração, ou seja, utilizando seus próprios módulos;
- Tesseract alterado: refere-se ao Tesseract cujo módulo de segmentação foi trocado pelo módulo de segmentação deste trabalho, discutido na Seção 3.3.

Avaliou-se o Tesseract em 3 diferentes cenários:

- Cenário 1: Tesseract completo em relação à abordagem proposta;
- Cenário 2: Tesseract alterado em relação à abordagem proposta;
- Cenário 3: Tesseract alterado em relação ao Tesseract completo.

#### 4.1.1 RESULTADOS DO CENÁRIO 1: TESSERACT COMPLETO EM RELAÇÃO À ABORDAGEM PROPOSTA

Neste cenário, avaliou-se o desempenho da abordagem proposta quando comparada ao Tesseract completo. A Tabela 9 apresenta os resultados desta comparação. Utilizando o Tesseract completo, observa-se que este obtém o melhor resultado em 23 dos 36 caracteres avaliados. Em termos percentuais, o Tesseract completo obteve o melhor resultado em 63.89% dos caracteres. Entretanto, percebe-se que alguns caracteres apresentam um baixo valor de revocação, como é o caso dos caracteres “0”, “4”, “5”, “6” e “B”. Uma baixa revocação sugere que o OCR está reconhecendo um número pequeno do total

de padrões de entrada apresentado. Por exemplo, uma revocação de 4,00%, como é o caso do caractere “6” que apresenta 25 padrões de entrada, indica que o Tesseract reconheceu somente um.

#### 4.1.2 RESULTADOS DO CENÁRIO 2: TESSERACT ALTERADO EM RELAÇÃO À ABORDAGEM PROPOSTA

No cenário 2 foi avaliado o desempenho do Tesseract alterado (ou seja, cujo módulo de segmentação foi substituído pelo implementado neste trabalho) em comparação com a abordagem proposta. Pela Tabela 10, em comparação com a Tabela 9, percebe-se que, de forma geral para os caracteres, tanto a precisão quanto a revocação aumentaram. As exceções, para a precisão, são os caracteres “3”, “H” e “N” e, para a revocação, os caracteres “9”, “A”, “L” e “V”.

Para o Cenário 2, os melhores métodos de segmentação foram o de Otsu e Niblack, sendo 16 e 11, respectivamente, o número de vezes que geraram um resultado com a melhor precisão. Em relação ao Cenário 1, o número de caracteres para os quais o Tesseract obteve a maior precisão aumentou de 23 para 29, resultando em um melhor resultado de precisão de 80,56% dos 36 caracteres.

#### 4.1.3 RESULTADOS DO CENÁRIO 3: TESSERACT ALTERADO EM RELAÇÃO AO TESSERACT COMPLETO

O Cenário 3 tem por objetivo averiguar qual dos cinco métodos que compõem o módulo de segmentação proposto no trabalho otimiza o desempenho do Tesseract. Para evitar dúvidas, na terceira coluna (método de segmentação) o termo Tesseract refere-se que o melhor método de segmentação foi o próprio módulo de segmentação do Tesseract.

Pela Tabela 11, percebe-se que o módulo de segmentação do Tesseract obteve os melhores resultados em relação à precisão em apenas quatro dos 36 caracteres: “3”, “E”, “H” e “N”. Para os outros 32 casos, ou seja, 88,89% dos caracteres, o módulo de segmentação proposto neste trabalho obteve resultados similares ou melhores.

O gráfico da Figura 17 exhibe os resultados do Tesseract completo e do Tesseract alterado em função da precisão, tornando possível determinar para quais caracteres o Tesseract alterado obtém uma precisão estritamente maior. O Tesseract alterado apresenta: 22 caracteres com precisão estritamente superior ao Tesseract completo, oito caracteres com precisão igual a este (10, se incluir os caracteres com 0% de revocação) e, como discutido no parágrafo anterior, quatro caracteres cuja precisão é menor do que a do Tesseract

**Tabela 9:** Resultados para o Cenário 1: Tesseract completo em relação à abordagem proposta.

Caractere	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
0	50,00	4,55	TESSERACT	TESSERACT
1	20,97	43,33	OTSU	KNN
2	90,91	62,50	TESSERACT	TESSERACT
3	100,00	57,69	TESSERACT	TESSERACT
4	100,00	4,76	WOLF	SVM
5	100,00	4,17	WOLF	MLP
6	100,00	4,00	TESSERACT	TESSERACT
7	100,00	66,67	TESSERACT	TESSERACT
8	25,00	5,88	SAUVOLA	GBT
9	100,00	16,00	TESSERACT	TESSERACT
A	58,49	67,39	TESSERACT	TESSERACT
B	16,67	6,67	TESSERACT	TESSERACT
C	100,00	40,00	TESSERACT	TESSERACT
D	36,36	44,44	TESSERACT	TESSERACT
E	100,00	9,09	SAUVOLA	KNN
F	0,00	0,00	–	–
G	0,00	0,00	–	–
H	20,00	50,00	TESSERACT	TESSERACT
I	9,09	20,00	NIBLACK	MLP
J	33,33	75,00	TESSERACT	TESSERACT
K	25,00	100,00	TESSERACT	TESSERACT
L	63,64	87,50	TESSERACT	TESSERACT
M	100,00	20,00	WOLF	MLP
N	62,50	100,00	TESSERACT	TESSERACT
O	25,00	33,33	NIBLACK	GBT
P	100,00	66,67	TESSERACT	TESSERACT
Q	25,00	14,29	WOLF	GBT
R	100,00	16,67	TESSERACT	TESSERACT
S	40,00	66,67	TESSERACT	TESSERACT
T	100,00	60,00	TESSERACT	TESSERACT
U	40,00	66,67	TESSERACT	TESSERACT
V	75,00	100,00	TESSERACT	TESSERACT
W	9,09	14,29	NIBLACK	MLP
X	30,77	44,44	TESSERACT	TESSERACT
Y	10,00	50,00	NIBLACK	MLP
Z	14,29	100,00	TESSERACT	TESSERACT

**Tabela 10:** Resultados para o Cenário 2: Tesseract alterado em relação à abordagem proposta.

Caractere	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
0	75,00	13,64	WOLF	TESSERACT
1	20,97	43,33	OTSU	KNN
2	100,00	65,62	OTSU	TESSERACT
3	94,74	69,23	OTSU	TESSERACT
4	100,00	28,57	NIBLACK	TESSERACT
5	100,00	4,17	WOLF	MLP
6	100,00	4,00	OTSU	TESSERACT
7	100,00	72,22	OTSU	TESSERACT
8	100,00	5,88	SNOOPERTEXT	TESSERACT
9	100,00	12,00	SAUVOLA	TESSERACT
A	85,19	50,00	OTSU	TESSERACT
B	37,50	20,00	NIBLACK	TESSERACT
C	100,00	40,00	OTSU	TESSERACT
D	45,45	55,56	SNOOPERTEXT	TESSERACT
E	100,00	9,09	SAUVOLA	KNN
F	100,00	50,00	OTSU	TESSERACT
G	100,00	50,00	NIBLACK	TESSERACT
H	12,50	50,00	OTSU	TESSERACT
I	9,09	20,00	NIBLACK	MLP
J	40,00	100,00	OTSU	TESSERACT
K	50,00	100,00	NIBLACK	TESSERACT
L	75,00	75,00	OTSU	TESSERACT
M	100,00	20,00	WOLF	MLP
N	33,33	100,00	OTSU	TESSERACT
O	25,00	33,33	NIBLACK	GBT
P	100,00	66,67	NIBLACK	TESSERACT
Q	25,00	14,29	OTSU	TESSERACT
R	100,00	100,00	NIBLACK	TESSERACT
S	80,00	66,67	OTSU	TESSERACT
T	100,00	60,00	OTSU	TESSERACT
U	66,67	66,67	NIBLACK	TESSERACT
V	83,33	83,33	SNOOPERTEXT	TESSERACT
W	9,09	14,29	NIBLACK	MLP
X	42,86	66,67	OTSU	TESSERACT
Y	100,00	50,00	NIBLACK	TESSERACT
Z	50,00	100,00	WOLF	TESSERACT

**Tabela 11:** Resultados para o Cenário 3: Tesseract alterado em relação ao Tesseract completo.

Caractere	Precisão (%)	Revocação (%)	Método de segmentação	Técnica de aprendizado de máquina
0	75,00	13,64	WOLF	TESSERACT
1	8,33	3,33	SNOOPERTXT	TESSERACT
2	100,00	65,62	OTSU	TESSERACT
3	100,00	57,69	TESSERACT	TESSERACT
4	100,00	28,57	NIBLACK	TESSERACT
5	84,62	45,83	SNOOPERTXT	TESSERACT
6	100,00	4,00	OTSU	TESSERACT
7	100,00	72,22	OTSU	TESSERACT
8	100,00	5,88	SNOOPERTXT	TESSERACT
9	100,00	16,00	SAUVOLA	TESSERACT
A	85,19	50,00	OTSU	TESSERACT
B	37,50	20,00	NIBLACK	TESSERACT
C	100,00	40,00	OTSU	TESSERACT
D	45,45	55,56	SNOOPERTXT	TESSERACT
E	32,14	81,82	TESSERACT	TESSERACT
F	100,00	50,00	OTSU	TESSERACT
G	100,00	50,00	NIBLACK	TESSERACT
H	20,00	50,00	TESSERACT	TESSERACT
I	2,35	40,00	SNOOPERTXT	TESSERACT
J	40,00	100,00	OTSU	TESSERACT
K	50,00	100,00	NIBLACK	TESSERACT
L	75,00	75,00	OTSU	TESSERACT
M	20,00	20,00	OTSU	TESSERACT
N	62,50	100,00	TESSERACT	TESSERACT
O	0,00	0,00	-	-
P	100,00	66,67	NIBLACK	TESSERACT
Q	25,00	14,29	OTSU	TESSERACT
R	100,00	100,00	NIBLACK	TESSERACT
S	80,00	66,67	OTSU	TESSERACT
T	100,00	60,00	OTSU	TESSERACT
U	66,67	66,67	NIBLACK	TESSERACT
V	83,33	83,33	SNOOPERTXT	TESSERACT
W	0,00	0,00	-	-
X	42,86	66,67	OTSU	TESSERACT
Y	100,00	50,00	NIBLACK	TESSERACT
Z	50,00	100,00	WOLF	TESSERACT

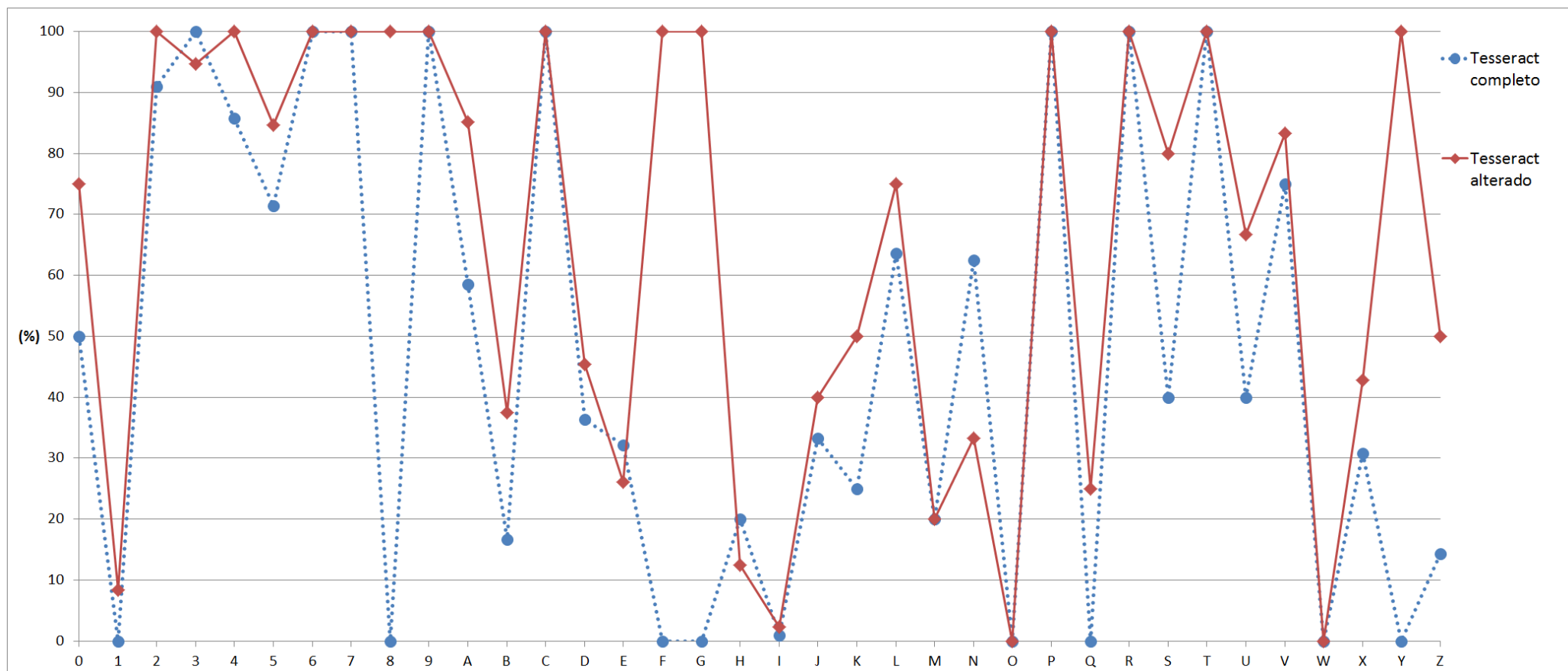


completo.

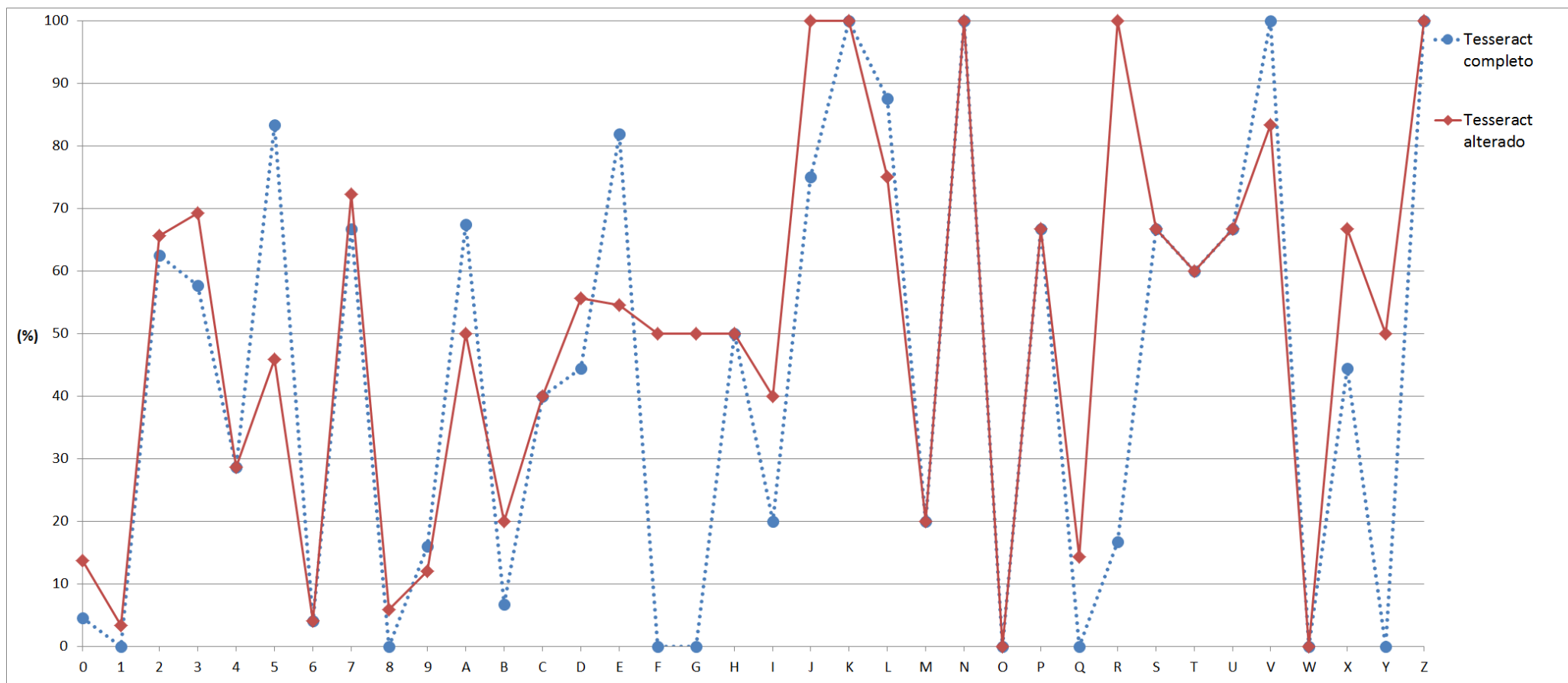
A Figura 18 é análoga à Figura 17 mas em relação à revocação do Tesseract completo e do Tesseract alterado. Observa-se que o Tesseract alterado apresenta: 16 caracteres cujo valor de revocação é estritamente maior em comparação ao respectivo resultado do Tesseract completo, 14 resultados iguais e 6 cujo desempenho foi menor.

Desta forma, percebe-se que o módulo de segmentação proposto neste trabalho conseguiu otimizar os resultados do Tesseract. Por exemplo, para o caso particular do caractere “R”, o Tesseract atingiu o expressivo resultado de 100% de precisão e revocação. Em outras palavras, ele acertou os seis (Tabela 5) caracteres “R” que haviam nos padrões de entrada sem cometer nenhum erro com outro caractere. Outros casos relevantes para análise são os caracteres “8”, “F” e “G”. Percebe-se pela Figura 17 que o caractere “8” apresenta uma diferença notável em relação à precisão, entre o Tesseract completo e o Tesseract alterado. Contudo, ao avaliar a Figura 18 nota-se que o Tesseract completo apresenta uma revocação de 0,00%, logo sua precisão só poderia ser 0,00%. Já para o caso do Tesseract alterado, este possui uma revocação de 5,88%, o que corresponde a um acerto do total de 17 padrões de entrada para esse caractere. Como neste caso o Tesseract alterado identificou apenas um caractere e o reconheceu corretamente, têm-se uma precisão de 100,00%. Esta situação é similar para os caracteres “F” e “G” que apresentam, para o Tesseract completo, uma revocação de 0,00%. Para o Tesseract alterado têm-se, para ambos, uma revocação de 50,00% e uma precisão de 100,00%.

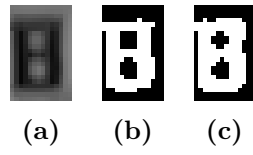
É importante salientar que, apesar do Otsu ser o método de segmentação utilizado pelo Tesseract, detalhes de implementação podem gerar resultados diferentes para este método, como discutido na Seção 5.1. Por exemplo, na Figura 19, (a) representa a imagem original em níveis de cinza do caractere “B”, (b) é a imagem resultante da segmentação pelo método de Otsu implementado neste trabalho, cujo limiar é  $T = 69$  e (c) é o resultado da segmentação do Tesseract, limiarizada com o valor de  $T = 73$ . Embora visualmente a diferença não seja expressiva, computacionalmente o Tesseract reconheceu de forma correta apenas a Figura 19 (b). Com isso, pequenas variações do valor de limiar para cada imagem podem aumentar a probabilidade de acerto do Tesseract.



**Figura 17:** Análise gráfica dos resultados do Cenário 3: Tesseract alterado em relação ao Tesseract completo em função da precisão.  
 Fonte: Autoria Própria.



**Figura 18:** Análise gráfica dos resultados do Cenário 3: Tesseract alterado em relação ao Tesseract completo em função da revocação.  
 Fonte: Autoria Própria.



**Figura 19:** Comparação entre o método de segmentação de Otsu implementado neste trabalho e o do Tesseract.

Fonte: Autoria própria.

## 5 CONSIDERAÇÕES FINAIS

Com este trabalho foi possível verificar na prática os desafios encontrados na área de visão computacional. Reconhecer um caractere, algo trivial para um ser humano, apresenta uma alta complexidade para um sistema computacional.

Todos os objetivos do trabalho foram cumpridos, assim sendo, desenvolveu-se o módulo de segmentação e OCR de um sistema ANPR, além de avaliar o desempenho destes módulos em relação ao Tesseract. Também substituiu-se o módulo de segmentação do Tesseract pelo implementado nesse trabalho e verificou-se uma melhora dos resultados do Tesseract, tanto em relação à precisão quanto à revocação. Implementou-se o descritor momentos de Hu e foi criada uma base de dados de placas de identificação de veículos, que pode ser distribuída de forma a permitir que outros pesquisadores realizem estudos de vários métodos para OCR em um campo específico como é o caso do ANPR.

### 5.1 PROBLEMAS ENCONTRADOS

O primeiro problema encontrado é inerente às técnicas de aprendizado de máquina: sua configuração. Encontrar uma configuração adequada dos vários atributos que interferem nos resultados é uma tarefa árdua, pois deve-se avaliar várias configurações diferentes e não se pode garantir que a configuração final empregada seja ótima.

O segundo problema foi a criação da base de dados. Como a seleção de cada caractere e placa foi feita manualmente, este processo, além de demorado, mostrou-se passível de imperfeições devido ao posicionamento da placa na imagem. Por vezes a placa estava em posições rotacionadas, fazendo com que a região de um caractere ficasse sobreposta à de outro.

## 5.2 TRABALHOS FUTUROS

Para trabalhos futuros sugere-se avaliar o uso de um outro descritor de imagem, tal como os descritores de Fourier, que também são invariantes à translação, rotação e mudança de escala. Propõe-se também automatizar o processo de extração da placa de identificação do veículo. Para isto, problemas como: (a) o fundo da imagem pode conter padrões semelhantes à placas, como números ou letras estampadas no veículo, (b) a placa de identificação existe, mas em diferentes pontos da imagem e (c) uma imagem pode conter mais de uma placa de identificação; devem ser considerados e tratados. Além disso, recomenda-se a utilização de um método de avaliação que pondere precisão e revocação como sendo um único valor de medida como, por exemplo, a *F-Measure*, assim como realizar uma etapa de análise multiresolução das imagens.

## REFERÊNCIAS

- ALPAYDIN, E. **Introduction to Machine Learning**. [S.l.]: The MIT Press, 2004.
- BAGGIO, D. L.; EMAMI, S.; ESCRIVÁ, D. M.; LEVGEN, K.; MAHMOOD, N.; SARAGIH, J.; SHILKROT, R. **Mastering OpenCV with Practical Computer Vision Projects**. [S.l.]: Packt Publishing, 2012.
- BEITZEL, S. M.; JENSEN, E. C.; GROSSMAN, D. A. Retrieving OCR text: A survey of current approaches. In: **Symposium on Document Image Understanding Technologies**. [S.l.: s.n.], 2003.
- BHATIA, N.; VANDANA. Survey of nearest neighbor techniques. **International Journal of Computer Science and Information Security**, v. 08, n. 02, p. 302–305, 2010.
- BOROVIKOV, E. A survey of modern optical character recognition techniques. **CoRR**, abs/1412.4183, 2014.
- BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library**. [S.l.]: O’Reilly Media, 2008.
- CHEN, X.; LI, S.; HU, J.; LIANG, Y. A survey on Otsu image segmentation methods. **Journal of Computational Information Systems**, v. 10, n. 10, p. 4287–4298, 2014.
- CHERIET, M.; KHARMA, N.; LIU, C.-L.; SUEN, C. **Character Recognition Systems: A Guide for Students and Practitioners**. [S.l.]: Wiley-Interscience, 2007.
- COVER, T.; HART, P. Nearest neighbor pattern classification. v. 13, p. 21– 27, 1967.
- DETRAN-SC. **Combinação de Letras e Placas**. 2012. Disponível em: <<http://www.detran.sc.gov.br/index.php/veiculos/combinacao-de-letras-e-placas>>. Acesso em: 04 de abril de 2014.
- DOUGHERTY, G. **Digital Image Processing for Medical Applications**. [S.l.]: Cambridge University Press, 2009.
- DU, S.; IBRAHIM, M.; SHEHATA, M.; BADAWY, W. Automatic license plate recognition (ALPR): A state-of-the-art review. **IEEE Transactions On Circuits and Systems for Video Technology**, v. 23, n. 02, p. 311–325, 2013.
- FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, v. 29, p. 1189–1232, 2000.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. [S.l.]: Prentice Hall, 2007.
- JIANG, L.; CAI, Z.; WANG, D.; JIANG, S. Survey of improving k-nearest-neighbor for classification. **Fuzzy Systems and Knowledge Discovery**, v. 1, p. 679–683, 2007.

- KAE, A.; HUANG, G.; DOERSCH, C.; LEARNED-MILLER, E. Improving state-of-the-art OCR through high-precision document-specific modeling. **IEEE**, p. 1935–1942, 2010.
- KHURSHID, K.; SIDDIQI, I.; FAURE, C.; VINCENT, N. Comparison of Niblack inspired binarization methods for ancient documents. **Document Recognition and Retrieval XVI**, v. 7247, 2009.
- KLUZNER, V.; TZADOK, A.; CHEVION, D.; WALACH, E. Hybrid approach to adaptive OCR for historical books. In: **ICDAR**. [S.l.]: IEEE, 2011. p. 900–904.
- LUVIZON, D.; NASSU, B.; MINETTO, R. Vehicle speed estimation by license plate detection and tracking. In: **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2014. p. 6563–6567.
- MINETTO, R.; THOME, N.; CORD, M.; LEITE, N. J.; STOLFI, J. Snoopertext: A text detection system for automatic indexing of urban scenes. **Computer Vision and Image Understanding**, v. 122, p. 92–104, 2014.
- MOESLUND, T. B. **Introduction to Video And Image Processing**. [S.l.]: Springer, 2012.
- NEUROPH. **Java Neural Network Framework**. 2014. Disponível em: <neuroph.sourceforge.net>. Acesso em: 19 de setembro de 2014.
- OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions On Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979.
- PRADHAN, A. Support Vector Machine - a survey. **International Journal of Emerging Technology and Advanced Engineering**, v. 2, n. 8, p. 82–85, 2012.
- QADRI, M. T.; ASIF, M. Automatic number plate recognition system for vehicle identification using optical character recognition. **International Conference on Education Technology and Computer**, p. 335–338, 2009.
- RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. [S.l.]: Elsevier, 2013.
- SAUVOLA, J.; PIEIKAINEN, M. Adaptive document image binarization. **Pattern Recognition**, v. 33, p. 2256–236, 2000.
- SEBE, N.; COHEN, I.; GARG, A.; HUANG, T. S. **Machine Learning in Computer Vision**. [S.l.]: Springer Netherlands, 2006.
- SHUTLER, J. **Statistical Moments - An Introduction**. 2015. Disponível em: <[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/SHUTLER3/node1.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/node1.html)>. Acesso em: 23 de março de 2015.
- SMITH, R. An overview of the Tesseract OCR engine. In: **Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR)**. [S.l.: s.n.], 2007. p. 629–633.
- TEAM, O. D. **Gradient Boosted Trees**. 2015. Disponível em: <[http://docs.opencv.org/modules/ml/doc/gradient\\_boosted\\_trees.html](http://docs.opencv.org/modules/ml/doc/gradient_boosted_trees.html)>. Acesso em: 23 de fevereiro de 2015.



WINDER, S. A. J.; BROWN, M. Learning local image descriptors. In: **2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA**. [S.l.: s.n.], 2007.

WOLF, C.; JOLION, J.-M.; CHASSAING, F. In: **International Conference on Pattern Recognition**. [S.l.: s.n.], 2002. p. 1037–1040.

ZHANG, H. The optimality of naïve Bayes. In: **Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference**. [S.l.]: AAAI Press, 2004.

## APÊNDICE A – TABELAS DE RESULTADOS

Este apêndice apresenta as tabelas de resultados para todos os métodos de segmentação e técnicas de aprendizado de máquina, assim como para o Tesseract. Para uma melhor visualização das tabelas, utilizou-se as abreviaturas: P(%) e R(%), para a precisão e a revocação, respectivamente. Ainda para melhorar a elegibilidade das tabelas, optou-se por omitir os casos em que a revocação e, conseqüentemente, a precisão, é 0,00%.

As Tabelas 12 a 16 apresentam os resultados para todas as técnicas de aprendizado de máquina e para os métodos de segmentação de Otsu, Niblack, Sauvola, Wolf e *SnooperText*, respectivamente.

A Tabela 17 refere-se aos três cenários avaliados na Seção 4.1, de forma que para o Cenário 1 avaliou-se somente a segunda coluna (Otsu Tesseract), no Cenário 2 foram consideradas da terceira até a sétima coluna e para o Cenário 3 toda a tabela foi analisada.

**Tabela 12:** Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Otsu.

Caractere	MLP: P(%) / R(%)	SVM: P(%) / R(%)	NBC: P(%) / R(%)	KNN: P(%) / R(%)	GBT: P(%) / R(%)
0		5,23 / 86,36		13,85 / 40,91	10,53 / 9,09
1				20,97 / 43,33	11,11 / 3,33
2				11,90 / 15,62	7,14 / 3,12
3				12,90 / 15,38	4,17 / 3,85
4				7,56 / 42,86	22,22 / 19,05
5					
6				14,29 / 4,00	16,67 / 8,00
7				20,00 / 5,56	26,67 / 22,22
8				11,11 / 5,88	
9					3,70 / 4,00
A		44,44 / 8,70		28,30 / 32,61	26,67 / 60,87
B					
C					
D			7,14 / 11,11		
E			6,38 / 27,27	20,00 / 9,09	25,00 / 18,18
F					
G					
H					
I					
J					16,67 / 25,00
K			1,33 / 100,00		
L		25,00 / 12,50	14,29 / 25,00		
M					
N					
O			2,94 / 33,33		
P					
Q			4,88 / 28,57		
R					
S					7,69 / 16,67
T					
U					
V					
W					
X					14,29 / 11,11
Y					
Z					

**Tabela 13:** Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Niblack.

Caractere	MLP: P(%) / R(%)	SVM: P(%) / R(%)	NBC: P(%) / R(%)	KNN: P(%) / R(%)	GBT: P(%) / R(%)
0		7,53 / 100,00		5,88 / 18,18	9,52 / 9,09
1				14,29 / 36,67	13,51 / 16,67
2	7,89 / 9,38			6,00 / 9,38	10,53 / 6,25
3	10,71 / 23,08			9,09 / 7,69	
4	2,13 / 4,76	28,57 / 9,52			
5	3,70 / 4,17			19,05 / 16,67	13,04 / 12,50
6	9,09 / 4,00			5,26 / 4,00	5,26 / 4,00
7					8,00 / 11,11
8	15,79 / 17,65			7,14 / 5,88	4,55 / 5,88
9	14,29 / 8,00			7,14 / 4,00	8,00 / 8,00
A		37,04 / 65,22		24,62 / 34,78	25,64 / 43,48
B					
C					
D	11,11 / 11,11		11,54 / 33,33		
E	7,14 / 9,09		11,43 / 36,36		
F					
G					
H					
I	9,09 / 20,00				
J			6,25 / 75,00		12,50 / 25,00
K			1,89 / 100,00		
L			5,00 / 12,50		
M					
N			6,00 / 60,00		
O					25,00 / 33,33
P					
Q			2,86 / 14,29		
R					
S					
T					
U					
V					
W	9,09 / 14,29				
X	14,29 / 11,11				
Y	10,00 / 50,00				
Z					

**Tabela 14:** Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Sauvola.

Caractere	MLP: P(%) / R(%)	SVM: P(%) / R(%)	NBC: P(%) / R(%)	KNN: P(%) / R(%)	GBT: P(%) / R(%)
0				8,70 / 27,27	11,11 / 9,09
1		8,33 / 10,00		18,18 / 40,00	14,29 / 13,33
2				13,04 / 18,75	12,50 / 6,25
3		6,47 / 34,62		17,39 / 15,38	7,14 / 3,85
4				13,16 / 23,81	15,00 / 14,29
5		18,18 / 16,67		11,54 / 12,50	5,56 / 4,17
6		4,88 / 8,00		7,14 / 8,00	7,50 / 12,00
7					11,76 / 11,11
8	5,11 / 70,59				25,00 / 5,88
9	4,69 / 12,00			6,67 / 4,00	10,00 / 12,00
A	13,04 / 6,52	25,64 / 21,74		26,76 / 41,30	25,97 / 43,48
B					
C					
D			12,50 / 22,22		
E			9,52 / 36,36	100,00 / 9,09	100,00 / 9,09
F					
G					
H					
I			8,00 / 40,00		
J			5,00 / 75,00		
K					
L			6,67 / 25,00		
M	25,00 / 20,00				
N			6,98 / 60,00		
O					
P					
Q			4,35 / 14,29		
R		2,00 / 16,67			
S					
T					
U					
V					
W					
X					
Y					
Z					

**Tabela 15:** Resultado de cada técnica de aprendizado de máquina para o método de segmentação de Wolf.

Caractere	MLP: P(%) / R(%)	SVM: P(%) / R(%)	NBC: P(%) / R(%)	KNN: P(%) / R(%)	GBT: P(%) / R(%)
0				10,53 / 27,27	4,17 / 4,55
1		6,90 / 6,67		18,18 / 26,67	12,90 / 13,33
2		5,56 / 3,12		11,36 / 15,62	5,26 / 3,12
3		7,55 / 15,38		12,50 / 15,38	23,08 / 11,54
4		100,00 / 4,76		16,00 / 19,05	
5	100,00 / 4,17			17,39 / 16,67	10,00 / 8,33
6		7,69 / 4,00			
7		7,78 / 72,22		6,67 / 5,56	4,55 / 5,56
8					
9				11,11 / 8,00	10,00 / 12,00
A	1,85 / 6,52	34,78 / 17,39	40,00 / 13,04	20,95 / 47,83	25,51 / 54,35
B	2,63 / 26,67				12,50 / 6,67
C					
D			15,38 / 22,22		
E			11,90 / 45,45		
F					
G					
H					
I			8,33 / 20,00		
J			4,76 / 75,00		
K					
L					
M	100,00 / 20,00				
N			1,79 / 20,00		25,00 / 20,00
O					
P			9,09 / 33,33		
Q			4,55 / 14,29		25,00 / 14,29
R					
S					
T		4,35 / 20,00			
U					8,33 / 33,33
V					16,67 / 16,67
W					
X					
Y					
Z					

**Tabela 16:** Resultado de cada técnica de aprendizado de máquina para o método de segmentação *SnooperText*.

Caractere	MLP: P(%) / R(%)	SVM: P(%) / R(%)	NBC: P(%) / R(%)	KNN: P(%) / R(%)	GBT: P(%) / R(%)
0					4,55 / 4,55
1		3,85 / 3,33			
2		4,55 / 6,25			
3					
4		7,14 / 4,76			
5		7,14 / 8,33			10,00 / 4,17
6		2,78 / 4,00		6,00 / 100,00	
7		6,67 / 5,56			4,17 / 5,56
8					8,33 / 11,76
9		11,76 / 16,00			11,11 / 12,00
A			10,53 / 4,35		16,31 / 50,00
B		2,50 / 6,67			
C		2,94 / 20,00			
D			2,78 / 11,11		
E					12,50 / 9,09
F					
G					
H			1,04 / 50,00		
I					
J					
K					
L			5,13 / 25,00		25,00 / 12,50
M					
N					
O			0,99 / 33,33		
P					
Q					
R					
S					
T					16,67 / 20,00
U					
V					
W					
X					
Y					
Z					

**Tabela 17:** Resultados do Tesseract para todos os métodos de segmentação.

Caractere	Otsu Tesseract: P(%) / R(%)	Otsu: P(%) / R(%)	Niblack: P(%) / R(%)	Sauvola: P(%) / R(%)	Wolf: P(%) / R(%)	<i>SnooperText</i> : P(%) / R(%)
0	50,00 / 4,55	66,67 / 9,09	50,00 / 9,09		75,00 / 13,64	
1						8,33 / 3,33
2	90,91 / 62,50	100,00 / 65,62	90,48 / 59,38	100,00 / 53,12	94,44 / 53,12	100,00 / 43,75
3	100,00 / 57,69	94,74 / 69,23	93,33 / 53,85	93,33 / 53,85	85,71 / 46,15	86,67 / 50,00
4	85,71 / 28,57	88,89 / 38,10	100,00 / 28,57	100,00 / 23,81	83,33 / 23,81	66,67 / 19,05
5	71,43 / 83,33	79,17 / 79,17	81,25 / 54,17	57,69 / 62,50	75,00 / 62,50	84,62 / 45,83
6	100,00 / 4,00	100,00 / 4,00		100,00 / 4,00		
7	100,00 / 66,67	100,00 / 72,22	70,00 / 38,89	77,78 / 38,89	70,00 / 38,89	88,89 / 44,44
8						100,00 / 5,88
9	100,00 / 16,00		50,00 / 4,00	100,00 / 12,00		
A	58,49 / 67,39	85,19 / 50,00	73,33 / 23,91	48,78 / 43,48	63,16 / 26,09	41,03 / 34,78
B	16,67 / 6,67	28,57 / 13,33	37,50 / 20,00	25,00 / 13,33	33,33 / 20,00	27,27 / 20,00
C	100,00 / 40,00	100,00 / 40,00				100,00 / 20,00
D	36,36 / 44,44	44,44 / 44,44	41,67 / 55,56	40,00 / 22,22	18,18 / 22,22	45,45 / 55,56
E	32,14 / 81,82	25,00 / 72,73	26,09 / 54,55	23,08 / 54,55	25,00 / 63,64	25,00 / 54,55
F		100,00 / 50,00	33,33 / 50,00	100,00 / 50,00	100,00 / 50,00	100,00 / 50,00
G			100,00 / 50,00	100,00 / 50,00	100,00 / 25,00	
H	20,00 / 50,00	12,50 / 50,00			12,50 / 50,00	
I	0,93 / 20,00	1,59 / 40,00	2,24 / 60,00	2,04 / 40,00	1,92 / 40,00	2,35 / 40,00
J	33,33 / 75,00	40,00 / 100,00	20,00 / 50,00	25,00 / 50,00	18,18 / 50,00	30,00 / 75,00
K	25,00 / 100,00	33,33 / 100,00	50,00 / 100,00	14,29 / 100,00	33,33 / 100,00	
L	63,64 / 87,50	75,00 / 75,00	50,00 / 50,00	50,00 / 50,00	44,44 / 50,00	50,00 / 75,00
M	20,00 / 20,00	20,00 / 20,00	14,29 / 20,00	12,50 / 20,00	20,00 / 20,00	
N	62,50 / 100,00	33,33 / 100,00	25,00 / 100,00	19,05 / 80,00	16,67 / 80,00	17,39 / 80,00
O						
P	100,00 / 66,67	100,00 / 66,67	100,00 / 66,67	100,00 / 66,67	100,00 / 66,67	100,00 / 33,33
Q		25,00 / 14,29		25,00 / 14,29		
R	100,00 / 16,67	100,00 / 16,67	100,00 / 100,00	100,00 / 50,00	83,33 / 83,33	75,00 / 100,00
S	40,00 / 66,67	80,00 / 66,67	37,50 / 50,00	37,50 / 50,00	33,33 / 50,00	33,33 / 33,33
T	100,00 / 60,00	100,00 / 40,00	75,00 / 60,00	100,00 / 60,00	100,00 / 60,00	100,00 / 20,00
U	40,00 / 66,67	16,67 / 33,33	66,67 / 66,67	50,00 / 66,67	20,00 / 33,33	16,67 / 33,33
V	75,00 / 100,00	62,50 / 83,33	60,00 / 50,00	75,00 / 50,00	66,67 / 66,67	83,33 / 83,33
W						
X	30,77 / 44,44	42,86 / 66,67	30,43 / 77,78	33,33 / 88,89	33,33 / 77,78	17,24 / 55,56
Y		50,00 / 50,00	100,00 / 50,00	100,00 / 50,00	100,00 / 50,00	
Z	14,29 / 100,00	33,33 / 100,00	33,33 / 100,00	20,00 / 100,00	50,00 / 100,00	