

**FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FELIPE GADELHA RUOSO

**USO DE TÉCNICAS PARA RECONHECIMENTO DE CARACTERES
EM PLACAS AUTOMOTIVAS VOLTADO AO CONTROLE DE
ACESSO EM LOCAIS PRIVADOS**

**MARÍLIA
2014**

FELIPE GADELHA RUOSO

**USO DE TÉCNICAS PARA RECONHECIMENTO DE CARACTERES
EM PLACAS AUTOMOTIVAS VOLTADO AO CONTROLE DE
ACESSO EM LOCAIS PRIVADOS**

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador
Prof. Msc. Maurício Duarte

**MARÍLIA
2014**



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Felipe Gadelha Ruoso

Reconhecimento de caracteres em placas automotivas para controle de acesso em locais privados

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação:

Nota: 8,5 (oito e meia)

Orientador: Mauricio Duarte

1º. Examinador: Renata Aparecida de Carvalho Paschoal

2º. Examinador: Leonardo Castro Botega







Marília, 05 de dezembro de 2014.

RUOSO, Felipe Gadelha

**Uso de técnicas para reconhecimento de caracteres em placas
automotivas voltado ao controle de acesso em locais privados** / Felipe
Gadelha Ruoso; orientador: Prof. Msc. Maurício Duarte. Marília, SP: [s.n.],
2014.

44 folhas

Monografia (Bacharelado em Ciência da Computação): Centro
Universitário Eurípides de Marília.

Dedico esse trabalho a todas aquelas pessoas que se mantiveram próximas e auxiliaram de alguma maneira para a construção do projeto, em especial, minha família.

AGRADECIMENTOS

Agradeço, primeiramente a Deus, por me conceder a capacidade de aprendizagem e forças para alcançar objetivos.

A minha família por sempre me apoiar durante esses quatro anos, especialmente minha avó Sueli Italia Pavelski Ruoso, meus pais, Albino Paulo Ruoso Junior e Angela Aparecida Gadelha Ruoso e minha irmã Lívia Gadelha Ruoso.

A todos meus amigos, que se mantiveram ao meu lado em todos os momentos, sendo esses desde antes do ingresso à graduação aos novos concretizados ao longo desta caminhada, onde destacam-se, Eduardo Arakaki, João Paulo dos Santos Moriyo e Vitor de Souza Rodrigues Luis.

Aos professores da universidade pelo compartilhamento do conhecimento, sendo fundamental para o crescimento acadêmico e pessoal.

Ao excelente Professor Msc. Maurício Duarte, meu orientador, pelo apoio, confiança e determinação, levando à concretização do projeto.

Enfim, todos aqueles que acreditaram no esforço desse trabalho.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”
Albert Einstein.

RESUMO

Reconhecimento de caracteres é uma área de grande interesse no cenário científico e empresarial, de modo a automatizar tarefas. Um sistema de reconhecimento de caracteres possibilita a verificação e/ou classificação de padrões, assim, aplicado em imagens, a abordagem de Reconhecimento Óptico de Caracteres (OCR) é uma maneira escolhida devido ao seu sucesso na atuação de resoluções de padrões, levando em consideração que o cenário aplicado possui poucas variações a serem identificadas. No desenvolvimento deste projeto foram analisadas ferramentas, bibliotecas e linguagens de programação, tornando possível a implementação um reconhecedor de caracteres, utilizando técnicas de Processamento de Imagens, de modo a facilitar o controle de acesso de um determinado automóvel em um local. Esse tipo de abordagem implica em uma menor interação do usuário com um controlador de acesso, proporcionando ao usuário maior conforto, minimizando tempo de espera.

Palavras-Chave: Reconhecimento Óptico de Caracteres; Processamento de Imagens.

ABSTRACT

Character recognition is an area of great interest in the scientific and business environment in order to automate tasks. A character recognition system allows the verification and / or classification of patterns; Thus, applied to images, the approach of Optical Character Recognition (OCR) is a chosen way due to its success in solving patterns, considering that the applied scenario has few variations to be identified. In the development of project, tools, programming languages and libraries were analyzed, making it possible to implement a characters recognizer , using techniques of Image Processing, to facilitate the access control of a vehicle in a particular place. This approach implies in a smaller user interaction with an access controller, giving the user greater comfort, minimizing the wait time.

Keywords: Optical Character Recognition; Image Processing.

LISTA DE ILUSTRAÇÕES

Figura 1. Imagem para demonstração de processamento de imagens.	19
Figura 2. Imagem gerada pelo método <i>2colorthresh</i> da biblioteca ImageMagick.	20
Figura 3. Imagem gerada pelo método <i>edgefx</i> da biblioteca ImageMagick.	20
Figura 4. Imagem gerada pelo método <i>histog</i> da biblioteca ImageMagick.	21
Figura 5. Imagem gerada pelo método <i>cvCvtColor</i> da biblioteca OpenCV.	22
Figura 6. Imagem gerada pelo método <i>cvSmooth</i> da biblioteca OpenCV.	22
Figura 7. Imagem gerada pelo método <i>cvCanny</i> da biblioteca OpenCV.	23
Figura 8. Demonstração do funcionamento do Tesseract.	25
Figura 9. Modelo de placa automotiva.	27
Figura 10. Layout da aba de reconhecimento.	30
Figura 11. Layout da aba de cadastro de placas.	31
Figura 12. Matriz de exemplo para o algoritmo de segmentação.	32
Figura 13. Ponto inicial identificado e transferido para a matriz auxiliar.	33
Figura 14. Pontos vizinhos.	33
Figura 15. Pontos que serão adicionados na lista.	33
Figura 16. Último ponto a ser transferido.	34
Figura 17. Caractere "E" após a segmentação.	34
Figura 18. Caractere "E" após a remoção da área vazia.	34

LISTA DE TABELAS

Tabela 1. Cores das placas referentes à categoria.	28
Tabela 2. Resultado do teste.	38

LISTA DE GRÁFICOS

Gráfico 1. Resultado da performance do algoritmo de segmentação de caracteres.	38
Gráfico 2. Resultado do teste de precisão da OCR.....	39

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CONTRAN	Conselho Nacional de Trânsito
GIF	Graphics Interchange Format
HP	Hewlett Packard
IDE	Integrated Development Environment
JPEG	Joint Photographic Experts Group
OCR	Optical Character Recognition
OpenCV	Open Source Computer Vision
PNG	Portable Network Graphics
RGB	Red, Green, Blue
RNA	Rede Neural Artificial
RNAs	Redes Neurais Artificiais
SNT	Sistema Nacional de Trânsito
TIFF	Tagged Image File Format

SUMÁRIO

INTRODUÇÃO	15
1 PROCESSAMENTO DE IMAGENS	18
1.1 Definição	18
1.1.1 ImageMagick.....	19
1.1.2 OpenCV.....	21
1.2 Reconhecimento de padrões.....	23
1.3 Reconhecimento Óptico de Caracteres.....	24
1.3.1 Tesseract.....	24
1.4 Considerações Finais.....	25
2 USO DE TÉCNICAS PARA RECONHECIMENTO DE CARACTERES EM PLACAS AUTOMOTIVAS VOLTADO AO CONTROLE DE ACESSO EM LOCAIS PRIVADOS	27
2.1 Placas Automotivas	27
2.2 Objetivos Gerais e Específicos.....	28
2.3 Metodologia	29
2.4 Considerações Finais.....	36
2.5 Testes Efetuados.....	37
2.6 Resultados e Contribuições	39
CONCLUSÃO	41

INTRODUÇÃO

Atualmente, com a constante evolução tecnológica, soluções automatizadas vêm sendo um foco importante em muitas áreas. Essa automação busca melhorar uma solução, principalmente diminuindo o tempo de execução de determinada tarefa, proporcionando melhor aproveitamento para a execução de outras tarefas, ou até mesmo aumento de produtividade.

O reconhecimento de placas automotivas visa, além de diminuir o tempo de execução, também proporcionar maior conforto para o usuário, já que o processo é feito automaticamente, sem que haja interação com o sistema. Essa abordagem requer fundamentos de Inteligência Artificial (IA) para a construção de um agente capaz de identificar padrões através de Reconhecimento Óptico de Caracteres (OCR), além de técnicas de tratamento de imagens que melhoram sua qualidade auxiliando na identificação.

O processo ocorre durante uma filmagem em tempo real, onde são geradas imagens (frames) a partir dos quadros do vídeo gravado. Esses frames são processados na tentativa de identificar uma placa automotiva (realizado por reconhecimento de padrões) e posteriormente, caso haja sucesso, submetidas a uma Rede Neural Artificial (RNA) para a identificação dos caracteres que a compõem, sendo possível reconhecer o veículo através de um banco de dados pré-definido.

Sistemas que abrangem essa funcionalidade “têm a capacidade de identificar e interpretar os códigos alfanuméricos, para que os seus utilizadores possam consultar a informação guardada nos dados” (TOMPKINS, 2003).

O escopo principal concentra-se no desenvolvimento de um algoritmo para a segmentação dos caracteres impressos em uma placa automotiva, tendo como auxílio, bibliotecas para processamento de imagens e reconhecimento de padrões.

Para a implementação do projeto é necessário um embasamento teórico sobre a principal área envolvida nas funcionalidades a serem implementadas, Processamento de Imagens, que recebe um espaço no decorrer desse trabalho envolvendo suas principais técnicas, bibliotecas e exemplos, além de pesquisas focadas em uma solução para o problema proposto.

Motivação e Justificativa

Em ambientes privados como hotéis, condomínios e estacionamentos é importante a

identificação dos veículos, permitindo, por meio de um monitoramento, um controle mais apurado do fluxo, além de prover maior segurança para quem os frequenta. Isso é possível pois, a Computação Gráfica possui a subárea de reconhecimento de padrões, onde técnicas de Inteligência Artificial são aplicadas com o intuito de encontrar uma característica definida em uma imagem, como as placas dos automóveis.

O reconhecimento de caracteres proporciona a extração de informações de uma imagem, possibilitando que sejam utilizadas para a automatização de uma tarefa como por exemplo, em comparações com bases de dados.

Objetivos Gerais

Tem-se por objetivo o estudo de técnicas de reconhecimento de padrões e processamento de imagem, visando uma solução aceitável para a identificação e reconhecimento de placas automotivas (somente veículos privados) possibilitando o controle de acesso de um veículo em determinado local privado.

O propósito desse trabalho é construir um sistema para controle de acesso com reconhecimento de caracteres em placas automotivas para a avaliação dos métodos de processamento de imagem e reconhecimento óptico de caracteres (Tesseract-OCR) utilizados, além do desenvolvimento de um algoritmo para segmentação dos caracteres.

Vale destacar que o sistema não possui interação com nenhum componente externo, com exceção de câmeras utilizadas na captura da imagem para reconhecimento.

Organização do Trabalho

Esse trabalho está organizado em 6 (seis) partes: (1) Introdução, (2) Processamento de Imagens, (4) Projeto, (5) Conclusão e (6) Referências. A relação abaixo descreve com mais detalhes o contexto de cada uma delas:

1. Introdução: Parte do projeto que dá início ao tema sugerido, descrevendo problemática, motivação e objetivos.
2. Processamento de Imagem: No denominado Capítulo 2, o objetivo continua no referencial teórico apresentando a segunda funcionalidade fundamental para o desenvolvimento do projeto. Essa parte do trabalho conta com teorias relacionadas a Processamento de Imagens, demonstrando as bibliotecas ImageMagick e OpenCV com alguns exemplos de sua funcionalidade, e Reconhecimento de Padrões, mais especificamente, a OCR Tesseract.
3. Projeto: Esse capítulo leva o nome do trabalho, pois, diz respeito ao seu desenvolvimento como um todo. Iniciando com as teorias relacionadas às características das placas automotivas, linguagens de programação e banco de

dados e seguindo com motivação, objetivos, implementação, testes e seus respectivos resultados em relação ao tema proposto.

4. Conclusão: Espaço reservado para conclusões e trabalhos correlatos, comparando-os e identificando diferenciais em relação ao desenvolvido.
5. Referencias: Todo o material de estudo, como livros, artigos, páginas Web que auxiliaram na construção desse trabalho, ficaram registrados nesse capítulo.

1 PROCESSAMENTO DE IMAGENS

Esse capítulo envolve conceitos sobre técnicas para processamento de imagens, incluindo exemplos de ferramentas e demonstração de seu funcionamento. Tais conceitos são utilizados para avaliar os itens descritos, possibilitando a escolha de uma ferramenta adequada para o desenvolvimento do projeto, visando a eficiência do mesmo.

O foco principal se mantém na biblioteca *Open Source Computer Vision* (OpenCV) que se destaca por sua variedade de funcionalidades, permitindo a realização do reconhecimento de padrões de maneira simplificada, eficiente e elegante; e Reconhecimento de Padrões voltado para a utilização de Reconhecimento Óptico de Caracteres (OCR).

1.1 Definição

De acordo com Jayaraman, Esakkirajan e Veewakumar. (2009, p.43), imagens, no âmbito computacional, são compostas por elementos chamados *pixels*, sendo esses a menor amostra de uma imagem que representa o brilho em um ponto. Cada pixel é responsável por armazenar os dados relacionados ao ponto da imagem representado, de acordo com o sistema de cor utilizado.

O Processamento de Imagens envolve técnicas baseadas em fórmulas matemáticas capazes de alterar as propriedades de uma imagem como por exemplo, tamanho, nitidez, brilho, contraste, cor, entre outras.

De acordo com Petrou C. e Petrou M. (2010, p.1), o processamento de imagens é concentrado em três problemas principais: (1) diagnóstico e codificação para facilitar transmissão, impressão e armazenamento, (2) aprimoramento e restauração afim de facilitar a interpretação e (3) segmentação e descrição como estágio inicial para visão computacional.

Abordando esse tema, duas ferramentas são apresentadas: ImageMagick e OpenCV, utilizando a Figura 1 como base para os tratamentos realizados.

Figura 1. Imagem para demonstração de processamento de imagens.



Fonte: Próprio autor.

1.1.1 ImageMagick

ImageMagick é um conjunto de software para criar, editar, constituir ou converter imagens que possibilitam leitura e escrita de uma grande variedade de imagens, incluindo GIF, JPEG, PNG e TIFF. Pode ser usada para alterar as propriedades das imagens. (IMAGEMAGICK, 1999).

Com suporte para várias linguagens como: C, C++, Java, Ruby, Python, etc., ImageMagick é uma poderosa ferramenta para ser aplicada em soluções que necessitam realizar alguma alteração nas propriedades das imagens, ou até mesmo criação de uma nova imagem.

Seguindo a ideia, alguns *scripts* são apresentados com o intuito de prover um pouco de conhecimento sobre a capacidade dessa biblioteca.

- *2colorthresh*: Também conhecido como binarização, converte uma imagem qualquer para uma imagem semelhante contendo somente 2 cores: preto e branco. A conversão é feita através de uma fórmula matemática que, quando aplicada em um pixel tem a capacidade identificar o nível de cor e transformá-lo em uma das duas cores. A Figura 2 ilustra essa transformação. Outros *scripts* relacionados são: *fuzzythresh*, *trianglethresh*, *ptilethresh*, *kapurthresh*, *otsuthresh*, entre outros.

Figura 2. Imagem gerada pelo método *2colorthresh* da biblioteca ImageMagick.



Fonte: Próprio autor.

- *Edgefx*: Tem a capacidade de identificar bordas na imagem através da mudança da tonalidade. Esse algoritmo pode ser usado para facilitar a segmentação de imagens, conforme ilustra a Figura 3.

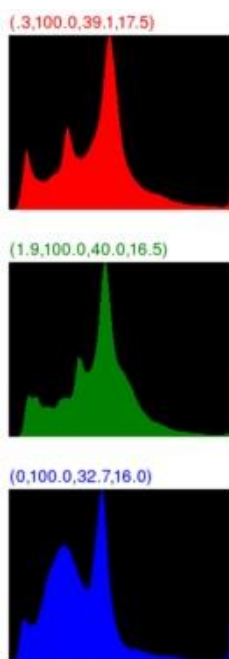
Figura 3. Imagem gerada pelo método *edgefx* da biblioteca ImageMagick.



Fonte: Próprio autor.

- *Histog*: Basicamente, conta os *pixels* (pontos) referentes a uma tonalidade relacionados a determinada linha ou coluna da imagem e do valor obtido, gera um gráfico. A Figura 4 ilustra essa funcionalidade.

Figura 4. Imagem gerada pelo método *histog* da biblioteca ImageMagick.



Fonte: Próprio autor.

1.1.2 OpenCV

OpenCV (*Open Source Computer Vision*) é uma biblioteca com licença livre para uso acadêmico e comercial. Construída em C/C++, disponibiliza interfaces para C, C++, Java, Python, entre outras e suporta os Sistemas Operacionais Windows, Linux, Mac OS, iOS, e Android. Projetada para aplicações de tempo real, podendo tirar proveito de multiprocessamento (OPENCV, 2014).

Utilizada em processamento de vídeos e imagens, possui uma grande variedade de algoritmos de visão computacional incluindo imagens médicas, segurança, interface com o usuário, calibração de câmera, visão estéreo e robótica (OPENCV, 2014). Devido a essa quantidade de algoritmos e suas diferentes funcionalidades, o OpenCV torna-se uma ferramenta robusta para reconhecimento de padrões em imagens em comparação a técnicas manuais (aplicação de filtros e algoritmo de reconhecimento).

Possui código aberto, sendo possível realizar alterações em suas funções, permitindo um melhor ajuste para uma determinada finalidade. Também é possível contribuir para a construção dessa biblioteca que está alocada em um repositório do *Git*.

Conforme demonstrado no item anterior, o OpenCV também possui métodos que possibilitam a transformação das propriedades das imagens. Para demonstrar esse funcionamento, os métodos *cvCvtColor*, *cvSmooth* e *cvCanny* são explicados e ilustrados a

seguir:

- *cvCvtColor*: Esse método tem a função de converter os níveis de cores de uma imagem para outra ou sobrescrever a imagem original. O novo nível de cor é definido de acordo com um parâmetro indicado na chamada do método. A Figura 5 ilustra um tratamento de conversão RGB para escala de cinza.

Figura 5. Imagem gerada pelo método *cvCvtColor* da biblioteca OpenCV.



Fonte: Próprio autor.

- *cvSmooth*: A finalidade desse método é suavizar a imagem original. Essa funcionalidade é extrema importância no reconhecimento de padrões, pois, faz com que a diferença de tonalidade entre os pixels seja mais suave. Um exemplo dessa funcionalidade está ilustrado na Figura 6.

Figura 6. Imagem gerada pelo método *cvSmooth* da biblioteca OpenCV.



Fonte: Próprio autor.

- *cvCanny*: Tem seu comportamento parecido com o método *edgefx* do ImageMagick, porém, com algumas diferenças. Canny é um tratamento do

OpenCV que tem a capacidade de detectar bordas de contorno, convertendo os pontos que representam uma borda em branco e o restante em preto. A Figura 7 demonstra uma imagem após a utilização do método *cvCanny*.

Figura 7. Imagem gerada pelo método *cvCanny* da biblioteca OpenCV.



Fonte: Próprio autor.

1.2 Reconhecimento de padrões

Reconhecimento de padrões é uma técnica aplicada com o objetivo de reconhecer semelhanças pré-definidas por um modelo entre dois ou mais objetos distintos. De acordo com Murty e Devi (2011, p. 1), assim como Pal A. e Pal S. (2001, p. 2), essa técnica é descrita como a classificação de características de objetos.

O reconhecimento utilizando Redes Neurais Artificiais (RNAs) é indicado quando se tem um ambiente limitado, preferencialmente pequeno, pois, o verdadeiro problema do reconhecimento “é o de gerar uma teoria que especifica a natureza dos objetos, de tal maneira que uma máquina será capaz de identificá-los de forma robusta” (PANDYA; MACY, 1995). De acordo com Braga, Carvalho e Ludermir (2012, p. 198), é usada para complementar técnicas clássicas, geralmente envolvem reconhecimento de caracteres escritos à mão, compactação de dados, codificação, reconhecimento de fala e análise de componentes independentes, entre outras.

Além das redes neurais, existem dois métodos principais de reconhecimento: (1) estatístico e (2) sintático.

1. Um padrão é definido por um conjunto de características e a classificação é feita por meio de teorias estatísticas com a finalidade de estabelecer limite entre eles (PANDYA; MACY, 1995).

2. Usa ferramentas e técnicas de linguagem computacional, tais como gramáticas e analisadores, para especificar e busca de padrões em dados (SEARLS; DONG, 1993).

1.3 Reconhecimento Óptico de Caracteres

Reconhecimento Óptico de Caracteres (OCR) é uma técnica de reconhecimento de padrões aplicada com o intuito de realizar a detecção dos caracteres de determinado ambiente visual, como imagens e vídeos. OCR converte uma palavra ou caracteres de uma imagem em códigos de caracteres como ASCII ou Unicode (GOVINDARAJU; SETLUR, 2009).

Uma aplicação comum com uso de OCR é o scanner que identifica o caractere na folha e o transfere em forma de bits para o computador, possibilitando que o usuário leia, pesquise, busque, insira, remova e edite o texto reconhecido.

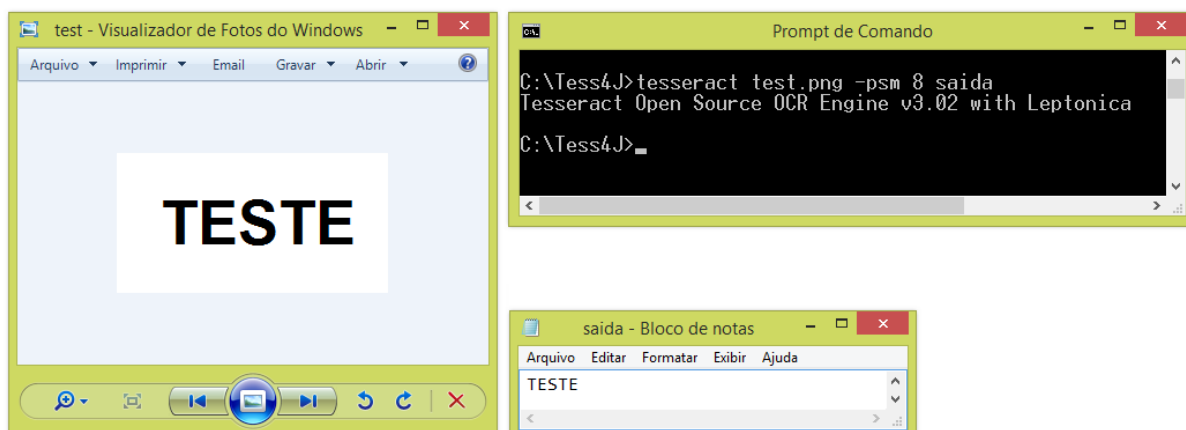
1.3.1 Tesseract

Tesseract é uma OCR desenvolvida pela Hewlett Packard (HP) entre 1984 e 1994. A partir dessa data, passou a ser aperfeiçoada pelo Google, onde maior parte do trabalho se concentrou entre 1995 à 2005, continuando até hoje.

De acordo com a documentação (TESSERACT-OCR, 2014), é provavelmente a OCR *open source* com maior precisão disponível atualmente, podendo realizar a leitura de uma grande variedade de formatos de imagens, convertendo seu conteúdo para mais de 60 línguas.

O Tesseract é executado pela linha de comando (Prompt de Comando), recebendo como parâmetro a imagem de entrada, o tipo de leitura (no caso, -psm 8) e o nome do arquivo a ser gerado com a resposta do reconhecimento. A Figura 8 ilustra o reconhecimento do texto de uma imagem.

Figura 8. Demonstração do funcionamento do Tesseract.



Fonte: Próprio autor.

Essa ferramenta permite, além de configurações especificando linguagem utilizada, caracteres aceitáveis, tipo de leitura (linha única, colunas, etc.), também a realização de treinamento para que a taxa de acerto dos caracteres seja mais eficiente. O treinamento é feito manualmente, onde o usuário fica encarregado de disponibilizar amostras de imagens contendo caracteres para reconhecimento e sua respectiva resposta desejada para gerar uma “biblioteca” de caracteres.

O treinamento descrito anteriormente é caracterizado como supervisionado, pois apresenta entradas e saídas desejadas para a realização do mesmo. Quando aplicado em cenários pequenos, como por exemplo, reconhecimento de dígitos das cartas de um baralho, tem maior chance de sucesso, pois existem poucas variações (mesma fonte e tamanho).

1.4 Considerações Finais

Esse capítulo abordou os seguintes conceitos do Processamento de Imagens:

- O Processamento de Imagens envolve técnicas que transformam as características das imagens. Duas bibliotecas com essa funcionalidade foram apresentadas: ImageMagick e OpenCV.
- Tanto a biblioteca ImageMagick como OpenCV são gratuitas, possuem código aberto e são aplicáveis em diversas plataformas e linguagens.
- Reconhecimento de padrões é uma técnica capaz de detectar padrões nas características da imagem, se classificando em três métodos, redes neurais, estatístico e sintático.
- O Reconhecimento Óptico de Caracteres funciona como um analisador de características, sendo assim classificado como um reconhecedor de padrões.

- Tesseract é uma OCR gratuita desenvolvida pela HP que atualmente está sob os cuidados do Google. Possui a capacidade de treinamento para identificar novos padrões ou até mesmo melhorar o reconhecimento de um padrão existente.

2 USO DE TÉCNICAS PARA RECONHECIMENTO DE CARACTERES EM PLACAS AUTOMOTIVAS VOLTADO AO CONTROLE DE ACESSO EM LOCAIS PRIVADOS

Esse capítulo apresenta as realizações do trabalho, envolvendo as técnicas listadas nos capítulos 1, explicando motivo de escolhas e descrevendo o funcionamento total do sistema desenvolvido.

Inicialmente, apresentam-se as especificações do Conselho Nacional de Trânsito (CONTRAN) referente às placas automotivas, visando a percepção do ambiente que o projeto será aplicado.

Nesse capítulo também são demonstrados os objetivos e toda a metodologia utilizada para o desenvolvimento do projeto.

2.1 Placas Automotivas

O órgão brasileiro responsável por coordenar os órgãos do Sistema Nacional de Trânsito (SNT) é o CONTRAN. Compete a ele, realizar tarefas relacionadas à normas, regulamentos e padronização da política de trânsito.

O Art. 1 da resolução 231 de 15 de março de 2007 estabelece que cada veículo deve ser identificado por placas dianteiras e traseiras, contendo 7 (sete) caracteres alfanuméricos, sendo os três primeiros, letras e os 4 restantes, números (BRASIL, 2007).

Conforme estabelecido pelo anexo dessa resolução, a tipologia dos caracteres das placas e tarjetas devem utilizar a fonte *Mandatory*. A Figura 9 ilustra um modelo de placa automotiva.

Figura 9. Modelo de placa automotiva.



Fonte: Brasil (2007).

Complementando as especificações citadas, o Item 5.1 do anexo da resolução 231 de 15 de março de 2007 (BRASIL, 2007) estabelece as cores das placas referentes à categoria do veículo conforme demonstrado na Tabela 1.

Tabela 1. Cores das placas referentes à categoria.

Categoria	Cor do Fundo	Cor dos Caracteres
Particular	Cinza	Preto
Aluguel	Vermelho	Branco
Experiência Fabricante	Verde	Branco
Aprendizagem	Branco	Vermelho
Coleção	Preto	Cinza
Oficial	Branco	Preto
Missão Diplomática Corpo Consular Organismo Internacional Acordo Cooperação Internacional	Azul	Branco
Representação	Preto	Dourado

Fonte: Brasil (2007).

O projeto desenvolvido leva em consideração somente o padrão de cores de veículos particulares. Uma implementação contemplando todos os modelos descritos na Tabela 1 deveria ser capaz de identificar formas de tonalidades diferentes ou submeter-se a um processamento sujeito a isolar as tonalidades.

2.2 Objetivos Gerais e Específicos

Tem-se por objetivo o estudo de técnicas de reconhecimento de padrões e tratamento de imagem, visando uma solução aceitável para a identificação e reconhecimento de placas automotivas (somente veículos privados) possibilitando o controle de acesso de um veículo em determinado local privado.

Consideram-se objetivos específicos no projeto:

- Implementar um algoritmo que reconheça a placa do veículo capturada através de vídeo;
- Desenvolver um algoritmo para segmentação dos dígitos possibilitando a classificação em letras ou números, visando melhorar o desempenho do reconhecimento.

- Promover maior conforto para o usuário do serviço, uma vez que, não será necessário interagir diretamente com um controlador de acesso, acarretando em um menor tempo de execução dessa tarefa.

2.3 Metodologia

Para o desenvolvimento do projeto sugerido, foi realizado um estudo sobre as tecnologias envolvidas no assunto com o intuito de buscar uma solução com alta performance visando desempenho e o curto tempo para desenvolvimento, dentre elas destacam-se a linguagem de programação Java, a biblioteca OpenCV e a OCR Tesseract.

O critério para a escolha da linguagem foi estabelecido pela facilidade de desenvolvimento levando também em consideração o conhecimento do autor sobre a mesma. Java se destaca por ser a segunda linguagem mais usada no mundo, além de possuir uma ampla comunidade de suporte facilitando assim o desenvolvimento e o esclarecimento de dúvidas.

A biblioteca ImageMagick era considerada a mais poderosa até o momento do início da pesquisa, sendo ela empregada juntamente com a linguagem Ruby (desenvolvimento já realizado pelo autor). Porém com um estudo pouco mais aprofundado foi possível notar o imenso poder da biblioteca OpenCV, descoberta logo no início da pesquisa sobre reconhecimento de padrões e utilizada frequentemente por autores que desenvolvem projetos no âmbito de Processamento de Imagens.

O reconhecimento dos caracteres segmentados ficou por responsabilidade da OCR Tesseract, que se mostra de grande qualidade sendo considerada uma das mais precisas gratuitas no momento.

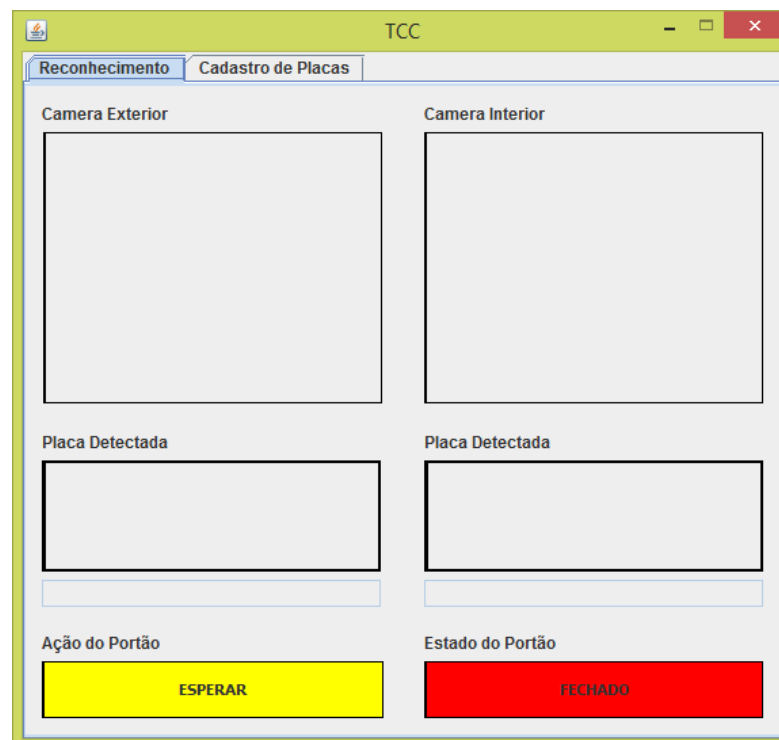
O projeto foi implementado utilizando duas câmeras onde, em um ambiente real, a primeira representaria a câmera fixada do lado externo do local e a segunda, do lado interno. Esse conceito foi adotado para evitar que fosse usado tempo para o fechamento do portão, diminuindo assim, o risco de acidente em relação ao portão e o veículo que utiliza o sistema.

Levando em consideração essa estrutura, o veículo, ao se aproximar do portão, é imediatamente identificado e caso a placa reconhecida esteja cadastrada no banco de dados do sistema e com acesso permitido, o portão se abre. Ao atravessar o portão e ser reconhecido pela outra câmera, o sistema identifica se foi a última placa a abrir o portão, e caso seja, o mesmo é fechado. Essa validação foi implementada com o intuito de evitar que um veículo abra o portão e poucos instantes depois, enquanto ainda está passando, outro tente abrir, ocasionando o fechamento.

É importante lembrar que não ocorre nenhuma interação com meios físicos externos, portanto, o portão é simulado através de um botão na interface do sistema que identifica o estado do mesmo, sendo estes, aberto, fechado ou parado.

O *layout* do projeto é dividido em duas abas referentes às funcionalidades do sistema, reconhecimento e cadastro de placas. A Figura 10 e Figura 11 ilustram essas abas, respectivamente.

Figura 10. Layout da aba de reconhecimento.



Fonte: Próprio autor.

Figura 11. Layout da aba de cadastro de placas.

ID	Placa	Acesso
2	ABC1234	true
4	AAA1111	true
5	CHU2012	true

Fonte: Próprio autor.

Na aba de reconhecimento é possível notar os frames (janelas) que representam as imagens capturadas das câmeras, onde as placas identificadas são visualizadas com seus respectivos caracteres reconhecidos pela OCR, a ação que o portão está realizando e a condição do mesmo. A aba de cadastro de placas demonstra as informações sobre as placas registradas no sistema e contempla as opções para editar, excluir e adicionar novas placas.

Ao ser iniciado, o sistema fica em um *loop* infinito capturando a imagem da câmera, chamados de *frame*, e passa por um identificador construído com auxílio da biblioteca OpenCV, que realiza alguns tratamentos e processamentos, e indica se há ou não alguma placa no *frame*.

O identificador consiste em transformar a imagem original RGB para uma imagem em escala de cinzas, aplicar os filtros *smooth* para suavizar a imagem e *canny* para detectar as bordas e executar um algoritmo para reconhecimento de contornos com formato determinado, no caso, retângulo, que para ser considerado uma placa, deve satisfazer algumas condições:

- Ter quatro vértices;
- Ter uma área de contorno identificado maior que 2000 (dois mil) *pixels*;
- Ser convexa; e
- Ter a proporção entre 2.2 e 3.4 em relação à altura com a largura.

Ao ser identificada, a placa pode estar em uma posição não muito favorável para o

reconhecimento, com uma certa perspectiva. Devido a esse problema, o contorno identificado é submetido à uma função que determina o ângulo de perspectiva de acordo com as coordenadas dos vértices e a corrige o máximo possível, mas dificilmente cem por cento.

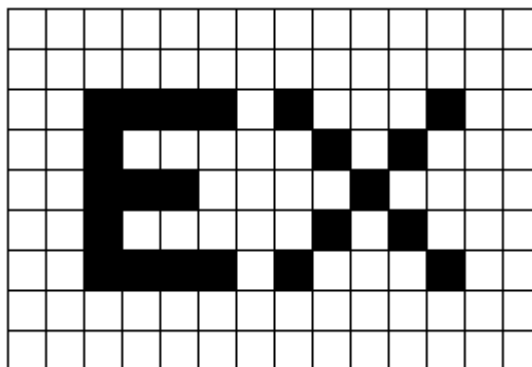
Após corrigida a perspectiva, a placa passa por um processo de binarização e segmentação de caracteres que converte a imagem com canais RGB (*red*, *green* e *blue*) em uma imagem binária (imagem contendo somente as cores preta e branca), possibilitando a separação dos mesmos e conseqüentemente a classificação em letra ou dígito.

O algoritmo que realiza a binarização consiste em transformar a imagem original em uma matriz de *pixels* e percorre-la, utilizando uma matriz auxiliar zero do mesmo tamanho da matriz original. Para cada posição lida, a tonalidade do *pixel* é identificada e atribuído o valor 0 ou 1 para a matriz auxiliar na mesma posição.

A segmentação dos caracteres é realizada a partir da matriz gerada no processo descrito acima, percorrendo-a de cima para baixo, da esquerda para a direita. Essas direções asseguram que os dígitos sejam capturados na seqüência correta mesmo em caso do processamento ser aplicado em uma imagem com uma certa rotação ou perspectiva.

O processo de segmentação se resume em um algoritmo que identifica um ponto com valor 1, transfere-o para uma nova matriz, incrementa um contador e repete o processo com os pixels “vizinhos” com valor 1. Detalhadamente o algoritmo implementado é explicado a seguir utilizando a Figura 12 como matriz de exemplo, onde, os campos com a cor preta representam o valor 1 e os campos em branco, o valor 0:

Figura 12. Matriz de exemplo para o algoritmo de segmentação.



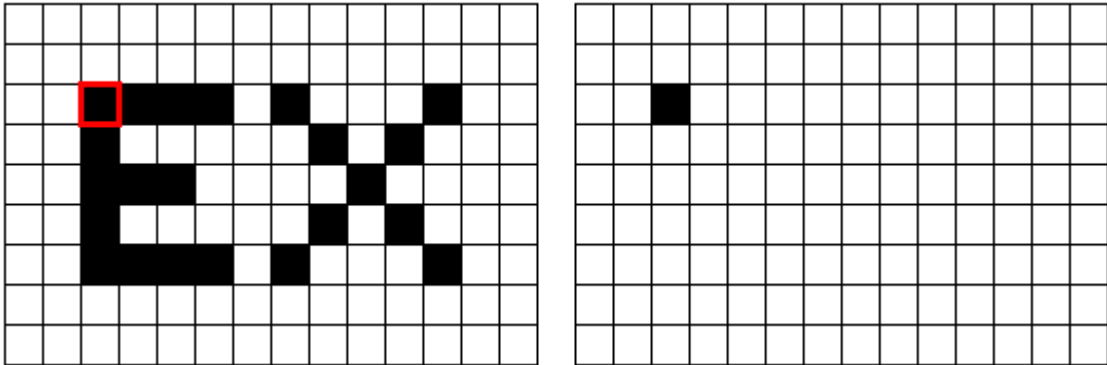
Fonte: Próprio autor.

A matriz é percorrida, primeiramente na posição vertical (de cima para baixo) e depois na posição horizontal (da esquerda para a direita), assegurando que os caracteres sejam segmentados na mesma seqüência que devem ser reconhecidos. Esse critério se torna indispensável caso a imagem apresente uma rotação anti-horária.

Seguindo o raciocínio, o ponto inicial ou ponto onde o primeiro valor 1 é identificado

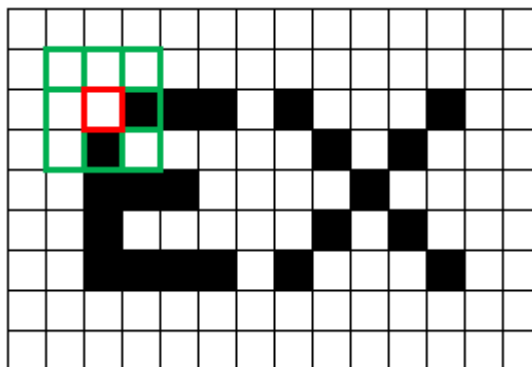
é transferido para uma matriz auxiliar, conforme ilustra a Figura 13 e, posteriormente, os pontos vizinhos são identificados. Caso seu valor seja 1, o ponto é adicionado à uma lista e um contador incrementado. As Figuras 14 e 15 ilustram os pontos vizinhos e os pontos que deverão ser adicionados na lista, respectivamente.

Figura 13. Ponto inicial identificado e transferido para a matriz auxiliar.



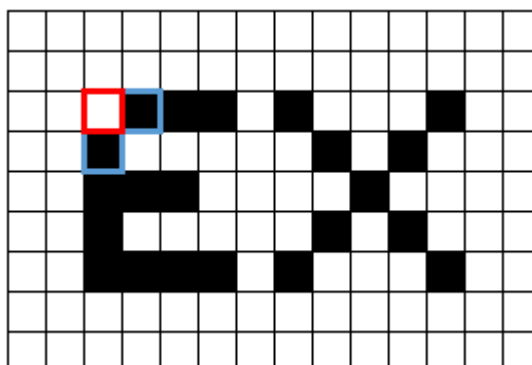
Fonte: Próprio autor.

Figura 14. Pontos vizinhos.



Fonte: Próprio autor.

Figura 15. Pontos que serão adicionados na lista.

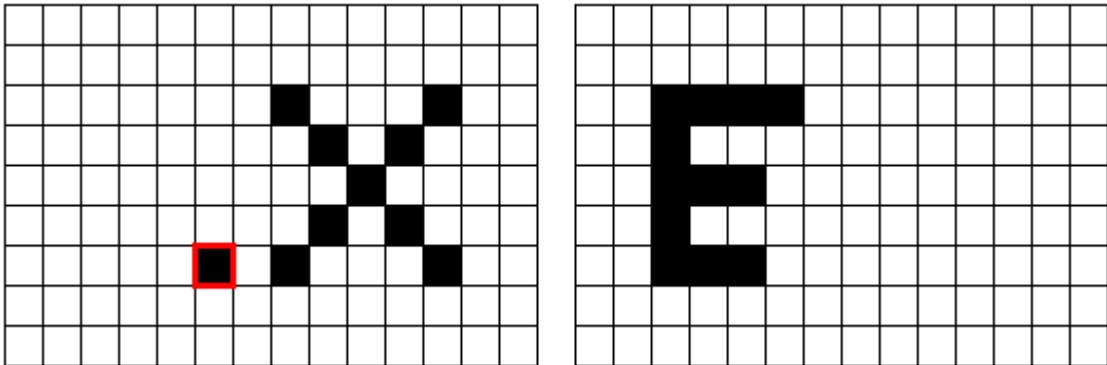


Fonte: Próprio autor.

Para cada ponto existente na lista, o algoritmo repete os procedimentos relacionados às Figuras 13, 14 e 15 até que se encontre vazia, conforme indicado pela Figura 16. Então, busca-se um novo ponto inicial para realizar todos os passos novamente até que não haja mais

pontos com valor 1 na matriz original.

Figura 16. Último ponto a ser transferido.

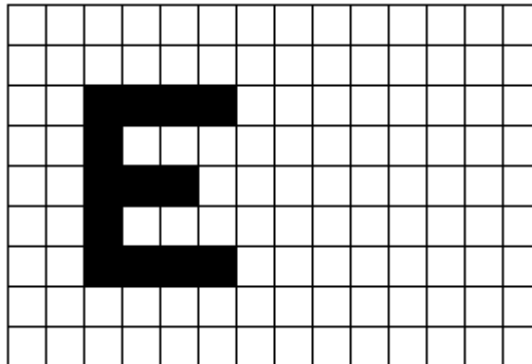


Fonte: Próprio autor.

Após o término do processamento, a matriz auxiliar se resultará em um caractere com uma grande área vazia, onde, um outro algoritmo que detecta as bordas do caractere é aplicado com o intuito de diminuir o tamanho da imagem, tornando o processo de gravação da mesma mais rápido, além de ser fundamental para obter-se uma alta taxa de sucesso no reconhecimento pela OCR.

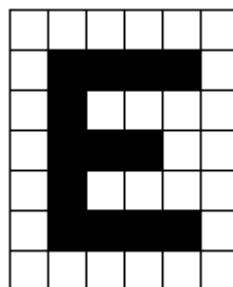
As Figuras 17 e 18 representam o caractere após a segmentação e após a remoção da área vazia, respectivamente.

Figura 17. Caractere "E" após a segmentação.



Fonte: Próprio autor.

Figura 18. Caractere "E" após a remoção da área vazia.



Fonte: Próprio autor.

Vale ressaltar que o caractere somente é considerado válido para reconhecimento caso o contador, que representa o número de pontos contido no conjunto identificado, seja maior que uma constante X.

O algoritmo para remover a área vazia citado anteriormente consiste em identificar os pontos de início e fim do conjunto na matriz e convertê-la em uma imagem final que será utilizada no reconhecimento pela OCR. A imagem gerada possui uma borda de um pixel que é fundamental para realizar o reconhecimento com sucesso pelo Tesseract. Assim, uma nova condição é imposta para identificar se o conjunto é realmente um caractere ou uma “sujeira” da imagem: a altura do caractere deve ser maior que uma outra constante Y.

Devido aos processamentos descritos acima, a imagem original capturada pela câmera resulta em 7 imagens contendo somente os caracteres a serem reconhecidos, possibilitando a classificação dos 3 primeiros como letras e dos 4 últimos como dígitos, conforme indica a resolução citada no item 2.1.

A implementação com a OCR Tesseract substituiu a criação e o treinamento de uma RNA que deveria ser realizado para se alcançar uma alta taxa de sucesso no reconhecimento dos caracteres. Ao invés disso, somente um filtro de configuração é utilizado na chamada da OCR para identificar quais caracteres são aceitáveis em cada posição a ser reconhecida.

Após o reconhecimento dos caracteres impressos na placa automotiva, o valor resultante do processamento é comparado com os valores existentes no banco de dados, verificando se existe alguma placa cadastrada com a sequência reconhecida e se seu acesso é ou não permitido.

Durante o desenvolvimento do projeto, muitas dificuldades foram encontradas, destacando-se algumas como suporte e documentação da biblioteca OpenCV, integração da OCR com o sistema e correção da perspectiva da placa identificada.

Diferentemente das classes comuns do Java, a biblioteca OpenCV não dispõe de uma documentação similar, o que facilitaria o desenvolvimento devido à identificação dos parâmetros a serem atribuídos aos métodos. Essa dificuldade é considerada um fator que conseqüentemente ampliou o tempo para o desenvolvimento do projeto, principalmente em relação à correção da perspectiva da imagem.

Outro problema encontrado, porém, resolvido facilmente devido ao grande apoio da comunidade em casos similares, foi a integração da biblioteca Tesseract com o sistema, pois, o desenvolvimento foi realizado em um dispositivo com barramento de 64 bits e a biblioteca somente oferecia suporte para 32 bits. Após algum tempo de pesquisa, foi possível encontrá-la em compatibilidade com o dispositivo, em um arquivo executável.

O grande diferencial entre a biblioteca e o arquivo executável é que no primeiro caso, a chamada é realizada através de uma classe, funcionando como uma API (*Application Programming Interface*) e a resposta é retornada dessa chamada. No segundo caso, é necessário realizar a chamada por linha de comando e ler o arquivo de texto gerado contendo a resposta. O funcionamento é o mesmo da biblioteca, porém essa manipulação é abstraída com seu uso.

2.4 Considerações Finais

O capítulo 3 apresentou critérios de determinação de tecnologias e ferramentas a serem utilizadas e demonstrou o funcionamento do projeto. Em tópicos, os principais pontos são citados:

- O Art. 1 da resolução 231 de 15 de março de 2007 estabelece padrões fundamentais para o desenvolvimento do projeto, contemplando as características de uma placa automotiva, sendo a mesma expressa por 7 caracteres, 3 letras e 4 números;
- Java foi selecionada como a linguagem mais trivial para o desenvolvimento, devido à popularidade, possuir uma grande comunidade, fator que auxilia no esclarecimento de dúvidas e suporte às bibliotecas utilizadas;
- A principal motivação para o projeto foi a automatização de uma tarefa, visando principalmente um aumento de satisfação e conforto de um usuário do sistema oferecido;
- Identifica-se por objetivo o estudo de técnicas de reconhecimento de padrões e tratamento de imagem, visando o desenvolvimento de um sistema que possibilite o controle de acesso em um determinado local privado;
- Os objetivos específicos se resumem em: implementar um algoritmo que reconheça a placa do veículo; desenvolver um algoritmo para segmentação de caracteres e promover conforto para o usuário do serviço;
- O projeto utiliza duas câmeras, sendo a primeira considerada como fixada no exterior do local e a segunda no interior, o que facilita o controle de abertura e fechamento do portão;
- Cada frame capturado é processado na tentativa de identificação de uma placa automotiva;
- Após a identificação, a placa passa por um processo de correção de perspectiva e segmentação de caracteres;
- A segmentação proposta tem por objetivo a classificação dos caracteres em letras ou números, facilitando a leitura da OCR Tesseract;
- O valor reconhecido da placa é confrontado com os existentes no banco de dados para verificar se o acesso para determinado veículo é ou não permitido;

Trabalhos Correlatos

Reconhecimento de caracteres é uma área que abrangida por muitos trabalhos em diferentes aplicações. Dois trabalhos com o foco do reconhecimento voltado para placas automotivas foram selecionados para uma breve comparação com o trabalho desenvolvido.

Em sua monografia, Schneider (2008), tem como tema o reconhecimento de caracteres através de técnicas de Inteligência Artificial utilizando a linguagem Java. Esse trabalho teve como resultado um software chamado Kapta que utiliza um Banco de Dados MySQL e técnicas de processamento de imagens do MATLAB.

O sistema Kapta utiliza duas Redes Neurais Artificiais, implementadas pelo software MATLAB, para o reconhecimento dos caracteres onde, explica o autor, devem ser reconhecidos um a um. A segmentação dos caracteres é realizada através da função *Histograma*, que possibilita a identificação da localização dos caracteres através dos níveis de cor. Esse processo é descrito no item 1.1.1.

Assim como o presente trabalho, o sistema Kapta também permite o reconhecimento somente de placas automotivas de veículos privados.

Nascimento (2012), propõe a detecção e reconhecimento de placa automotiva com baixo custo. Desenvolvido em C/C++ e tomado como base para a detecção da placa automotiva do presente trabalho, também utiliza a biblioteca OpenCV para processamento de imagem e a OCR Tesseract para reconhecimento dos caracteres.

O sistema desenvolvido por Nascimento (2012), conta com a realização do treinamento da OCR, para apurar o reconhecimento da placa, que é realizado sem a separação dos caracteres.

O trabalho proposto consiste em unir as duas ideias citadas acima, integrando partes das funcionalidades dos mesmos, porém, com o desenvolvimento de novas técnicas para a separação de caracteres.

Do primeiro trabalho relacionado nesse item, foi retirada a somente a ideia da separação dos caracteres, porém, utilizando o método descrito no item 2.3. E do segundo, o algoritmo de detecção foi fundamental para a construção do projeto proposto, levando-o como base, aperfeiçoando o resultado final dessa etapa com a correção de perspectiva.

2.5 Testes Efetuados

Para a validação do projeto, o mesmo foi testado em uma amostra real de 57 veículos de diversas cores de um estacionamento, onde, 27 placas foram identificadas e somente 10

estavam em condições de serem segmentadas. O teste foi executado, por volta das 16:00hs com aproximadamente 2 horas de duração, manualmente, aproximando a câmera do dispositivo de uma placa para a coleta das informações descritas. A Tabela 2 indica os resultados obtidos:

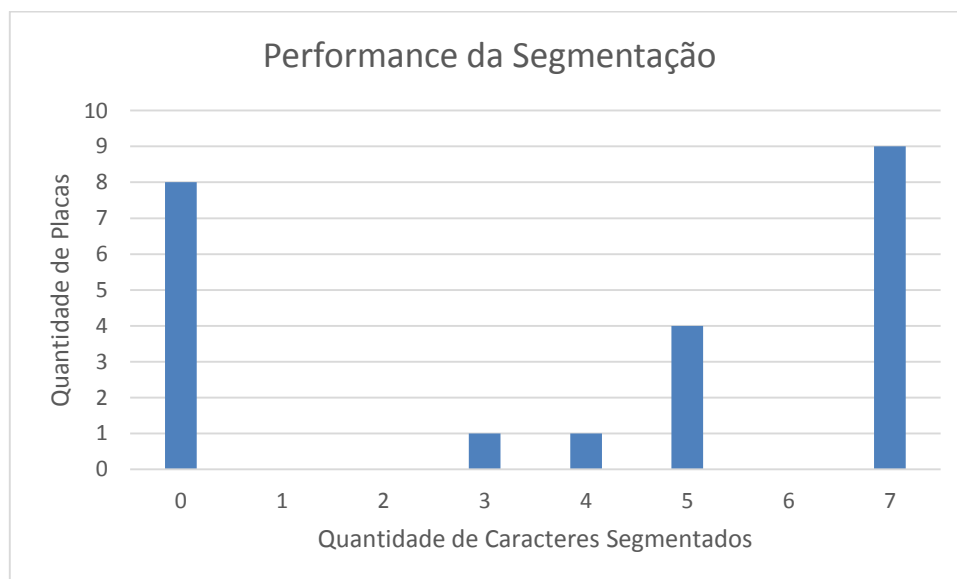
Tabela 2. Resultado do teste.

Cor do veículo	Quantidade	Identificados	Segmentados
Branco	12	5	1
Cinza	7	3	1
Bege	1	0	0
Prata	18	5	1
Preto	16	12	5
Verde	1	1	1
Vermelho	2	1	1
Total	57	27	10

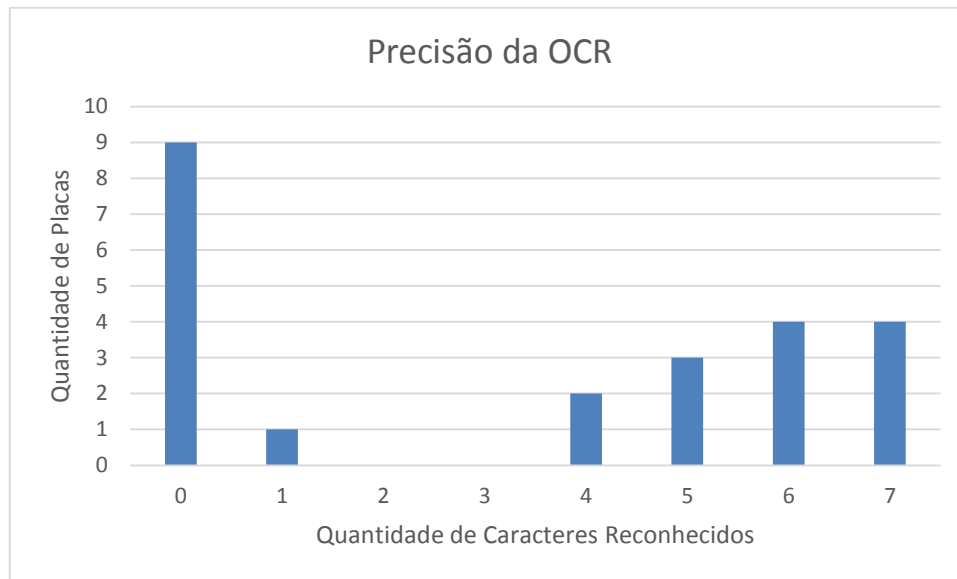
Fonte: Próprio autor.

Tomando como base uma amostra das placas identificadas, um teste para a verificação da precisão da OCR também foi realizado, obtendo-se uma taxa de 84,4% de reconhecimento de caracteres como resultado dessa avaliação. A base contou com 90 caracteres segmentados, sendo 76 deles, reconhecidos com sucesso. Os gráficos 1 e 2 representam a quantidade caracteres segmentados em relação à quantidade de placas e a quantidade de caracteres reconhecidos em relação à quantidade de placas, respectivamente.

Gráfico 1. Resultado da performance do algoritmo de segmentação de caracteres.



Fonte: Próprio autor.

Gráfico 2. Resultado do teste de precisão da OCR.

Fonte: Próprio autor.

Utilizando a taxa de reconhecimento obtida por caracteres, é possível estimar que a taxa de sucesso do reconhecimento da placa por completo é de aproximadamente 30,5% através da teoria da probabilidade de eventos simultâneos, representada pela seguinte fórmula matemática:

$$p(A \cap B) = p(A) \times p(B)$$

Alguns fatores como luminosidade e qualidade da placa, implicam diretamente na qualidade da identificação e principalmente da segmentação dos caracteres. O ideal é que o local a ser instalado as câmeras tenha uma luminosidade razoável, influencia no algoritmo de segmentação, impossibilitando que o mesmo “enxergue” os caracteres, assim como a escassez da mesma não permite a diferenciação do fundo em relação aos caracteres.

2.6 Resultados e Contribuições

Tem-se como resultado um embasamento para a implementação de um sistema mais robusto, considerando os pontos negativos a serem aprimorados, como o algoritmo de identificação e o ajuste do método de segmentação para que assim, possa se tornar um produto final.

Vale salientar que o presente trabalho tem como principal foco uma implementação acadêmica, visando somente estudos e validação do mesmo através da implementação e testes.

São consideradas contribuições desse trabalho, o estudo e apresentação das

ferramentas para processamento de imagens e reconhecimento de padrões, tais como ImageMagick e OpenCV; e Tesseract, respectivamente. Além desses, também é apresentado um algoritmo para segmentação de imagens, podendo ser ajustado para reconhecimento de outros conjuntos, mesmo com características diferentes como cor e tamanho, desde que seus *pixels* estejam conectados ao menos em 1 dos 8 *pixels* vizinhos.

CONCLUSÃO

A ideia inicial do projeto era realizar o desenvolvimento utilizando a linguagem Ruby juntamente com a biblioteca ImageMagick para o processamento de imagens e duas Redes Neurais Artificiais, sendo uma delas voltada para o reconhecimento dos caracteres numéricos e outra para as letras.

Após revisão bibliográfica dos temas envolvidos, observou-se que muitos trabalhos utilizam a biblioteca OpenCV devido ao seu poder computacional no âmbito de processamento de imagens, por esse motivo, o estudo desse módulo do projeto passou a focar o desenvolvimento de uma solução utilizando-a.

Com o estudo se aprofundando na biblioteca OpenCV, verificou-se que a linguagem Ruby não fornecia total suporte à mesma, portanto, ainda sem o conhecimento dos métodos que seriam utilizados, a linguagem selecionada passou a ser Java, que além de possuir uma grande comunidade disponível para esclarecimento de dúvidas e materiais de consulta, também facilita o desenvolvimento de interfaces utilizando a *Integrated Development Environment* (IDE) NetBeans.

Nesse ponto, o escopo do projeto passa a ter como objetivo a implementação de uma solução para reconhecimento de caracteres em placas automotivas utilizando a linguagem Java, a biblioteca OpenCV e duas Redes Neurais Artificiais.

Devido ao escasso tempo para a implementação, a utilização de RNAs passou a ser uma solução não muito viável, pois, haveria a necessidade de implementar, reunir amostras para treinamento e realiza-lo, o que acarretaria um maior consumo de tempo. Levando essa afirmação em consideração, optou-se pela utilização de uma OCR, o Tesseract, implicando em menor tempo de desenvolvimento.

O algoritmo para detecção da placa na imagem foi baseado no algoritmo proposto por Nascimento (2012), aplicando-se uma modificação em relação à perspectiva da imagem. Esse aprimoramento visa remover a rotação da imagem, permitindo que a mesma passe a ter seu ângulo de inclinação e rotação o mais próximo de 0 possível.

Após a detecção, inicia-se o escopo principal do projeto, a segmentação dos caracteres. Esse processo é realizado em 3 etapas: (1) binarização, (2) segmentação e (3) reconhecimento, explicados com detalhes no item 2.3.

Com base nos testes efetuados e nas pesquisas de campo, conclui-se que o algoritmo proposto possui uma taxa de sucesso consideravelmente boa se aplicado em cenários “perfeitos”. O termo perfeito refere-se às placas automotivas de qualidade e luminosidade

adequada.

Vale ressaltar ainda que o projeto possui somente propósitos acadêmicos, o que torna os resultados obtidos, fonte de informação para melhorias e implementações futuras, podendo assim, após retrabalho, tornar-se um produto final.

No decorrer do trabalho, obteve-se algumas dificuldades como principalmente, documentação da biblioteca OpenCV, o que provocou um acréscimo no tempo de desenvolvimento, conseqüentemente, sujeitando uma leve mudança no escopo, porém, mantendo o tema proposto desde o início.

A OCR Tesseract também teve contribuição nas dificuldades do trabalho, devido ao sistema ser desenvolvido em um dispositivo com barramento e sistema operacional de 64 bits, a utilização da OCR tornou-se inviável, uma vez que, sua implementação somente fornecia suporte para sistemas de 32 bits. Com o auxílio da comunidade em dificuldades similares, o problema foi solucionado sem muito impacto no projeto.

Trabalhos Futuros

O reconhecimento apresentado no trabalho possui algumas limitações referentes à tonalidade do veículo a ser reconhecido. Conforme indicado no item Testes Efetuados, caso seja de tonalidade clara, o sistema tem dificuldade na localização da placa.

Tem-se como algumas melhorias para trabalhos futuros:

- Aprimorar o algoritmo de detecção de placas;
- Desenvolver duas Redes Neurais Artificiais, uma para números e outra para letras, incrementando a taxa de sucesso no reconhecimento dos caracteres da placa;
- Disponibilizar reconhecimento de caracteres em placas de motocicletas;
- Sugerir posicionamento da câmera para a identificação de placas de motocicletas;

REFERÊNCIAS

- BRAGA, Antônio de Pádua; CARVALHO, André Ponce de Leon F. de; LUDERMIR, Teresa Bernanda. **Redes Neurais Artificiais: Teoria e aplicações**. 2. ed.: LTC Editora, 2012.
- BRASIL. **Conselho Nacional de Trânsito. Resolução no 231, de 15 de março de 2007**. Estabelece o Sistema de Placas de Identificação de Veículos. Disponível em: <http://www.denatran.gov.br/download/Resolucoes/RESOLUCAO_231.pdf>. Acesso em: 13 mai. 2014.
- GOVINDARAJU, Venu; SETLUR, Srirangaraj (Ranga). **Guide to OCR for Indic Scripts: Document Recognition and Retrieval**. Nova York, 2009. 346 p. Disponível em: <http://books.google.com.br/books?id=WdSR9OJ0kxYC&printsec=frontcover&dq=GOVINDARAJU&ei=EmpiVL_CAofwsge64YDwDw&hl=pt-PT>. Acesso em: 18 jun. 2014.
- IMAGEMAGICK Studio LLC. **ImageMagick: Convert, Edit, or Compose Bitmap Images**. 1999. Disponível em: <<http://www.imagemagick.org/>>. Acesso em: 14 jun. 2014.
- JAYARAMAN, S; ESAKKIRAJAN, S; VEEWAKUMAR T. **Digital Image Processing**. New Delhi: Tata McGraw Hill Education, 2009. 723p.
- MURTY, M. Narashima; DEVI, V. Susheela. **Pattern Recognition: An Algorithmic Approach**. Springer Science & Business Media, 2011. 275 p. Disponível em: <<http://books.google.com.br/books?id=uBWD3HnzYFUC&dq=murty&hl=pt-PT>>. Acesso em: 15 jun. 2014.
- NASCIMENTO, Jean Dias do. **Deteção e reconhecimento de placa automotiva com baixo custo**. Brasília, 2012. 110p. Trabalho de Conclusão de Curso de Engenharia da Computação – Centro Universitário de Brasília. Disponível em: <<http://www.repositorio.uniceub.br/bitstream/235/3665/2/Monografia%20JEAN%20DIAS%202012.pdf>>. Acesso em: 30 ago. 2014.
- OPENCV. 2014. Disponível em: <<http://opencv.org/>>. Acesso em: 14 jun. 2014.
- PAL, Sankar K.; PAL, Amita. **Pattern Recognition: From Classical to Modern Approaches**. Singapura: World Scientific Publishing Co. Pte. Ltd., 2001. 612 p. Disponível em: <<http://books.google.com.br/books?id=taSaltlKBpsC&hl=pt-PT>>. Acesso em: 29 abr. 2014.
- PANDYA, Abhijit S.; MACY, Robert B.. **Pattern Recognition with Neural Networks in C++**. Crc Press, 1995. 432 p. Disponível em: <<http://books.google.com.br/books?id=t-zfAxpQGIC&hl=pt-PT>>. Acesso em: 29 abr. 2014.
- PETROU, Maria; PETROU, Costas. **Image Processing: The Fundamentals**. 2. ed. Willey, 2010. 793 p. Disponível em: <<http://books.google.com.br/books?id=w3BpSIxN9ZYC&printsec=frontcover&dq=petrou&ei=U2ViVLXE0caiswfCuYCWdW&hl=pt-PT>>. Acesso em: 12 jun. 2014.
- SCHNEIDER, Jeison Isair. **Reconhecimento de caracteres através de técnicas de Inteligência Artificial utilizando a linguagem Java**. Rio do Sul. Trabalho de Conclusão de Curso de Sistemas de Informação – Faculdade de Ciência e Tecnologia da Universidade para o Desenvolvimento do Alto Vale do Itajaí. 2008. 65p

- SEARLS, David; DONG, Shan. **A Syntactic Pattern Recognition System For DNA Sequences**. Philadelphia, 1993. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=DB1EB22ECD3B4EA475AE7704535E427C?doi=10.1.1.48.9897&rep=rep1&type=pdf>>. Acesso em: 29 abr. 2014.
- TESSERACT-OCR: An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google. 2014. Disponível em: <<https://code.google.com/p/tesseract-ocr/>>. Acesso em: 25 jun. 2014.
- TOMPKINS, James A. et al. - **Facilities planning**. 3a ed. Nova Iorque: John Wiley & Sons, 2003. Disponível em: <<http://books.google.com.br/books?id=-xBIq6Qm2SQC&printsec=frontcover&dq=TOMPKINS&hl=pt-PT>>. Acesso em: 10 jan. 2014.