

Processamento de Identificação de Placas de Carros

Kim José Koto Murakawa

*Instituto de Ciéncia e Tecnologia
Universidade Federal de São Paulo*

São José dos Campos, Brasil
k.murakawa@unifesp.br

Matheus Santos Volotão

*Instituto de Ciéncia e Tecnologia
Universidade Federal de São Paulo*

São José dos Campos, Brasil
matheus.volotao@unifesp.br

Resumo—O Processamento de Identificação de Placas de Carros tem sido muito aplicado no nosso cotidiano por possuir diversas utilidades como fiscalizações, monitoramento do tráfego, estacionamentos e nas administrações de pedágio. Neste trabalho utilizaremos um conjunto de dados com 52 imagens de veículos na qual iremos aplicar algumas técnicas aprendidas na unidade curricular de Processamento de Imagens, afim de descobrir o quanto eficaz ele será para detectar diversos tipos de placas e obter assim uma taxa de precisão e de acerto.

Palavras-chave: Detecção, Reconhecimento, Placas Veiculares, Processamento de Imagens

I. INTRODUÇÃO E MOTIVAÇÃO

O Brasil, segundo o Mapa de Motorização de 2019 do Observatório das Metrópoles, indicou que entre 2008 e 2018 houve um crescimento de mais 28 milhões de automóveis, demonstrando um aumento na taxa de motorização (definida pelo total de automóveis dividida por cem habitantes) que passou de 19,6 para 31,5 no país dando a entender que a motorização individual possui uma presença dominante sobre os demais meios de transporte. Além de que as dezessete principais regiões metropolitanas são responsáveis por 40% desse crescimento, indicando que as cidades que enfrentavam os piores problemas de trânsito e transporte continuaram recebendo boa parte da carga de novos automóveis povoando ainda mais as ruas nos últimos anos [1].

Segundo a Polícia Rodoviária Federal (PRF), no período de 2009 a 2019, foram aplicadas mais de 5,7 milhões de multas, apontando um aumento de 88% se comparado ao início do intervalo. O professor da Universidade de Brasília, David Duarte Lima, ressaltou que durante esse tempo houve um grande aumento de veículos circulando, o que consequentemente aumenta o número de pessoas cometendo infrações [2].

Porém nesse mesmo período, segundo a Confederação Nacional do Transporte (CNT), houve uma redução de aproximadamente 8.4% de acidentes em rodovias federais. O diretor na consultoria Mobilidade Segura EAD, Cassiano Ferreira Novo, explica que atualmente, os órgãos públicos possuem tecnologias melhores para a análise e investigação de ocorrências de trânsito, fazendo com que as ações sejam mais precisas em relação à fiscalização e infraestrutura, gerando uma fiscalização mais efetiva, com mais multas e menos acidentes [2].

Dessa forma, o reconhecimento de placas veiculares se tornou de grande relevância para a administração de tráfego,

apresentando um objetivo importante em diversas aplicações como na garantia da execução das leis de trânsito, implementação automática de multas e administração dos pedágios.

O aumento de automóveis nas vias urbanas, tornou a tarefa de controlar tráfego em rodovias mais difícil devido a grande quantidade de dados a serem analisados e reportados aos órgãos competentes. Dessa forma, este trabalho visa auxiliar o trabalho dos agentes de trânsitos no gerenciamento de veículos através de uma técnica para a processamento de identificação de placas de carros em imagens capturadas por câmeras convencionais.

Logo, devido a dificuldade de encontrar sistemas eficientes e de baixo custo, este documento apresenta um sistema simples e funcional capaz de detectar e realizar a identificação de placas de carros, possibilitando o gerenciamento de forma mais automática nas atividades relacionadas a automóveis. Por meio das técnicas aprendidas de processamento de imagens, conseguimos de forma satisfatória detectar e interpretar as placas dos carros, embora apresente obstáculos como localização, iluminação e inclinação.

As próximas seções estão organizadas da seguinte maneira: A fundamentação teórica citando os principais conceitos teóricos envolvidos no trabalho referentes aos algoritmos de Processamento de Imagens estão na Seção II. Nossa metodologia de trabalho, apresentando nossos objetivos, materiais e métodos utilizados estão na Seção III. Na Seção IV, apresentamos os resultados obtidos. E, na Seção V são apresentadas nossas considerações finais.

II. FUNDAMENTAÇÃO TEÓRICA

A. Imagem Monocromática

Uma imagem monocromática é descrita matematicamente como uma função $f(x, y)$ da intensidade luminosa, tendo como valor em qualquer ponto de coordenadas espaciais (x, y) , proporcional ao nível de cinza da imagem naquele ponto [14]. A figura 1 mostra uma imagem monocromática da Lenna e a convenção utilizada pela biblioteca *OpenCV* para o par de eixos (x, y) .

A função $f(x, y)$ é tratado como o produto da interação entre a iluminância $i(x, y)$ - a quantidade de luz que incide sobre o objeto - e as propriedades de refletância do objeto, que podem ser representadas pela função $r(x, y)$ exprimindo a fração de luz incidente que o objeto vai transmitir ou refletir



Figura 1. Figura da Lenna em Escala Cinza (Fonte: Autor)

ao ponto (x, y) [14]. Ela pode ser representada pela seguinte fórmula matemática:

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (1)$$

com $0 < i(x, y) < \infty$ e $0 < r(x, y) < 1$. A intensidade de uma imagem monocromática f nas coordenadas (x, y) é chamada de nível de cinza (ou tom de cinza) (L) da imagem naquele ponto. Este valor estará no intervalo:

$$L_{min} \leq L \leq L_{max}$$

sendo L_{min} e L_{max} valores positivos e finitos.

O intervalo $[L_{min}, L_{max}]$ é chamado de escala de cinza da imagem. Sendo possível deslocar este intervalo para o intervalo dos inteiros $[0, W]$, onde $L = 0$ significa pixel preto e $L = W - 1$ representa pixel branco [14].

B. Captura e Digitalização da Imagem

A captura, ou aquisição, de uma imagem se trata do processo de conversão de uma cena real tridimensional em uma imagem analógica.

Esse imagem analógica obtido à saída do dispositivo de aquisição deve ser submetido a uma discretização espacial e em amplitude para tomar o formato desejável ao processamento computacional. Essa amostragem converte a imagem analógica em uma matriz de M por N pontos, cada qual denominado pixel (ou elemento de imagem):

$$\begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (2)$$

quanto maior forem os valores de M e N maior será sua resolução [14].

C. Canal RGB

Imagens que possuem informações em intervalos ou bandas distintas de frequência utilizam-se uma função $f(x, y)$ para cada banda. É o caso de imagens coloridas padrão **RGB**, que são formadas pela informação de cores primárias aditivas, de suas

siglas **R** - Red (vermelho), **G** - Green (verde) e **B** - Blue (azul).

O modelo RGB é o mais utilizado por câmeras e monitores de vídeo, sendo baseado num sistema de coordenadas cartesianas, pode ser representado visualmente como um cubo, conforme a Figura 2, onde três de seus vértices são as cores primárias, outros três as cores secundárias, o vértice junto à origem é o preto e o mais afastado da origem corresponde à cor branca.

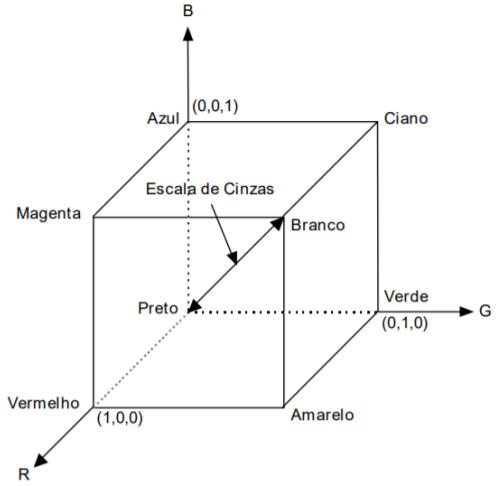


Figura 2. Modelo RGB [14]

Por conveniência, geralmente assume-se que os valores máximos de R, G e B estão normalizados na faixa de 0 a 1.

A figura 3 é uma demonstração da figura da Lenna colorida e seus canais RGB.



Figura 3. Lenna RGB (Fonte: Autor)

D. Conversão de Cores

A conversão de imagens RGB para escala de cinza é utilizado em diversos estudos de Processamento de Imagens sobre detecção de objetos, pois essa técnica nos permite trabalhar com uma imagem unidimensional, facilitando o processamento de informações e segmentação da imagem (detecção de padrões) [13]. A transformação de uma imagem em RGB pode ser feita pela fórmula de Craig:

$$GRAY = 0.299 * R + 0.587G + 0.114 * B \quad (3)$$

onde R , G e B são, respectivamente, as camadas Red, Green e Blue da imagem colorida.

E. Filtros e Realces

O principal objetivo das técnicas de realce de imagens é processar uma imagem para que ela seja mais adequada que a original para alguma aplicação específica.

Uma das técnicas mais conhecidas de realce de imagens é a Filtragem Gaussiana sendo utilizado para borrar ou desfocar a imagem na qual ele é aplicado com o objetivo de reduzir os ruídos visando assim suavizar a cena com uma máscara construída de forma condizente com a distribuição normal bidimensional. Esse método é muito conhecido por ser utilizado na etapa de pré-processamento da imagem com o objetivo de enaltecer a estrutura da imagem em diferentes escalas. Nixon e Aguado [20] definem que uma das formas de se implementar um filtro gaussiano é definindo a máscara através da equação:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

Outra técnica conhecida mas por realçar as bordas de uma imagem, é o Filtro Bilateral que foi proposto originalmente por Tomasi e Manduchi em 1998 [21], como uma ferramenta para remoção de ruídos, ela possui como característica, a capacidade de remover ruídos e preservar bordas. O filtro bilateral [22] é definido como:

$$I(X) = \frac{1}{C} \sum_{y \in N(x)} e^{-\frac{-||y-x||}{2\sigma_d^2}} e^{-\frac{-||I(y)-I(x)||^2}{2\sigma_r^2}} I(y) \quad (5)$$

onde σ_d e σ_r são parâmetros que controlam o peso no domínio espacial e de intensidade, respectivamente, $I(X)$ representa os pixels da imagem, $N(x)$ uma vizinhança espacial de $I(x)$ e C é a constante de normalização representada pela seguinte expressão:

$$C = \sum_{y \in N(x)} e^{-\frac{-||y-x||}{2\sigma_d^2}} e^{-\frac{-||I(y)-I(x)||^2}{2\sigma_r^2}} \quad (6)$$

F. Detecção de Bordas

A detecção de bordas, do inglês *edge detection* é um dos principais focos no estudo de Processamento de Imagens, sendo considerado um dos temas mais difíceis da área.

Uma borda (*edge*) é definida como uma fronteira entre duas regiões cujos seus níveis de cinza predominantes são razoavelmente diferentes [14]. Segundo Pratt [16], define uma borda

de luminosidade como uma descontinuidade na luminosidade de uma imagem.

O algoritmo de Canny [17] é um método que surgiu através dos trabalhos de Marr e Hildreth. Esse algoritmo é um operador gaussiano de primeira derivada que suaviza os ruídos e localiza as bordas da imagem. Suas principais características são a dualidade entre a detecção e a precisão de localização das bordas. Sendo assim, utilizando o estudo do José Roberto Nogueira [15] podemos resumir as seguintes etapas:

- 1) Ler a imagem X ;
- 2) Criar uma máscara Gaussiana unidimensional G para convolver com I . O fator de escala desta Gaussiana é um parâmetro do detector de bordas;
- 3) Criar uma máscara unidimensional para a derivada primeira da Gaussiana, nas direções x (linha) e y (coluna), denominando-as de G_x e G_y ;
- 4) Convolver a imagem I com G ao longo das linhas, gerando a imagem I_x e, analogamente, ao longo das colunas para gerar I_y ;
- 5) Convolver I_x com G_x para gerar I_x' , que é o resultado da convolução da imagem I com a primeira derivada da Gaussiana ao longo das linhas; analogamente convoluir I_y com G_y para gerar I_y' ;
- 6) A magnitude do gradiente é calculada em cada pixel (x,y) na forma que segue:

$$M(x, y) = \sqrt[2]{I_x'(x, y)^2 + I_y'(x, y)^2} \quad (7)$$

- 7) Supressão não Máxima, tratando-se da busca de máximos locais de $M(x,y)$ nas direções transversais às bordas, reduzindo num mapa de bordas afinadas - isto é, bordas com espessura de 1 pixel;
- 8) Limiarização adaptativa (histerese), que consiste em uma limiarização baseada em dois limiares τ_1 e τ_2 , onde $\tau_1 = 2\tau_2$ ou $\tau_1 = 3\tau_2$. Aplicando a limiarização duas vezes, uma para τ_1 e outra para τ_2 , o algoritmo efetua um processo de complementação das descontinuidades da primeira limiarização aproveitando o resultado da segunda.

G. Segmentação por Contorno

O processo de segmentação de uma imagem tem como objetivo particona-la em regiões com propriedades comuns, por isso essas técnicas estão associadas ao conceito de região que pode ser definida como um conjunto de pixels conectados envolvidos por uma borda de pixels criando um contorno, outra definição para região é quando ela é tratada como um conjunto de pixels conectados por meio de uma condição de uniformidade [23].

O método mais comum de segmentação são baseadas nos contornos das imagens, nessa técnica, um objeto pode ser entendido como um região dentro de um contorno. E para identificar os objetos na imagem, inicialmente detectam-se suas bordas e tenta-se construir uma região conectada no interior a partir delas. Assim, para detectar as bordas temos que verificar a variação da luminosidade de uma imagem

visando localizar os pixels da borda, porém requer que haja uma ampliação no contraste entre o objeto e o fundo. A técnica mais conhecida para isso é a partir da suavização da imagem antes da captura da diferença, para que assim possa minimizar efeitos não desejados, porém a aplicação dessa medida pode ocorrer a perda de localização de algumas bordas [24], o que torna a tarefa de encontrar os contornos mais difícil.

H. Reconhecimento Ótico de Caracteres

O Reconhecimento Ótico de caracteres(OCR) é um acrônimo para o inglês *Optical Character Recognition*, é uma tecnologia para reconhecer caracteres a partir de um arquivo de imagem ou mapa de bits sejam, eles escaneados, escritos a mão, datilografados ou impressos. Dessa forma, através do OCR é possível obter um arquivo de texto editável por um computador(cite). Ele possui mais de 100 linguagens em sua biblioteca. Além de ter a possibilidade de adição de outras linguagens utilizando como suporte o *machine learning*, fazendo-o reconhecer novos conjuntos de padrões de um alfabeto.

A *Pytesseract* [18] é uma biblioteca que realiza o reconhecimento óptico de caracteres, fazendo assim a leitura do texto das imagens. Utilizando a linguagem Python ela é escolhida devido sua facilidade e alta abstração. Ela é derivada do módulo de código aberto *Tesseract* da Google o qual é uma OCR Engine que foi desenvolvida originalmente em C/C++ [19].

III. METODOLOGIA EXPERIMENTAL

A. Objetivos

O objetivo geral desse trabalho é modelar um sistema capaz de detectar e reconhecer placas de veículos de trânsito simulando um *Automatic Number/License Plate Recognition*, que significa, **Reconhecimento Automático de Placas de Veículos**, uma técnica utilizada para controle de tráfego, estacionamento, monitoração de faixas e entre outras aplicações [3]. Os objetivos específicos são definidos a seguir:

- Coletar imagens ou vídeos de veículos;
- Detectar e localizar a placa das imagens coletadas;
- Extrair as letras e os números da placa;
- Aplicar um Algoritmo de Reconhecimento de Caracteres;
- Retornar o resultado em formato de texto;

B. Ferramentas Computacionais

O estudo foi todo desenvolvido em **Python 3.7**, uma linguagem de alto nível, muito aplicada em estudos científicos por ser dinâmica, interpretada, robusta, multiplataforma, orientada a objetos e funcional [4].

O projeto foi programado totalmente no aplicativo Web gratuito chamado **Google Colab** que fornece um ambiente semelhante ao **Jupyter Notebook** sendo utilizado para o desenvolvimento interativo, simulando um software de código aberto que permite criar e compartilhar documentos que contêm códigos ativos, equações, visualizações e textos narrativos. Sendo nele possível algumas aplicações como: limpeza e transformação de dados, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina e muito mais [5].

C. Bibliotecas

- **OpenCV**: Fornece uma biblioteca, ferramentas e hardware otimizados para uma melhor visão computacional em tempo real [6].
- **NumPy**: Disponibiliza diversas funções matemáticas capazes de trabalhar com arranjos, vetores e matrizes multidimensionais [7].
- **Imutils**: Fornece uma série de funções capazes de tornar as funções básicas de processamento de imagem utilizando o OpenCV [8]
- **Pytesseract**: O seu objetivo é a leitura de textos e caracteres de uma imagem. Ela é capaz de transformar a imagem de um texto em arquivos editáveis de texto [9].
- **Matplotlib**: Uma biblioteca de plotagem que produz figuras 2D, em uma variedade de formatos impressos, como gráficos de barras, gráficos de setores, gráficos de dispersão e histogramas [10].

D. Dataset

O dataset do trabalho é um registro com 52 fotos de automóveis sendo 22 delas retirados da internet e 30 capturados pelo grupo. Todas as imagens dos veículos possuem placas Mercosul, devido as novas regras de trânsito que entraram em vigor em 2019 [12].

As imagens não serão disponibilizadas pelo grupo por questões de segurança pública.

E. Tratamento da Imagem

Assim como foi mencionado anteriormente, tarefas que tem como objetivo detectar objetos nas imagens é um dos tópicos mais difíceis na área de Processamento de Imagem, por isso para facilitar a detecção da placa resolvemos fazer um tratamento na imagem antes de localizarmos a placa.



Figura 4. Placa Original RGB [25]

Inicialmente, selecionamos uma imagem, Figura 4, convertemos a imagem para a escala cinza, utilizando a função da *OpenCV* chamada **COLOR_BGR2GRAY**, gerando a Figura 5, permitindo-nos trabalhar com uma imagem unidimensional, facilitando os processos seguintes.

Seguimos então aplicando um filtro de Desfoque Bilateral desfocando seletivamente uma imagem de forma que as bordas e outros detalhes sejam preservados, resultando na Figura 6



Figura 5. Placa Gray (Fonte: Autor)



Figura 6. Placa com Desfoco Bilateral (Fonte: Autor)

E finalizamos o tratamento da imagem aplicando o algoritmo de detecção de bordas *Canny Edge Detector*, resultando na imagem que destaca os contornos da Figura 7, facilitando o algoritmo para a detecção da placa através de seu contorno.

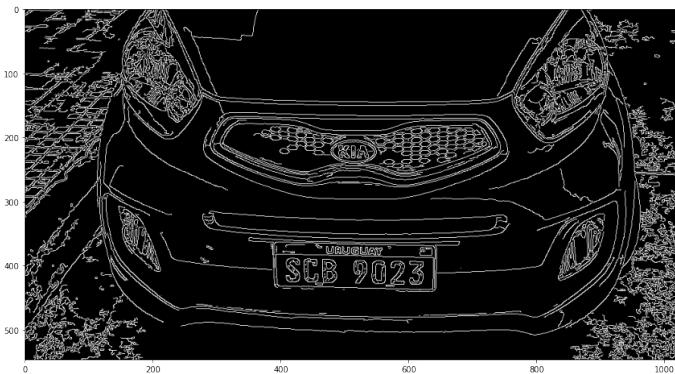


Figura 7. Placa com o *Canny Edge Detector* (Fonte: Autor)

F. Localizando a Placa do Veículo

Com o tratamento da imagem feito, passamos para a segunda parte do algoritmo onde teremos que localizar a placa a partir dos contornos da imagem.

Inicialmente queremos detectar todos os contornos da placa, então aplicamos o comando **findContours** da biblioteca

OpenCV, a tratamos como uma lista de vetores e a ordenamos. Com todos os contornos localizados e ordenados, passamos um laço de repetição calculando o perimetro dos mesmos, utilizando a função **arcLength** e aplicamos o **approxPolyDP** que aproxima uma curva ou um polígono com outra curva/polígono com menos vértices para que a distância entre eles seja menor ou igual à precisão especificada, caso essa aproximação retorne um polígono convexo, em seguida verificamos se o mesmo possui quatro vértices e o laço se quebra retornando a localização da placa na imagem.

Para uma melhor visualização geramos uma máscara preta com as mesmas dimensões da imagem original, porém com a parte da placa preenchida de branco, conforme a Figura 8.

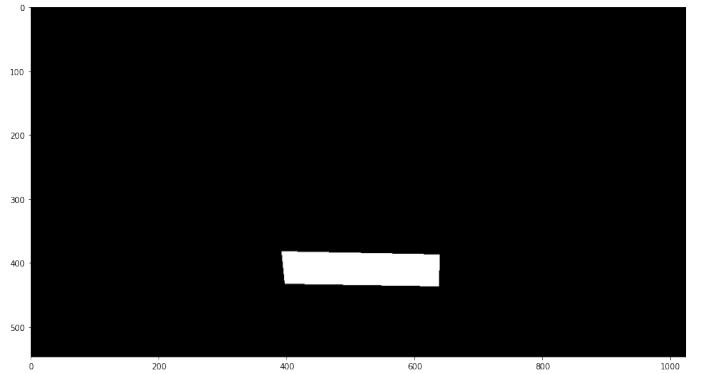


Figura 8. Mascara da Imagem com a Localização da Placa (Fonte: Autor)

Em seguida fazemos uma mesclagem utilizando a função **bitwise_and** da imagem anterior com a imagem original, resultando na Figura 9.

E para observarmos melhor a imagem da placa e facilitar no próximo passo, manipulamos a imagem para imprimir somente a placa, conforme a Figura 10.

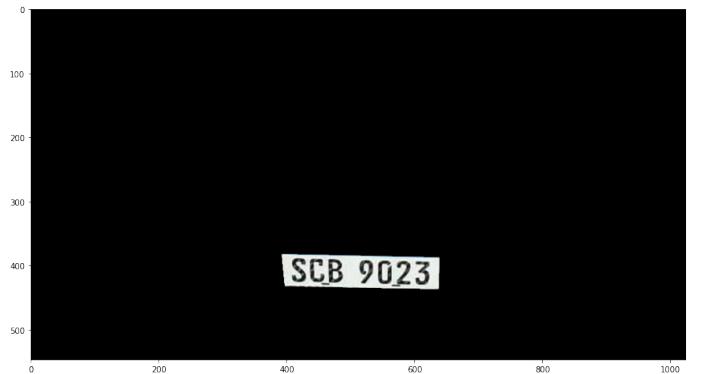


Figura 9. Mescla da Mascara com a Imagem Original (Fonte: Autor)

G. Reconhecimento dos Caracteres

Com a localização a placa feita e sua imagem gerada, nosso próximo objetivo foi conseguir interpretar os caracteres da imagem. Após a pesquisa e análise de diversas ferramentas computacionais resolvemos utilizar o *Pytesseract*, uma biblioteca que realiza a leitura do texto das imagens.



Figura 10. Imagem da Placa Captada (Fonte: Autor)

Configuramos ela para o padrão português, e que retornasse somente as letras do alfabeto brasileiro e os números, imprimindo assim o texto que foi localizado. Porém, para uma melhor visualização resolvemos gerar novamente uma imagem porém com a localização da placa e a sua descrição acima dela, como na Figura 11.



Figura 11. Imagem Final com a Placa Captada (Fonte: Autor)

IV. RESULTADOS

Nesta seção descrevemos sobre os resultados obtidos a partir do algoritmo desenvolvido para este estudo, na qual citamos a precisão do algoritmo em cima da nossa base de dados, possíveis motivos dos erros de captura da placa e erros na leitura de caracteres da imagem. O algoritmo desenvolvido pelo grupo apresentou bons resultados tanto na captura da placa quanto na leitura de caracteres da mesma.

Segundo a Tabela I, o algoritmo desenvolvido conseguiu detectar mais de 90% das placas nas imagens.

Tabela I
TABELA DE RESULTADO DAS PLACAS CAPTURADAS

	Imagens	Placas Detectadas	Porcentagem
Internet	22	21	95,45%
Capturadas	30	28	93,33%
Total	52	49	94,23%

Analisando as imagens que não foram possíveis detectar a placa percebemos os três motivos do erro da localização da placa:

- A placa estava em um ângulo ruim para ser detectada;
- O algoritmo detectou o farol retangular do veículo;
- A iluminação atrapalhou o contorno da placa na imagem.

A corretude dos caracteres das placas também demonstrou ótimos resultados, apresentando uma precisão de acima de 80%, conforme a tabela II.

Tabela II
TABELA SOBRE A CORRETUDE DOS CARACTERES DAS PLACAS

	Placas Detectadas	Aceritos	Porcentagem
Internet	21	19	90,48%
Capturadas	28	23	82,14%
Total	49	42	85,71%

Analizando os sete erros detectados, anotamos as seguintes observações:

- O algoritmo conseguiu detectar a placa porém devido a angulação da imagem capturada [26] isso acabou confundindo a leitura de caracteres da biblioteca, como demonstrado na Figura 12;
- A biblioteca confundia algumas vezes o '0' com a letra 'G', como a Figura 13;
- Outra confusão de carácter que ocorreu foi a letra 'I' com '1', como observado na Figura 14, ou o caso contrário 15;
- E um caso inesperado foi a interpretação do algoritmo do número '5' com a letra 'S', também demonstrado Figura 15.



Figura 12. Erro de Leitura de Caracter devido o Ângulo (Fonte: Adaptada [26] pelo Autor)

Vale ressaltar que as Imagens 4, 12 e 13 foram retiradas da internet [25] [26] [27]. E as Figuras 14 e 15 foram capturadas pelos autores, e por motivo de segurança pública censuramos os caracteres restante dos caracteres que não foram conectados.

Acreditamos que esses erros de caracteres ocorreram devido certas semelhanças gráficas entre os símbolos, como '1' e 'I', que mesmo tendo certas diferenças aplicadas pelo Mercosul ainda é possível causar uma confusão até para a visão de uma pessoa.

Baseando-se na quantidade total de imagens percebemos que a técnica desenvolvida conseguiu detectar e interpretar corretamente mais 75% das imagens, como demonstramos na Tabela III.



Figura 13. Erro de Leitura de Caracter de '0' com a letra 'G' (Fonte: Adaptada [27] pelo Autor)

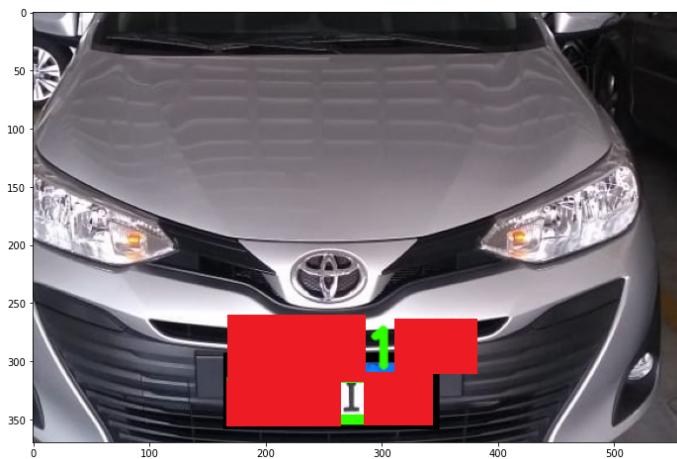


Figura 14. Erro de Leitura de Caracter de '1' com a letra 'I' (Fonte: Autor)



Figura 15. Erro de Leitura de Caracter de '1' com a letra 'I' e '5' e 'S' (Fonte: Autor)

Tabela III
TABELA SOBRE A PRECISÃO DO ALGORITMO

	Imagens	Acertos	Porcentagem
Internet	22	19	86,36%
Capturadas	30	23	76,67%
Total	52	42	80,77%

V. CONSIDERAÇÕES FINAIS

A quantidade de veículos no Brasil é notoriamente muito grande, dando destaque principalmente para a capital paulista, que registrou em 2018, 7,4 veículos motorizados para cada 10 habitantes segundo pesquisa da Companhia de Engenharia de Trânsito (CET) demonstrando possuir aproximadamente 8,6 milhões de carros, motos, ônibus ou caminhões [28].

O Mapa da Motorização Individual no Brasil de 2019 feito pela Observatorio das Metrópoles, um Instituto Nacional de Ciência e Tecnologia (INCT), demonstrou que essa tendência já havia sido apontada em outros levantamentos mencionando que desde os anos 2000 o país apresentou um crescimento contínuo da frota de veículos motorizados de todos os tipos [29].

Esse aumento de veículos ocorre pela preferência dos brasileiros em veículos individuais. A dissertação de Elenice da Silva [30] menciona que apesar do transporte rodoviário estar se consolidando no Brasil desde 1950 devido à necessidade de interiorização e integração do território nacional, o sistema de transporte coletivo brasileiro tem baixa confiabilidade e é considerada pouca produtividade, induzindo o brasileiro a ter seu próprio carro, aumentando o processo de degradação da circulação e da mobilidade urbana

Outro fator a ser considerado para esse estudo é a elevada quantidade de multas registradas em 2019, demonstrou que foram aplicadas só em setembro mais de 900.000 multas, um aumento de cerca de 5% comparada ao ano anterior, sendo maior parte deles multas por excesso de velocidade, o que levou ao governo a aumentar a taxa de multa de trânsito a fim de evitar números mais exorbitantes [31].

O estudo realizado tinha como objetivo analisar quais seriam as melhores técnicas de Processamento de Imagem para podermos desenvolver um algoritmo capaz de detectar e interpretar as placas de veículos com as novas placas Mercosul que foram implementadas devido às novas normas de trânsito [12]. Com as nossas pesquisas aprendemos que o melhor jeito de fazer o computador detectar uma placa de um veículo seria através da análise dos contornos da imagem, porém para que isso ocorra de maneira correta e eficiente foi necessário fazer um tratamento na imagem para facilitar sua localização. E graças a biblioteca *Pytesseract* conseguimos fazer o algoritmo interpretar esses dados corretamente.

O método implementado demonstrou uma precisão de 80,77% com detecção e interpretação correta da leitura da placa. Analisando as imagens que apresentaram erros percebemos que muitas vezes isso ocorreu devido um mal ângulo da imagem capturada ou uma confusão de caracteres semelhantes como '1' e 'I'.

Graças a esse trabalho entendemos o porquê de existir um trabalho especializado sobre a melhor localização de um radar de trânsito, pois alguns parâmetros como iluminação, ângulo e contraste influenciam diretamente na hora da captura da imagem de um veículo que ultrapassou a velocidade limite de uma rodovia.

Se continuarmos esse estudo pretendemos aplicar nosso algoritmo em vídeos, além de, se possível, utilizarmos bi-

bliotecas de aprendizado de máquinas ou redes neurais para treinarmos nosso programa a fim de ensiná-lo as diferenças entre os caracteres, mesmo que sejam muito semelhantes.

REFERÊNCIAS

- [1] Oliveira, A. Brasil tem 28 milhões de carros e 13 milhões de motos a mais. *Fundação Perseu Abramo*. (2019,10), <https://fpabramo.org.br/2019/10/31/brasil-tem-28-milhoes-de-carros-e-13-milhoes-de-motos-a-mais/>
- [2] Machado, P. Número de multas em rodovias federais cresce 88% em dez anos: o que leva a este elevado índice?. *Portal Do Trânsito*. (2020,9), <https://www.portaldotransito.com.br/noticias/numero-de-multas-em-rodovias-federais-cresce-88-o-que-leva-a-este-elevado-indice/>
- [3] ALVARENGA, E. [en] OPTICAL CHARACTER RECOGNITION FOR AUTOMATED LICENSE PLATE RECOGNITION SYSTEMS. (Pontifical Catholic University of Rio de Janeiro,2017)
- [4] Team, P. Python: A dynamic, open source programming language. *Python Software Foundation*. **78** (2015)
- [5] Bisong, E. Google colaboratory. *Building Machine Learning And Deep Learning Models On Google Cloud Platform*. pp. 59-64 (2019)
- [6] Opencv OpenCV. (2021,6), <https://opencv.org/>
- [7] Numpy Numpy. (2021,6), <https://numpy.org/>
- [8] Imutils Imutils. (2021,6), <https://pypi.org/project/imutils/>
- [9] Rosebrock, A. Using Tesseract OCR with Python. (Retrieved from pyimagesearch: <https://www.pyimagesearch.com/2017/07/10 ...>,2017)
- [10] Matplotlib Matplotlib. (2021,6), <https://matplotlib.org/>
- [11] Car License Plates Dataset. (2021,6), <https://makeml.app/datasets/cars-license-plates> (Make ML, 2021)
- [12] AutoPapo Novas Regras Para placas mercosul entram em vigor. *Auto-Papo*. (2020, 03), <https://autopapo.uol.com.br/noticia/novas-regras-para-placas-mercosul-comecam-a-valer/>
- [13] Cordeiro, T. RECONHECIMENTO AUTOMÁTICO DE PLACAS UTILIZANDO INTELIGÉNCIA COMPUTACIONAL. *UNIVERSIDADE CANDIDO MENDES, CAMPOS DOS GOYTACAZES, RJ*. (2015)
- [14] Marques Filho, O. & Neto, H. Processamento digital de imagens. (Brasport,1999)
- [15] Santos Lensone, E., Dal Poz, A. & Nogueira, J. Detector de Bordas de Canny: Um Estudo da Dualidade entre a Precisão de Localização e a Sensibilidade aos Ruidos da Imagem, 2004.
- [16] Pratt, W. K., Digital Image Processing, Wiley Interscience, 1991. (2nd ed.).
- [17] Canny, J. A computational approach to edge detection. *IEEE Transactions On Pattern Analysis And Machine Intelligence*., 679-698 (1986)
- [18] Python-Software-Fundation. Pytesseract - pypl, jun 2019. URL <https://pypi.org/ project/pytesseract/>.
- [19] Google-Open-source. Tesseract ocr, jun 2019. URL <https://opensource.google. com/projects/tesseract>.
- [20] Nixon, M. & Aguado, A. Feature extraction and image processing for computer vision. (Academic press,2019)
- [21] Tomasi, C. & Manduchi, R. Bilateral filtering for gray and color images. *Sixth International Conference On Computer Vision (IEEE Cat. No. 98CH36271)*. pp. 839-846 (1998)
- [22] Zhang, M. Bilateral filter in image processing. (2009)
- [23] Gonzalez, R. & Woods, R. Processamento de imagens digitais. (Editora Blucher,2000)
- [24] MAIA, G. & PORFÍRIO, V. O processo de detecção de bordas de canny: fundamentos, algoritmos e avaliação experimental. (2002)
- [25] Vince, Live. Placa Mercosul: saiba tudo sobre as novas placas de carro no Brasil (2020,12), [https://live.apto.vc/placa-mercosul-saiba-todo-sobre-as-novas-placas-de-carro-no-brasil/](https://live.apto.vc/placa-mercosul-saiba-tudo-sobre-as-novas-placas-de-carro-no-brasil/)
- [26] Contagem Motorpeças. Saiba tudo sobre as Placas Mercosul. (2020, 11), <https://contagemmotorpecas.com.br/blog/curiosidades-sobre-placa-mercosul/>
- [27] Quadra, D. Placa do Mercosul começa a ser implantada no Brasil em setembro. (2018, 03), <https://www.radioculturafoz.com.br/2018/03/08/placa-do-mercosul-comeca-a-ser-implantada-no-brasil-em-setembro/>
- [28] Menezes, C. Cidade de SP tem 7,4 veículos para cada 10 habitantes, aponta levantamento da CET. (2018, 05), <https://g1.globo.com/sp/sao-paulo/noticia/cidade-de-sp-tem-74-veiculos-para-cada-10-habitantes-aponta-levantamento-da-cet.ghtml>
- [29] Rodrigues, J. Mapa da Motorização Individual 2019 . (2019, 09), <https://www.observatoriodasmetropoles.net.br/mapa-da-motorizacao-individual-2019/>
- [30] Silva, E. Análise do crescimento da motorização no Brasil e seus impactos na mobilidade urbana. (Universidade Federal do Rio de Janeiro,2011)
- [31] Cardoso, W. Numero de Multas de Transito em SP Aumenta . (2020, 01), <https://agora.folha.uol.com.br/sao-paulo/2020/01/numero-de-multas-de-transito-em-sp-aumenta.shtml>