

# **Deteção e Reconhecimento de Placas Licenciadas**

Terceira entrega: Aplicação

## **Autores:**

201507733 - **Arnold Christian Veloso Lima**

201407353 - **Luana Guedes Barros Martins**

201514365 - **Rafael Alessandro Alves Tomé**

Data de entrega: 17 de maio de 2018

## **1. Introdução**

A proposta da construção de um Sistema de Reconhecimento Automático de Placas (ALPR) foi sugerida como um projeto final da disciplina Multimídia, na qual o projeto deve ser desenvolvido durante todo o semestre, e entregue no final do mesmo.

Os sistemas ALPR envolvem grande parte dos conceitos de Multimídia, entre eles, a utilização de filtros e morfologia matemática para a redução de ruídos, a utilização de algoritmos de detecção de borda durante o processo de localização da placa em uma determinada imagem e a extração de características de uma imagem e sua classificação.

O produto a ser entregue deve ser capaz de extrair a placa de uma imagem, segmentar e classificar cada caracter que compõe uma placa veicular, a fim de o resultado possa ser utilizado com alguma outra finalidade definida, como a aplicação de multa, por exemplo.

## **2. Métodos utilizados**

Na fase de pré-processamento da imagem, realizamos algumas operações com a finalidade de simplificar o problema e deixá-lo o mais livre o possível de ruídos que poderiam interferir na classificação posterior dos caracteres.

Para isso, é realizada uma redução no tamanho da imagem, porém mantendo a escala, conversão das cores RGB para os níveis de cinza, filtros para a redução de ruídos, como o filtro da mediana, por exemplo, e também operações morfológicas com o mesmo intuito. Através das operações de dilatação e erosão podemos obter as bordas da imagem com uma simples subtração (imagem dilatada – imagem erodida) que será de extrema importância na fase de extração dos caracteres. Antes de finalizarmos a fase de pré-processamento, realizamos algumas operações como detecção de linhas horizontais na imagem, para conseguirmos tirar partes indesejáveis, como, por exemplo, a borda da placa, que pode atrapalhar no resultado dos processos seguintes. Com a imagem contendo as bordas, podemos realizar outras operações, como fechar buracos, exclusão de regiões conexas que possuem uma pequena área que também são realizadas com a finalidade de eliminação de pequenos ruídos.

Durante a fase de extração de caracteres, utilizamos as bordas para detectar componentes conexas na imagem, que dada uma filtragem nos componentes da imagem, podemos obter somente as letras e número que desejamos, que são segmentados

separadamente e guardados para a fase de reconhecimento. Uma vez segmentado os caracteres, passamos então cada caractere para uma rede neural convolucional onde será retirado as características e então classificado com base nos resultados do treinamento da mesma.

### 3. Resultados de outros trabalhos

Em [3] trata da extração de placas em uma dada imagem. Os experimentos resultantes utilizando a biblioteca OpenCV (Biblioteca Open Source de Visão Computacional) em algumas bases de dados reais mostram que o método proposto se desempenha melhor do que outros algoritmos em termos de acurácia e tempo computacional. Tal comparação é ilustrada nos gráficos abaixo:

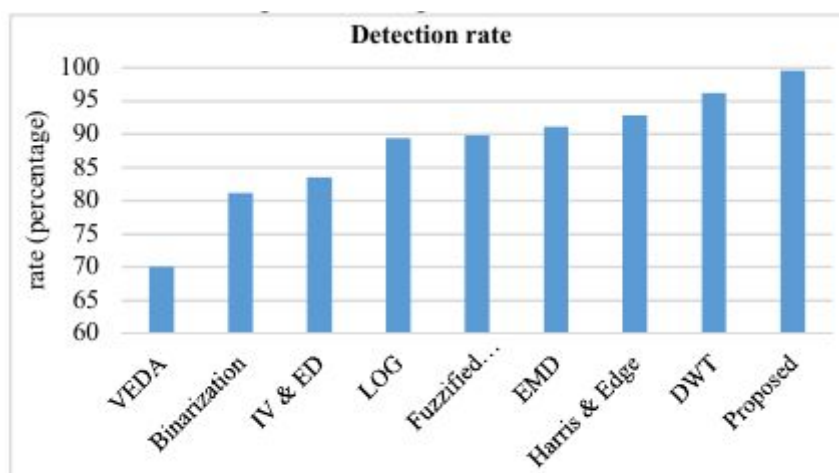


Fig. 7 A comparison between detection rates of proposed approach with some related works.

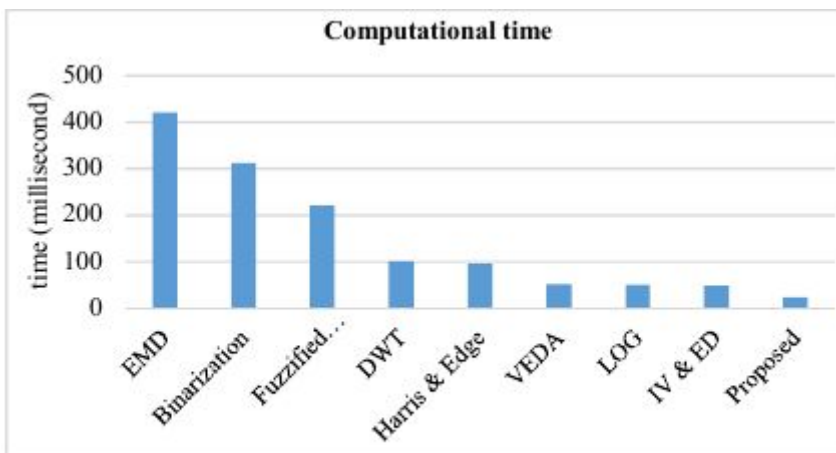


Fig. 8 A comparison of computational time of the proposed approach with some related works.

O método proposto pelo autor possui 3 passos: adaptive morphological closing, local adaptive thresholding e morphological opening. O dataset utilizado consiste em 2232 imagens obtidas de câmaras comerciais, com uma taxa de acerto maior do que 99%.

No artigo [2] em seus testes foram utilizados, para o treinamento, três vídeos com duração aproximada de 18, 10 e 3 minutos respectivamente e, para os testes, foram utilizados 2 vídeos de 25 e 20 minutos de duração. Os vídeos foram obtidos da portaria de

um condomínio residencial. Para o reconhecimento de caracteres, o autor propõe a utilização do classificador chamado Random Trees, baseado em árvores de decisão, onde podemos ver sua acurácia na tabela abaixo:

<b>Total de letras</b>	63	100%
Letras corretamente reconhecidas	59	93,7%
Letras não reconhecidas	4	6,3%
<b>Total de dígitos</b>	84	100%
Números corretamente reconhecidos	73	86,9%
Números não reconhecidos	11	13,1%

Em [1] o método trata-se de um Averaged Multiclass Perceptron aplicado em 11 datasets diferentes para o reconhecimento dos caracteres, onde os resultados podem ser vistos na tabela abaixo:

Dataset	Acc. Números	Acc. Letras	Total
<b>(Wang et al., 2010)</b>	-	-	<b>98%</b>
Pixel	99,1%	96,0%	98,1%
Pixel + EFG	99,1%	96,0%	98,1%
Pixel_Proj	99,1%	96,1%	98,1%
Pixel_Proj + EFG	99,1%	96,1%	98,1%
Pixel_Proj_Bloc	99,0%	96,0%	98,0%
Pixel_Proj_Bloc + EFG	99,1%	96,1%	98,1%
Proj_Bloc	99,1%	96,5%	98,2%
Proj_Bloc + EFG	99,2%	96,6%	98,3%
<b>(Comelli et al., 1995)</b>	-	-	<b>98,6%</b>
Proj_Bloc + EFG (Itália)	99,2%	97,4%	98,6%

Na análise de erros do reconhecimento de números existe uma taxa de variação de acerto entre 98.3% e 100%, sendo os dígitos considerados como mais problemáticos, os números 5 e 6, sendo estes confundidos entre si. Já na parte de reconhecimento de caracteres vemos uma variação entre 82.4% e 100%, onde temos os caracteres D, Q e O não apresentando diferença visual entre eles significativa, e o caractere Y apresentando uma grande deformação, ambos causados pela baixa resolução da imagem.

#### 4. Resultados da aplicação proposta

A fase de pré-processamento é de suma importância para a fase de reconhecimento, utilizando os passos descritos na seção 2, segue alguns resultados obtidos:



Através desse resultado é possível extrair os caracteres, utilizando métodos de busca contornos na imagem, a serem enviados para a próxima fase, que é a de reconhecimento.

Exemplos de caracteres extraídos:

259H

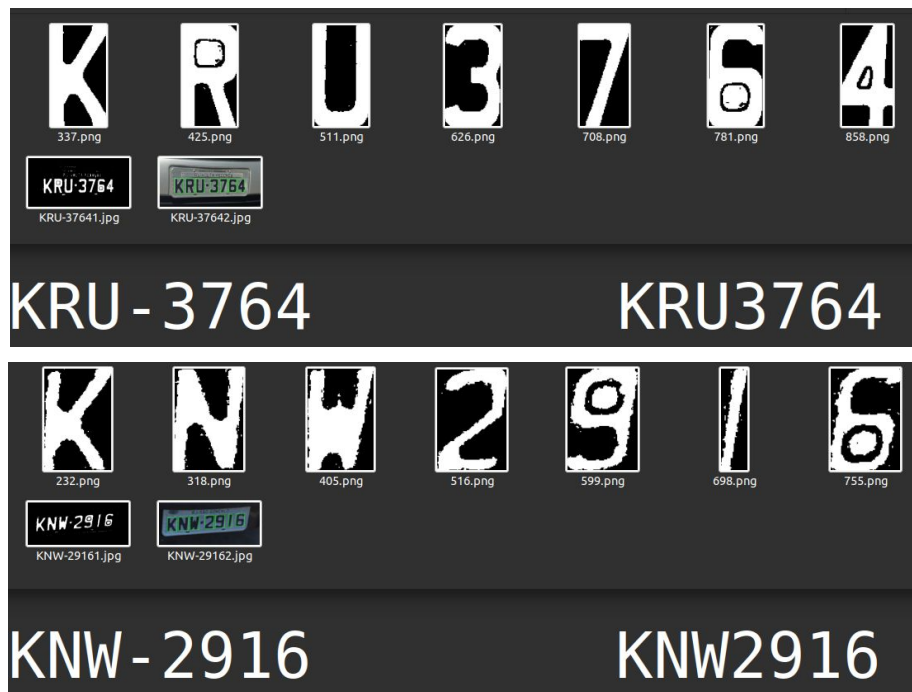
Na fase de reconhecimento, alguns problemas encontrados vem do “0”/“O” e “1”/“l” possuírem a mesma fonte. Para tratar esse problema, foram usados exemplos de apenas um dos casos para os treinamento (para evitar confusão da rede e a ausência de algum resultado), e então feito uma análise de contexto ao final. Outros exemplos de caracteres semelhantes, cujo problema foi resolvido com análise de contexto foi o caso de: “2”/“Z”, “8”/“B”, “5”/“S” e em alguns casos o número “0” veio a ser caracterizado como “D”.

Alguns erros que não foi possível ser resolvido é a confusão entre o número “8” e o número “4”.



No geral, obtivemos uma acurácia média de 84.17%, com a acurácia mais alta de 92.92%. Nos testes realizados, em 96% dos casos de segmentação de caracteres bem sucedida, a rede conseguiu reconhecer com sucesso a placa dada. Segue alguns exemplos:





## 5. Referências

- [1] Alvarenga, E. P. [2014]. “Identificação de Caracteres para Reconhecimento Automático de Placas Veiculares”, Pontifícia Universidade Católica do Rio, Departamento de Informática, Dissertação de Mestrado.
- [2] Trentini, V. B.; Godoy, L. A. T.; Marana, A. N. “Reconhecimento Automático de Placas de Veículos”. Anais do WVC 2010 - VI Workshop de Visão Computacional. Presidente Prudente, 2010, pp. 267-272
- [3] Fomani, B. A.; Shahbahrami, A..“License Plate Detection using Adaptive Morphological Closing and Local Adaptive Thresholding”, IPRIA, 2017.
- [4] Haider, S. A.; Khurshid, K.. “An Implementable System for Detection and Recognition of License Plates in Pakistan”, ICIEECT, 2017.
- [5] Li, H.; Shen, C.. “Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs”, 2016.
- [6] Kakani, B.; Gandhi, D.; Jani, S.. “Improved OCR based Automatic Vehicle Number Plate Recognition using Features Trained Neural Network”, ICCCNT, 2017.