

RECONHECIMENTO AUTOMÁTICO DE PLACAS DE VEÍCULOS UTILIZANDO PROCESSAMENTO DIGITAL

Marcelo A. da Silva^{1*}, Robinson L. de Souza Alves², Ricardo Alessandro de M. Valentim³ e Ivanilson F. Vieira Júnior⁴

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Avenida Senador Salgado Filho, 1559, Tirol. Natal - RN - CEP 59015-000
e-mail: marcelothis@gmail.com

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Universidade Federal do Rio Grande do Norte
Avenida Senador Salgado Filho, 1559, Tirol. Natal - RN - CEP 59015-000
e-mail: {robinson.alves@ig.com.br, ricardo.lahb@gmail.com, ivanilson.junior@ifrn.edu.br}

Palavras-chave: Processamento de imagens, reconhecimento ótico de caracteres (OCR), reconhecimento automático de placa de veículo.

Resumo. *Algumas décadas atrás, o processamento de imagem era feito de forma analógica, através de dispositivos óticos. Devido ao grande aumento no poder computacional, tais técnicas foram substituídas por algoritmos de processamento digital de sinais. Neste trabalho é apresentado o desenvolvimento de um software projetado para o reconhecimento automático de placas de veículos. Os Algoritmos Shape Detection e Canny Edge foram utilizados para extrair características das placas dos veículos. Com o objetivo de diminuir o esforço computacional, foi incorporado no sistema proposto um pré-processamento da imagem para ajustá-la ao processamento ótico. A base de dados utilizada para validar e analisar a taxa de acerto do sistema foi produzido no Instituto Federal do Rio Grande do Norte, de onde foram obtidas as imagens. O desempenho do software proposto obteve 92% de acerto e baixo custo computacional, tendo como base 180 imagens, dentre elas, algumas imagens estão distorcidas ou com muita luminosidade ofuscando os caracteres a serem reconhecidos.*

1. INTRODUÇÃO

Hoje em dia, é comum a utilização de computadores como auxiliar do homem na execução de tarefas que antes eram executadas manualmente. Ultimamente, observamos um crescimento no uso de computadores em tarefas que antes pareciam impossíveis, tais como o reconhecimento de caracteres em imagens e cores. Com o avanço no processamento digital de imagens, o mesmo passou a ser aplicado em diversas áreas, como na medicina, em ressonância magnética; na indústria, para reconhecimento de placas de circuitos com peças faltantes; e na segurança pública, para reconhecimento de placas de veículos.

Existe um número cada vez maior de veículos trafegando em vias públicas de cidades em todo o mundo. O aumento nesta quantidade significa uma maior dificuldade no controle do trânsito em grandes cidades, com isso, surge à necessidade de um sistema capaz de reconhecer placas

de veículos automotores é cada vez maior.

O sistema proposto, neste artigo, foi comparado com outros dois sistemas. Primeiro, foi realizada a comparação com o sistema license plate recognition[1], e, por último, foi comparado com o sistema OpenALPR[2]. Observamos que o sistema proposto obteve maior taxa de acerto comparado com os dois sistemas citados acima. O sistema license plate recognition[1] foi desenvolvido com a biblioteca Emgu CV[3] de código aberto. Já o sistema OpenALPR[2] também utiliza uma biblioteca de código aberto chamada de Open CV[4] e utiliza a linguagem C++.

No desenvolvimento do sistema proposto, neste artigo, foram utilizadas várias técnicas de processamento digital. Essas técnicas vão desde a localização da placa até o reconhecimento óptico de caracteres (OCR)[5]. Uma aplicação interessante para o trabalho proposto seria um dispositivo em que o reconhecimento automático de placas seria utilizado no controle eletrônico de estacionamentos, onde poderia ser emitido um recibo automaticamente. Podendo também ser utilizado no monitoramento em vias públicas, capturando a velocidade e a placa do veículo, para um melhor controle do trânsito. Tudo isso é possível sem muito custo computacional, tornando viável a utilização de reconhecimento por imagens aplicado a problemas do cotidiano, como o reconhecimento de placas de veículos.

O restante deste trabalho está organizado como segue: a seção 2 aborda a fundamentação teórica, os algoritmos utilizados para o reconhecimento automático das placas. A seção 3 descreve como foi o desenvolvimento do sistema proposto, como os algoritmos de processamento foram utilizados. A seção 4 apresenta os resultados obtidos e comparação com outros dois sistemas. A seção 5 apresenta as conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento deste trabalho, foram utilizadas algumas técnicas para o processamento das imagens. Diversos métodos de computação visual foram empregados, combinando aspectos de computação gráfica e processamento de imagem. A seguir, apresentaremos as técnicas que foram utilizadas no desenvolvimento do mesmo.

2.1. Algoritmo Shape Detection

Este algoritmo é utilizado para detecção de bordas em imagens. No desenvolvimento deste sistema, utilizou-se a linguagem VB.Net, que utiliza a biblioteca Emgu CV para realização de processamento em imagens.

O Shape Detection[6] é um conjunto de métodos matemáticos que visam à identificação de pontos em uma imagem digital. A detecção de bordas é uma ferramenta fundamental no processamento de imagem, visão de máquina e visão computacional, particularmente nas áreas de detecção de recurso e de extração de características.

O processo de detecção de bordas de uma imagem pode conduzir a um conjunto de curvas conectado que indicam os limites da placa do veículo. Assim, o algoritmo para localizar a placa em uma imagem, pode reduzir significativamente a quantidade de dados a serem processados e, conseqüentemente, pode filtrar informações que podem ser menos relevantes, enquanto preserva as propriedades estruturais mais importantes de uma imagem. Após a etapa de detecção de borda for bem sucedida, a tarefa subsequente de interpretar o conteúdo de informação na imagem original pode ser bastante simplificado. No entanto, nem sempre é

possível obter isso, devido a alguns fatores externos, como luminosidade e o desgaste das placas nos veículos.

2.2. Algoritmo Canny Edge

O algoritmo Canny Edge[7][9] tem sido amplamente aplicado em vários sistemas de visão computacional, o que é uma técnica importante para extrair informação útil estrutural, a partir de diferentes objetos de visão e reduzir drasticamente a quantidade de dados a serem processados. Canny descobriu que os requisitos para a aplicação de detecção de bordas em diversos sistemas de visão são relativamente o mesmo. Assim, um desenvolvimento de uma solução de detecção de borda para resolver estes requisitos pode ser implementado numa vasta gama de situações.

Este pré-processamento é utilizado para filtrar todos os ruídos da imagem. Um exemplo de um núcleo gaussiano de tamanho cinco que pode ser utilizado é mostrado na equação 1.

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (1)$$

Este algoritmo é utilizado na terceira etapa do sistema, logo após a placa ser encontrada. Na imagem, ela é recortada e aplica-se o pré-processamento Canny Edge[7][9]. Esse filtro foi escolhido pela eficiência para destacar as bordas dos caracteres da placa. Abaixo segue a figura 1 com exemplo do algoritmo Canny Edge[7][9].



Figura 1. Figura com o filtro Canny Edge.

2.3. Histograma

O histograma de uma imagem digital é uma distribuição dos seus níveis de intensidade discretas no intervalo $[0, L-1]$. A distribuição é uma função h discreta de associação para cada nível de intensidade: r_k . O número de pixels com esta intensidade: n_k [8].

2.4. Equalização de um histograma

Equalização de histograma é um método para processar imagens, a fim de ajustar o contraste de uma imagem pela modificação da distribuição de intensidade do histograma. O objetivo desta técnica é o de dar uma tendência linear para a função de probabilidade cumulativa associada à imagem. Abaixo segue um exemplo de histograma.

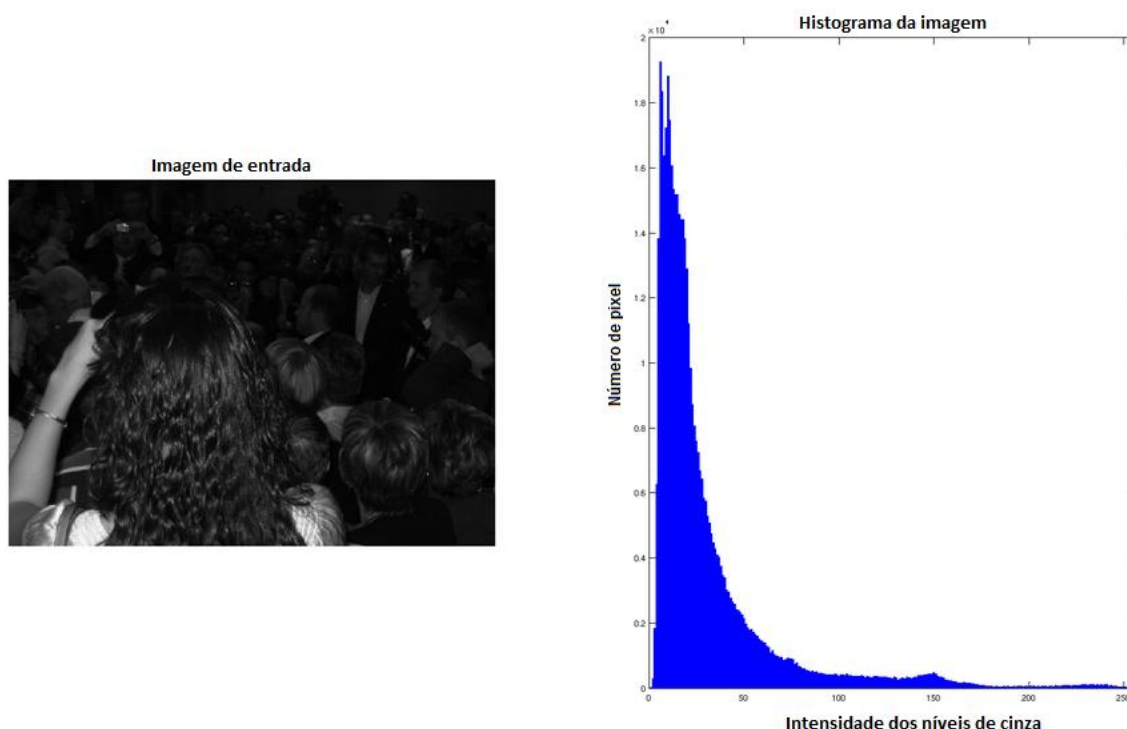


Figura 2. Figura com o exemplo de histograma.

O processamento de equalização de histograma baseia-se na utilização da função de distribuição cumulativa (FDC). A função de distribuição cumulativa (FDC) é uma soma cumulativa de todas as probabilidades que se encontram no seu domínio, e, é definida pela equação 2.

$$f_{dc}(x) = \sum_{k=-\infty}^x P(k) \quad (2)$$

Onde X é a variação dos níveis de intensidade, FDC(x) é a função de equalização de histograma e P(k) é o histograma da imagem de entrada.

2.5. Reconhecimento ótico de caracteres

O reconhecimento ótico de caracteres (OCR)[5] é um processo pelo qual os caracteres de texto de uma imagem podem ser introduzidos a um computador. O computador utiliza um programa com uma função específica para converter os caracteres da imagem em texto. Este mecanismo é utilizado para reconhecer arquivo de imagem ou mapa de bits, sejam eles escaneados, escritos à mão, datilografados ou impressos. Dessa forma, através do OCR[5], é possível obter um arquivo de texto editável por um computador.

No desenvolvimento do sistema proposto neste artigo foi utilizado o Microsoft Office

Document Imaging (MODI)[5], é uma aplicação que suporta edição de documentos digitalizados com o aplicativo Microsoft Office Document Xining. Foi introduzido pela primeira vez no Microsoft Office XP e está incluído no Office 2003 e Office 2007. Embora ele não esteja disponível no Office 2010, é possível instalá-lo a partir de uma versão anterior do Microsoft Office e usá-lo com o Office 2010. O MODI permite aos usuários digitalizar documentos (via MODS), executar o reconhecimento ótico de caracteres (OCR)[5].

3. DESENVOLVIMENTO

No desenvolvimento deste trabalho, utilizou-se uma plataforma chamada Emgu CV[3], que utiliza a biblioteca Open CV[4] para o processamento de imagem digital. Para o reconhecimento automático foram utilizadas várias técnicas. Essas técnicas vão desde a localização da placa na imagem até a extração dos caracteres. Para extrair os caracteres da placa, é preciso passar por quatro etapas. A primeira etapa é na localização da placa, onde o algoritmo Shape Detection[6] procura as possíveis placas dentro da imagem, após a localização da placa a mesma é recortada. Na segunda etapa, é aplicada a técnica histograma para pegar a intensidade da imagem. Na terceira etapa, é utilizado o pré-processamento Canny Edge[7][9] para destacar os caracteres da placa. E a última etapa é aplicada o reconhecimento automático de caracteres OCR.

3.1. Base de dados

Inicialmente, utilizou-se uma base de dados só com a placa dos veículos. Esta base de dados continha 40 imagens, de onde foram realizados os testes iniciais. Porém, após os testes iniciais, sentiu-se a necessidade de obter uma base de dados maior para o desenvolvimento deste software. Neste contexto, foi criada uma base de dados, exclusivamente para a realização deste trabalho. Esta nova base de dados foi criada em frente a uma cancela no Instituto Federal do Rio Grande do Norte (IFRN), onde as imagens foram obtidas com o intuito de desenvolver este software para reconhecimento de placas automático. Estas imagens são do veículo inteiro.

A primeira etapa consiste na localização da placa. Para isso, é utilizado o algoritmo Shape Detection[6], que procura as possíveis placas dentro da imagem. Após isso, é aplicado o reconhecimento ótico de caracteres (OCR) para verificar se existem caracteres dentro daquele retângulo. Após encontrar o retângulo com os caracteres, ele é recortado. Na figura 3, é mostrado um exemplo do algoritmo Shape Detection[6].



Figura 3. Localizando e recortando a placa no veículo utilizando o algoritmo Shape Detection.

A segunda etapa é a mais simples, na qual o histograma pega a intensidade da imagem, para depois aplicar o pré-processamento Canny Edge[7][9].

A terceira etapa corresponde ao pré-processamento da placa já recortada. Este pré-processamento é para destacar os contornos dos caracteres da placa recortada anteriormente. Inicialmente, o sistema apresentou certa lentidão, principalmente se a imagem da placa fosse clara, pois o tempo do pré-processamento tornaria alto. Até que os caracteres da placa ficassem totalmente legíveis para o reconhecimento automático de caracteres, o pré-processamento demorava cerca de três minutos, para cada imagem, então foi utilizada a técnica histograma, no passo anterior, para solucionar este problema. A figura 4 e figura 5 mostram um exemplo do pré-processamento.



Figura 4. Pré-processamento Canny Edge, antes do reconhecimento dos caracteres.



Figura 5. Pré-processamento Canny Edge com placa escura, antes do reconhecimento dos caracteres.

Acima, são mostradas a figura 4 e a figura 5, onde é mostrado o pré-processamento Canny Edge[7][9]. Na figura 4, é mostrado o pré-processamento com uma placa mais clara, onde o pré-processamento Canny Edge[7][9] demora mais tempo para destacar os caracteres da placa. Por isso, foi aplicada a técnica histograma, na qual, aplica-se o valor ideal para cada imagem. Já na figura 5, é aplicado o pré-processamento com a imagem da placa mais escura. Neste, o tempo de processamento é mais rápido, pois o pré-processamento não precisa se estender muito para destacar os caracteres da placa.

Na última etapa do reconhecimento da placa, após realizar o processamento da imagem, é aplicado o reconhecimento ótico de caracteres para extrair os caracteres da placa já processada. A figura 6 mostra um exemplo da última etapa, o reconhecimento dos caracteres utilizando OCR.

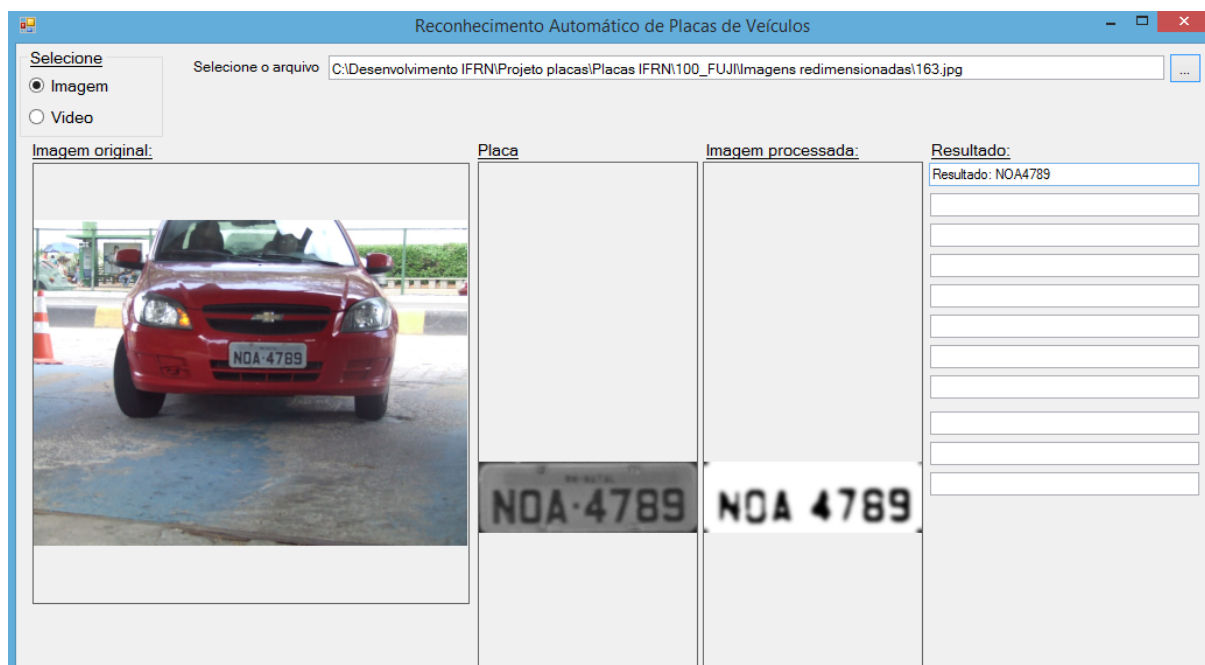


Figura 6. Figura com resultado, reconhecimento automático de caracteres (OCR).

Acima é mostrada a figura 6, onde é apresentado o processo final do sistema. Primeiro, aplica-se o algoritmo Shape Detecion[6] para encontrar a placa no veículo e recortá-la. Em seguida, aplica-se a técnica histograma para aplicar o pré-processamento Canny Edge[7][9] para destacar os contornos dos caracteres e, para finalizar, aplica-se o mecanismo de reconhecimento de caracteres (OCR) para extrair os caracteres da placa.

3.2. Arquitetura do sistema proposto

Nesta seção é apresentada a arquitetura proposto neste trabalho, onde explica de forma detalhada a arquitetura utilizada para o desenvolvimento do sistema proposto. Além disso, é apresentado o processo de localização da placa até o último passo que é a extração dos caracteres da placa.

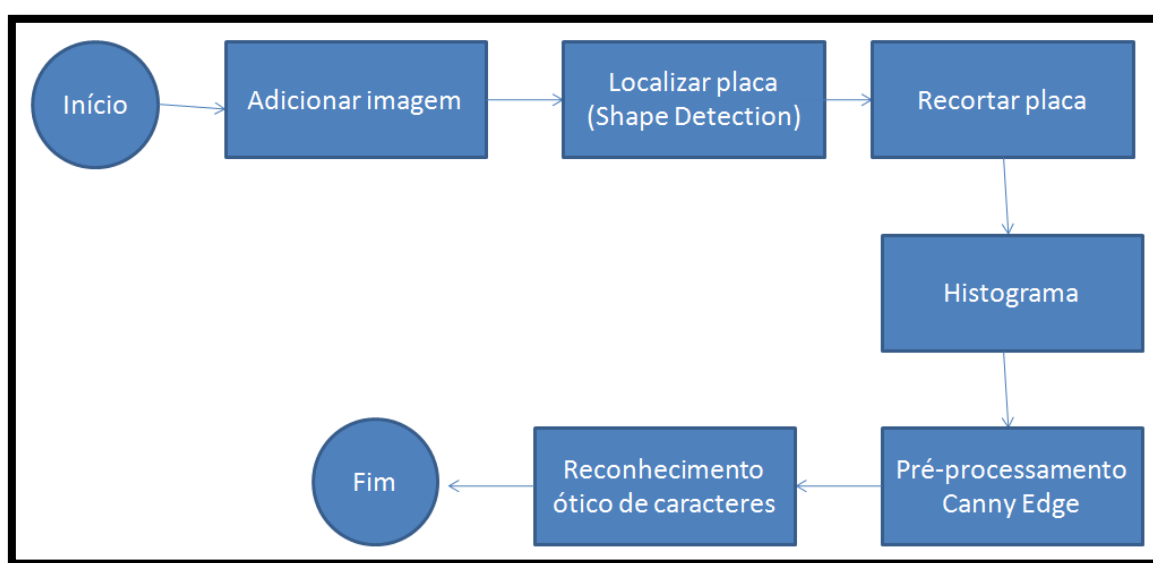


Figura 7. Figura com fluxograma, mostrando o processo completo para o reconhecimento dos caracteres.

Acima, é mostrada a figura 7 com um fluxograma, apresentando o processo completo para o reconhecimento automático dos caracteres da placa. Onde adicionamos a imagem através de uma câmera, usamos o algoritmo Shape Detection[6] para localizar a placa, após a localização da placa a mesma foi recortada, e aplicado a técnica histograma para pegar a intensidade da imagem, com isso foi aplicado o pré-processamento Canny Edge[7][9], com a placa já processada aplica-se o reconhecimento ótico de caracteres (OCR).

4. RESULTADOS

Inicialmente, os testes foram realizados com uma base de dados com 40 imagens. Essas imagens são somente da placa do veículo. Este primeiro teste obteve 100% de acerto, porém o sistema apresentou certa lentidão.

Para melhorar o desempenho do sistema proposto, utilizou-se uma técnica chamada de histograma, para obter a intensidade da imagem e, assim, aplicar o processamento ideal para

cada imagem, ganhando desempenho. Além disso, após os testes iniciais com as 40 imagens da placa, sentimos necessidade de uma base de dados maior para a evolução do sistema, então foi construída uma base de dados com 180 imagens exclusivamente para o desenvolvimento deste trabalho. Abaixo, segue uma figura (Figura 8) com um gráfico mostrando o primeiro teste realizado. O primeiro teste foi realizado com a técnica histograma e sem a técnica histograma. O teste realizado sem a técnica histograma obteve, aproximadamente, 17 minutos de processamento para reconhecer todas as placas. Já o teste com a técnica histograma, obteve aproximadamente 6 minutos para reconhecer todas as placas.

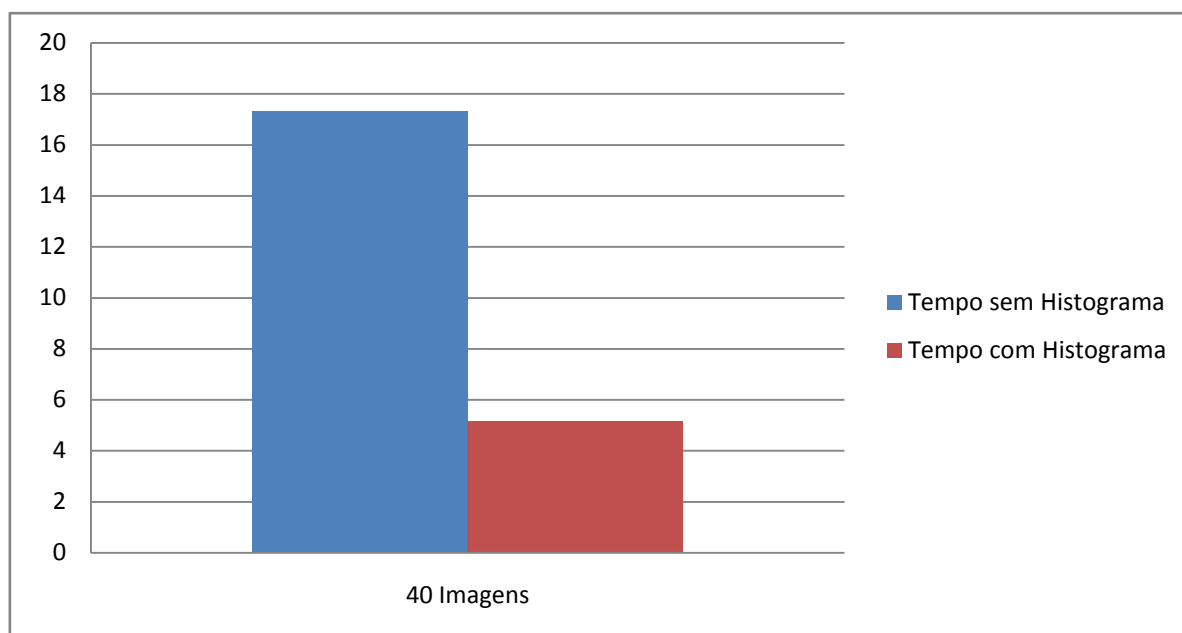


Figura 8. Figura com o tempo do reconhecimento dos caracteres com e sem a técnica histograma com as 40 imagens.

Após os testes iniciais, foi realizado um novo teste com a nova base de dados, cujo resultado obteve 92% de acerto. Estes testes também foram realizados com e sem a técnica histograma. Inicialmente foi realizado o teste sem o histograma, e obteve-se o pior tempo, aproximadamente 20 minutos e 6 segundos para reconhecer todos os caracteres. Com a técnica histograma, o desempenho do sistema melhorou consideravelmente, levando, aproximadamente, 7 minutos e 4 segundos para reconhecer todos os caracteres. Abaixo, segue a figura 9, mostrando o segundo teste com as 180 imagens.

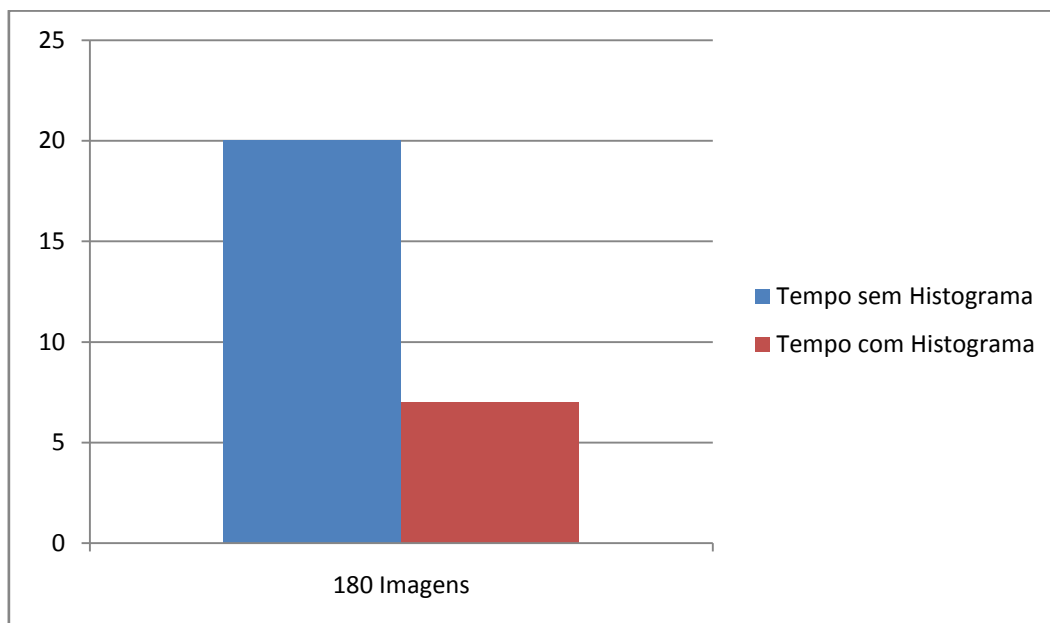


Figura 9. Figura com o tempo do reconhecimento dos caracteres com e sem a técnica histograma, com as 180 imagens.

As Tabelas 1 e 2 apresentam os testes realizados com as duas bases de dados utilizadas para o desenvolvimento deste trabalho. Como foi dito acima, inicialmente os testes foram realizados com uma base de dados com 40 imagens da placa dos veículos, e logo após, foram realizados testes com a segunda base de dados, com as 180 imagens que compreendem o veículo inteiro.

Primeira base de dados (40 imagens)					
Sem histograma			Com histograma		
Tamanho imagens	Tempo	Acerto (%)	Tamanho imagens	Tempo	Acerto (%)
160px X 60px	16 minutos	100%	160px X 60px	5 minutos e 6 segundos	100%

Tabela 1. Tabela com resultados com as 40 imagens 160px X 60px, com e sem a técnica histograma.

Segunda base de dados (180 imagens)					
Sem histograma			Com histograma		
Tamanho imagens	Tempo	Acerto (%)	Tamanho imagens	Tempo	Acerto (%)
900pxX 900px	16 Min. E 20 seg.	100%	900px X 900px	6Min. E6 seg.	92%

Tabela 2. Tabela com resultados com as 180 imagens de 900px X 900px, com e sem a técnica histograma.

Por último, foi realizado o teste comparando o sistema proposto neste artigo com outros

dois sistemas. A primeira comparação foi realizado com o sistema license plate recognition[1] e com o sistema OpenALPR[2]. A Tabela 3 apresenta os resultados comparativos, onde se observa que o trabalho proposto neste artigo obteve maior taxa de acerto em relação aos dois sistemas. A comparação foi realizada com a base de imagens de placas do IFRN utilizada para o desenvolvimento deste sistema.

Sistema de reconhecimento de placas	Acerto (%)	Erro (%)
Sistema proposto neste artigo	165 (92%)	15 (8%)
License Plate Recognition[1]	45 (25%)	135 (75%)
OpenALPR[2]	46 (26%)	134 (74%)

Tabela 3. Tabela comparando os resultados com os sistemas: License Plate Recognition, OpenALPR e Sistema proposto neste artigo.

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou um sistema de reconhecimento automático de placas de automóveis com técnicas de processamento digital. O sistema utilizou-se várias técnicas para extrair os caracteres da placa, desde a localização da placa na imagem até o reconhecimento ótico de caracteres. O software proposto tem sido eficiente no reconhecimento de imagens em bom estado.

Os erros apresentados resultam da influência direta de fatores externos ao sistema, como a qualidade da imagem adquirida, o posicionamento da placa na imagem, o estado de conservação da placa e/ou a luminosidade.

O sistema pode ser implementado para monitorar a entrada e a saída de automóveis em lugares particulares onde exista grande circulação de veículos, como estacionamentos de lojas, empresas, condomínios e outros estabelecimentos que queiram implementar, de alguma forma, o monitoramento de veículos. Este monitoramento pode proporcionar um maior controle, melhorando a agilidade e aumentando a segurança. O sistema também pode ser implementado em vias públicas, pegando a placa dos veículos que circulam nas grandes avenidas, permitindo um maior controle.

Com trabalhos futuros, poderiam ser aprimorados os algoritmos e técnicas utilizadas para o reconhecimento automático da placa, inserir novos filtros de processamento para o reconhecimento da placa no veículo e alterar o método de reconhecimento ótico de caracteres, pois a taxa de acerto pode melhorar, assim como a complexidade do algoritmo.

REFERÊNCIAS

- [1] License Plate Recognition in CSharp (2015). License Plate Recognition [Online]. Disponível: http://www.emgu.com/wiki/index.php/License_Plate_Recognition_in_CSharp
- [2] OpenALPR License Plate Recognition (2015).OpenALPR[Online]. Disponível: <http://www.openalpr.com/>
- [3] Emgu CV: Open CV in .Net (2014).Emgu CV [Online]. Disponível: http://www.emgu.com/wiki/index.php/Main_Page
- [4] Open CV (2014).Open CV [Online]. Disponível: <http://opencv.org/>
- [5] OCR Method (2014). OCR [Online]. Disponível: <https://msdn.microsoft.com/en->

- us/library/office/aa202819(v=office.11).aspx
- [6] Shape (Triangle, Rectangle, Circle, Line) Detection in CSharp (2014). Shape Detection [Online]. Disponível: [http://www.emgu.com/wiki/index.php/Shape_\(Triangle,_Rectangle,_Circle,_Line\)_Detection_in_CSharp](http://www.emgu.com/wiki/index.php/Shape_(Triangle,_Rectangle,_Circle,_Line)_Detection_in_CSharp)
 - [7] Feature Detectors - Canny Edge Detector (2014).Canny edge detector [Online]. Disponível: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>
 - [8] University of Utah: CS6640 Image Processing Report (2014). Histograma [Online]. Disponível: http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html
 - [9] C. Patel, D Shah, A. Patel, “Automatic Number Plate Recognition System (ANPR): A Survey”, *International Journal of Computer Applications*, Vol. 69– No.9, May 2013.