

Sistema de Detecção e Reconhecimento de Placas Veiculares

Daniel Arruda Groh 31061
Engenharia de Computação
Universidade Federal de Itajubá
Campus Itabira
daniagroh@gmail.com

Danilo Ricardo Crabi Freitas - 31195
Engenharia de Computação
Universidade Federal de Itajubá
Campus Itabira
danilocrabi@gmail.com

Douglas Venâncio - 28164
Engenharia de Computação
Universidade Federal de Itajubá
Campus Itabira
dougsov@gmail.com

Joao Pedro Costi- 31397
Engenharia de Computação
Universidade Federal de Itajubá
Campus Itabira
joao_costi@hotmail.com

Abstract—In the follow study, is made an algorithm that is capable of identify characters in a image, more specifically in a image that contains the plate of a vehicle. Using techniques of image processing, filters are applied to images to facilitate the segmentation of the image, in order to obtain a image that contain only the plate. With this result an artificial intelligence is trained, and in future searches the algorithm will be able to deduce each character contained in a unknown image. The results obtained in this study are satisfying, in order that was possible to deduce some characters on plate images.

Resumo—No seguinte estudo é elaborado um algoritmo que tem como objetivo a identificação de caracteres presentes e uma imagem, mais especificadamente em uma imagem de placa de automóveis. Com o auxílio de técnicas de computação gráfica são aplicados filtros para facilitar a segmentação da imagem, afim de obter como resultado uma imagem contendo somente a placa presente na imagem. Com este resultado, uma inteligência artificial é treinada, afim de que e futuras consultas, o algoritmo seja capaz de inferir quais são os caracteres presentes e uma imagem desconhecida. O resultado obtido com o algoritmo desenvolvido foi satisfatório, tendo e vista que foi possível realizar a inferência de caracteres presentes em imagens de placas de veículos.

Keywords—Computação gráfica; OpenGL; Identificação; Placas

I. INTRODUÇÃO

Com o passar dos anos o número de carros aumenta em todo o mundo, segundo SCHMITZ e HÜBNER (2002) esse fato trouxe graves problemas, como por exemplo: Congestionamentos constantes, falta de coordenação dos semáforos entre cruzamentos, desvios de tráfego, segurança do motorista e acidentes. Diante esses diversos problemas, nasce a necessidade de se criar soluções, uma delas é o uso de computação gráfica para reconhecimento de placas automotivas.

Um sistema que identifica placas de carro possui diversas aplicações, ele pode ser aplicado em radares que garantem uma certa para os motoristas, na entrada de um condomínio para identificar os carros que podem entrar.

Contudo, a aplicação de processamento de imagens na identificação dos caracteres de uma placa não é uma tarefa trivial, pois as imagens podem ser corrompidas por degradações como a distorção linear de frequências, ruídos, artefatos de blocagem e condições naturais de luminosidade, neblina e regiões com sombra ((DAMERA-VENKATA et al., 2000)

A. Estado da Arte

Diversos estudos e implementações são realizadas para identificar placas, esses sistemas podem ser feitos de diferentes maneiras, nesta seção será apresentado alguns trabalhos da literatura sobre esse tema.

No trabalho de NASCIMENTO (2012) uma câmera captura a imagem do carro, em seguida realiza um pré-processamento, onde empregaram-se técnicas de diminuição de ruído de Gauss, conversão de cores do formato RGB (com três canais, vermelho-verde-azul) para escala de cinza (único canal), experimentando também a técnica de conversão em HSL (matriz-saturação-luz) para manipulação das cores. Em seguida para a detectar a placa é aplicado a técnica de detecção e segmentação de bordas de Canny. Após encontrar a placa é feito o recorte de cada carácter. Para identificar o carácter é utilizado a biblioteca Tesseract OCR da linguagem C++, esta biblioteca pode ler uma grande variedade de formatos de imagem e converte-los em textos.

Em sua tese SHINZATO (2010) utiliza basicamente as mesmas técnicas de NASCIMENTO (2012) para o pré-processamento de imagem capturada por uma câmera, o que muda é o classificador que foi utilizado. Ele utiliza

uma rede neural para encontrar padrões dos caracteres e assim pode identifica-los.

B. Relato Sobre o Trabalho Feito

Neste estudo é desenvolvido um algoritmo capaz de realizar a inferência de caracteres presentes em imagem contendo a placa de automóveis. Para isso são realizados trabalhos no processamento de imagens, afim de desenvolver uma técnica capaz de isolar uma placa presente em uma imagem. Posteriormente com as placas isoladas, é realizado o treinamento de uma inteligência artificial, onde é utilizado um banco de imagens de diferentes placas de automóveis, e após o treinamento é possível inferir qual é o caracter presente em uma imagem. O algoritmo utilizado para a inteligência artificial é o Knearest, presente na biblioteca OpenCV.

C. Visão Geral da Técnica Utilizada

Para realização das atividade proposta seguiu-se o esquema representado na figura 1.

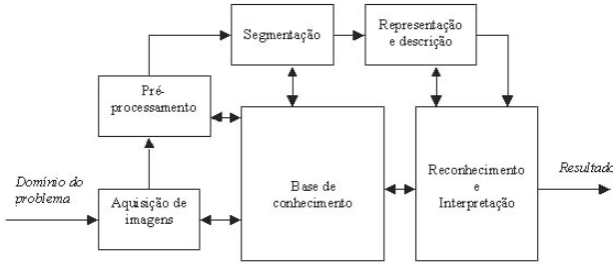


Figura 1. Fluxograma
Fonte: Wu, Shin - Ting

Neste estudo, primeiramente as imagens são adquiridas de uma base de conhecimento contendo placas de automóveis, estas imagens são pré-processadas, onde são aplicados filtros para melhorar a qualidade da imagem, e no processo de segmentação seja extraída a placa, de maneira satisfatória. Com as placas extraídas, é utilizada uma fração destas imagens para realizar o treinamento da inteligência artificial que posteriormente é capaz de realizar o reconhecimento e interpretação de novas imagens inseridas, sendo possível inferir quais são os caracteres presentes em uma nova imagem.

II. INFORMAÇÕES TÉCNICAS

Para a manipulação das imagens foi necessário utilizar conhecimentos sobre filtros, identificação de contornos e cropping utilizando a biblioteca OpenCV. Além disso, foram aplicados conceitos de inteligência artificial para possibilitar o treinamento de uma inteligência que posteriormente seja capaz de inferir quais são os caracteres de placas de automóveis em imagens.

A. Filtros

Foram utilizados diversos filtros para a otimização dos aspectos da imagem para realizar uma melhor análise dos contornos. Primeiramente antes do tratamento da imagem a mesma foi convertida do sistema de cores RGB para o sistema HSV que divide seus parâmetros em matiz, pureza e brilho conforme a figura 2.

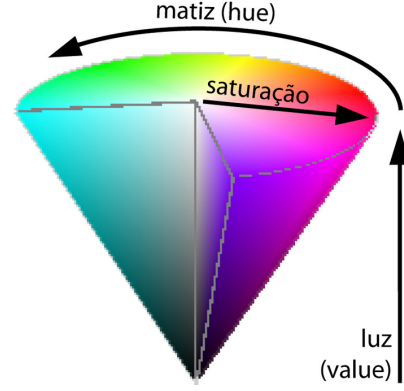


Figura 2. Sistema de cores HSV
Fonte: Ribeiro, Madalena Gomes, Abel

Escolheu-se utilizar apenas sua camada de brilho que quando sozinha apresenta apenas tons de cinza dessa forma obtendo a imagem em uma escala de cinza, Após a etapa anterior são aplicadas as operações de morfologia matemática Top-Hat e Black-Hat com o objetivo de realçar as bordas e os objetos escuros, e por fim utiliza-se um filtro gaussiano para gerar um efeito de embaçamento na imagem para reduzir seu ruído.

Como citado anteriormente, primeiramente é realizada uma transformação radiométrica no brilho da imagem para convertê-la do espaço RGB para o espaço HSV a sua conversão é dada pelas seguintes equações.

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{se } MAX = R \text{ e } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{se } MAX = R \text{ e } G < B \\ 60 \times \frac{G-B}{MAX-MIN} + 120, & \text{se } MAX = G \\ 60 \times \frac{G-B}{MAX-MIN} + 240, & \text{se } MAX = B \end{cases} \quad (1)$$

$$S = \frac{MAX - MIN}{MAX} \quad (2)$$

$$V = MAX \quad (3)$$

Onde MIN são os valores mínimos de RGB e MAX são os valores máximos de RGB.

Para tratar a imagem foram usadas operações morfológicas como a dilatação que consiste aplicação de um kernel pixel a pixel na imagem, onde seu centro

está posicionado no pixel onde se deseja aplicar a dilatação e o valor do pixel onde ele é aplicado resulta no valor máximo entre todos os pixels da imagem cobertos pelo kernel, essa operação gera uma expansão dos pixels mais claros na imagem. A erosão é calculada da mesma forma que a dilatação mas ao invés de aplicar o valor máximo como resultado, o resultado é o valor mínimo entre todos os pixels cobertos pelo kernel, o que faz com que os objetos com pixels mais claros na imagem emagreçam. A partir das transformações morfológicas citadas foi utilizado um conjunto de operações morfológicas com o objetivo de realçar as bordas chamado White Top-Hat dado pela seguinte equação.

$$WTH = (I, (E, V)) = I - (I \circ (E, V)) \quad (4)$$

Onde \circ é a operação de abertura que consiste em uma erosão seguida de uma dilatação.

Para realçar os objetos escuros foi usado outro conjunto de operações morfológicas denominado Black Top-Hat determinado por:

$$BTH = (I, (E, V)) = (I \bullet (E, V)) - I \quad (5)$$

Onde \bullet é a operação de fechamento que consiste em uma dilatação seguido de uma erosão.

A aplicação de um filtro em uma imagem é caracterizada pela convolução da imagem original com uma matriz(kernel) específico para cada filtro. Para o filtro de suavização gaussiano, o seu kernel é conforme a equação abaixo.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6)$$

Logo o seu kernel no plano x,y tem o feito de suavização devido ao seu sua curva no 2D como mostra a imagem a abaixo.

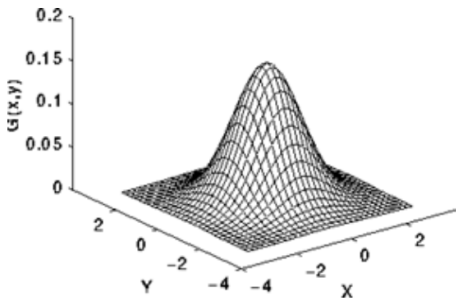


Figura 3. Filtro Gaussiano 2D
Fonte: Wu, Shin - Ting

A transformação linear consiste em manipular um intervalo alvo em um histograma, essa manipulação pode

ser tanto em sua localização no histograma ou quanto a largura deste intervalo, sua equação é dada por:

$$K = \begin{cases} k1 & , \text{ se } I < I1, \\ \frac{(k2-k1)}{(I2-I1)}(I - I1) + k1 & , \text{ se } I1 \leq I < I2 \\ k2 & , \text{ se } I \geq I2 \end{cases} \quad (7)$$

Onde o K1 e K2 são os limites de superior e inferior respectivamente do novo intervalo e I1 e I2 determinam os intervalos da transformação. Quando seu parâmetros I1 e I2 são iguais temos a transformação linear tem a função de limiarização(Threshold) ou seja, binarização da imagem onde os pixels podem só assumir 2 valores, logo esta propriedade da linearização foi utilizada para separar o primeiro objeto de interesse deste projeto as placas para separar a placa da cena que ela se situa na imagem ou em qualquer situação onde se deseja separar na imagem um objeto da cena determinar qual limiar quando I2=I1 que irá separar a placa da cena.

Para determinar esse limiar o método de otsu se mostra eficiente. O método de otsu funciona da seguinte forma ao ter duas classes em um conjunto de valores onde essas classes são em termos de escala de cinza C1=[0,1,2,3,4...,t] e C2=[t+1,t+2,t+3...,L] onde em uma imagem o segundo intervalo são os valores de pixels de um objeto e primeiro são os valores de pixel do fundo. O método de otsu determina conforme as 2 equações abaixo:

$$\sigma_c^2 = \sigma_{c1}^2(k)P_1(k) + \sigma_{c2}^2(k)P_1(k) \quad (8)$$

$$k = \min_{0 \leq k \leq L-1} (\sigma_c^2) \quad (9)$$

Logo temos que a variância intraclasse para os tons de cinza da imagem na equação 8 e o melhor limiar k que é o limiar que divide objeto desejado do fundo é k, que contem a menor variância intraclasses segundo a equação 9.

Feito as aplicações dos filtros descritos, o resultado obtido é demonstrado na figura 1, este resultado é encaminhado para a ser realizado a segmentação, para enfim identificar a placa na imagem.

B. Segmentação

Esta etapa é dedicada a identificar objetos na imagem através da identificação de contornos (regiões fechadas) na imagem preto e branco. Os contornos identificados são analisados afim de identificar o contorno que representa a placa do automóvel, e quando é identificada, sua localização é obtida, e a imagem original é cortada naquelas coordenadas, afim de isolar a placa do resto da imagem capturada. A partir de todos os contornos obtidos é necessário realizar diversas regras para poder isolar os contornos que representam a placa do automóvel. Entre as regras pode-se destacar valores máximos e mínimos dos contornos de interesse, assim como as suas semelhanças e alinhamento entre eles.

C. Inteligência Artificial

Com a placa identificada, e isolada em uma imagem à parte, é possível realizar através de técnicas de inteligência artificial.

Para tal feito é utilizado o k-nearest Neighbors Algorithm (k-NN), um algoritmo de reconhecimento de padrões do tipo Machine Learning. Foi utilizado suas funções presentes na biblioteca do OpenCV. A função Knearest, realiza o mapeamento das características de uma imagem no espaço, como é demonstrado pela Figura 4. Para a utilização do algoritmo Knearest, as características estéticas e as informações correspondente a cada imagem, são convertidas em um vetor de float, para poderem ser manipuladas pelo algoritmo do Knearest. Para o funcionamento desta função primeiramente é necessário que ela passe por um processo de treinamento, onde é realizado a população do espaço, fazendo com que os objetos com características similares sejam agrupados mais próximos. Desse modo, quando é finalizado o treinamento tem-se um espaço populado com os objetos inseridos, de modo que os objetos similares se encontram aglutinados. Em nosso caso, para o treinamento utiliza-se imagens de placas, para treinar o algoritmo com os caracteres do alfabeto e números decimais.

Assim, quando se tenta realizar a inferência dos caracteres presentes em uma imagem, primeiramente isola-se cada carácter da placa, para esse carácter ser pesquisado utilizando a função Knearest, de modo que o carácter é inserido no espaço já existente (que foi treinado previamente), e é analisado os K objetos mais próximos do carácter, e a função retorna o objeto mais recorrente nas proximidades do carácter. Desse modo é feita a inferência de cada carácter da imagem pesquisada.

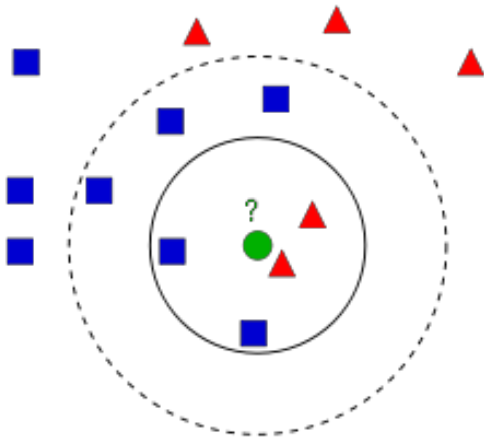


Figura 4. Knearest

Fonte: Documentação Knearest OpenCV

III. IMPLEMENTAÇÃO

Para a implementação do projeto pode-se considerar 3 etapas de grande importância para o sucesso da aplicação, sendo essas apresentadas no Diagrama Geral da Aplicação apresentado na Figura 5 abaixo:

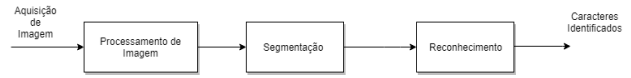


Figura 5. Diagrama Geral da Aplicação

Fonte: Autores do Estudo

A metodologia de implementação de cada uma dessas etapas serão apresentadas nesta seção, de acordo com a sequência do diagrama.

A. Processamento de Imagem

A primeira etapa do processamento de imagem consiste na aplicação de filtros afim de adequar a imagem de entrada de acordo com o pretendido, todos os filtros utilizados serão discutidos abaixo.

HSV e Cinza: Para a conversão da imagem para tons de cinza para melhor tratamento da imagem primeiramente a imagem foi transformada do RGB para o sistema de cores HSV utilizando utilizando a função `cvtColor()` que faz a conversão conforme as equações 1, 2 e 3 a imagem é armazenada em uma matriz cubica de 3 camadas onde sua terceira camada é o brilho é pega e separada das demais como ela separadamente defini os tons de cinza da imagem logo temos a imagem em tons de cinza.

Threshold:

Para a binarização da imagem foi aplicada a linearização conforme a equação 7 com seus parâmetros $K1=0$, $K2=255$ e $I=I1=I2$, logo dessa forma a imagem terá apenas 2 valores nos pixels 0 ou 255 como o objetivo é separar o objeto de interesse (placa) para isso foi necessário a definição do limiar I de modo a separar o objeto de interesse do fundo, assim usou-se o método de otsu para definir esse limiar. A implementação de threshold de duas formas a primeira foi o threshold local assim dividiu-se a imagem em regiões menores e calculou-se o seu limiar I apenas levando em conta os pixels da região e aplicando o threshold com esse limiar na região especificada e repetindo esse processo para as demais regiões para isso foi usado a função `adaptiveThreshold()` que executa todo este processo.

A segunda forma foi o threshold global após a placa separada em uma imagem o limiar I tendo em conta toda a imagem onde para executar essa operação foi usado a função `threshold()` passando como um dos parâmetros o limiar I calculado pelo método de otsu.

Filtro Gaussiano:

$$K = \frac{1}{16} * \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (10)$$

A função utilizada para aplicar o filtro gaussiano é: `GaussianBlur()`;

Onde a primeira variável é a imagem em escala de cinza que deve ser aplicada o filtro, a segunda variável é a imagem resultante da aplicação do filtro, a terceira variável

é o tipo de filtro que será aplicado, e a quarta variável é o ponto em que o kernel do filtro irá ser aplicado nos pixels da imagem.

B. Segmentação

Após a aplicação dos filtros foi possível obter uma imagem de Thesholding, onde é possível retirar os contornos contidos na imagem e posteriormente segmenta-la. Para isso foi utilizado a função `findContours` presente na biblioteca do OpenCv. Para a correta seleção dos caracteres dentre todos os contornos identificados pelo algoritmo é necessário aplicar algumas regras. Entende-se uma placa como uma sequencia de sete contornos com características similares(três letras e quatro números). Esse padrão pode ser conferido a partir da análise dos dados que se pretende obter, no caso desta aplicação as placas de automóveis em território brasileiro. Tendo em vista a similaridade esperada entre os contornos presentes numa placa, realiza-se comparações entre os contornos obtidos de modo a procurar sete contornos que tem posições similares em relação ao eixo Y do plano, e sabendo que a dimensão dos caracteres são similares, dentre os contornos que se encontram da mesma posição no eixo Y, é verificado se eles possuem valores próximos de área. Assim, são obtidos conjuntos de sete contornos que representam possíveis placas, e por ultimo é verificado se dentro desses conjuntos existem contornos que se sobrepõem, excluindo-os do conjuntos das possíveis placas. Assim ao fim da aplicação dessas regras é possível obter o conjuntos de contornos que representam a placa na imagem tratada.

Ao identificar a placa é realizado um corte na imagem nas posição que a placa esta presente, gerando como resultado uma imagem contendo apenas a placa.

Com a placa localizada é realizado novamente um tratamento com os filtros discutidos anteriormente, afim de obter uma melhor qualidade dos caracteres presentes na placa, sendo encontrados e segmentados, obtendo assim os dados necessário pra classificação e análise de padrões.

C. Knearest

As principais funções que são utilizadas neste estudo são as funções `train` e a `findnearest` do algoritmo, a primeira tem como característica o treinamento do algoritmo, e ela funciona da seguinte maneira:

```
train(InputArray samples,int layout,InputArray responses)
```

onde a variável `samples` contem as características dos objetos a serem inseridos no treinamento, a variável `layout` determina se cada objeto inserido irá ocupar uma linha ou coluna da matriz gerada, e a variável `response` contem as informações estéticas da variável que está sendo inserida. A segunda função utilizada é responsável por pesquisar e inferir qual carácter esta presente em uma imagem, e ela tem a seguinte características:

```
findNearest (InputArray samples, int k, OutputArray results)
```

Onde a variável `samples` contem as informações do carácter contido na imagem pesquisada, a variável `K` irá determinar a quantidade de vizinhos que serão analisados, e a variável `result` irá retornar o objeto mais recorrente dentre os `K` mais próximos do carácter pesquisado. Nesta implementação o valor de `K` utilizado foi 1, devido a pouca quantidade de dados disponível, o que poderia ter implicação nos resultados de comparação e identificação.

IV. RESULTADOS E DISCUSSÕES

Neste capítulo será apresentado os resultados obtidos. Este será dividido em 3 seções, de acordo com os processos realizados pela aplicação, sendo primeiro o processamento de imagem, a segmentação e Inteligencia artificial. A Figura 6 abaixo é um exemplo de dados de entrada do sistema, sendo está uma imagem sem nenhum tipo de processamento prévio.



Figura 6. Imagem Original
Fonte: Autores do Estudo

Os tratamentos na imagem foram realizados seguindo os passos discutidos na seção implementação. A primeira etapa é através da aplicação de filtros. A imagem é convertida para o espaço de cores HSV, e posteriormente para a escala de cinzas. Com a imagem em escala cinza é aplicado uma filtro de suavização Gaussiano, e posteriormente obtida a imagem de Threshold pretendida, está é apresentada na Figura 7 abaixo:



Figura 7. Imagem adaptativeTheshold
Fonte: Autores do Estudo

Com a imagem binária obtida através do threshold, foi possível obter os contornos presentes na imagem, a Figura 8 representa a imagem formada por todos os contornos da imagem.

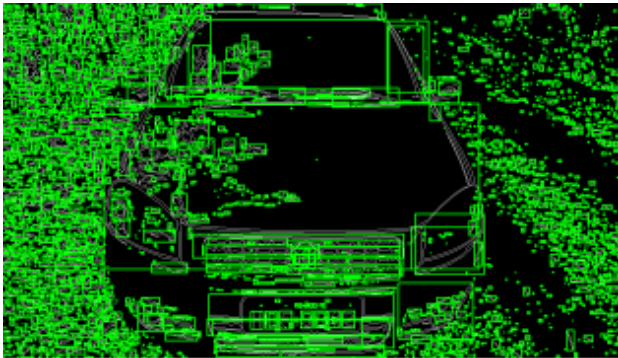


Figura 8. Imagem Contornos
Fonte: Autores do Estudo

Foram aplicados retângulos em todos os contornos, com os retângulos foi possível obter informações em relação ao contornos, tais como posição x, y, largura, altura e área ocupada. Uma vez que os objetos de interesse na imagem apresentam características parecidas, o resultado desta seleção é uma imagem com menor numero de contornos, representada pela Figura 11, onde está presente a região da placa do automóvel.

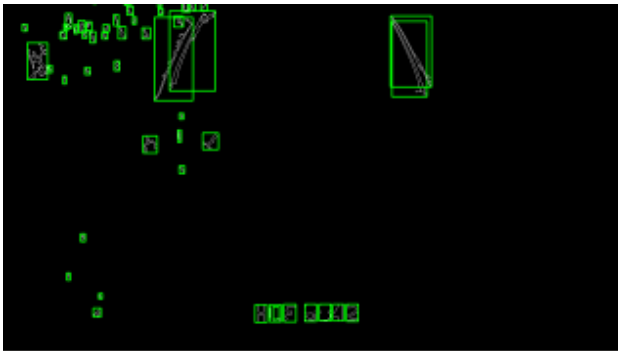


Figura 9. Imagem Contornos Interesse
Fonte: Autores do Estudo

Com poucos contornos presentes na imagem, a segmentação da região a qual a placa se apresenta é simplificada. Para isso foi realizado uma série de procedimentos nos contornos, primeiro a exclusão de objetos múltiplos e ordenação em relação ao eixo X. Com os objetos unificados e ordenados foi realizado um rastreamento percorrendo todos os contornos com o objetivo de encontrar 7 objetos de característica parecidas e alinhados no eixo Y, permitindo poucas variações em relação as suas características. Através desses procedimentos foram encontrados possíveis locais no qual a placa pode pertencer. A Figura 11 mostra a placa segmentada e tratada, pode-se observar a quali-

dade da imagem, com os caracteres bem nítidos, deve-se isso a um novo threshold aplicado, na placa segmentada



Figura 10. Imagem Placa Cortada Theshold
Fonte: Autores do Estudo

Com a imagem da placa isolada é possível segmentar a imagem em cada um dos seus caracteres, a Figura 11 apresenta o resultado deste processo, onde é exibido o numero 4 dá cadeia de caracteres segmentado e padronizado em um tamanho de XXXX, XXXX.



Figura 11. Imagem Segmentação Caracteres
Fonte: Autores do Estudo

Com os resultado referentes ao tratamento e segmentação da imagem apresentado, a próxima etapa se relaciona a Inteligencia Artificial.

O KNN se trata de um algoritmo de aprendizagem de maquina capaz de reconhecer padrões, ele tem seu aprendizado supervisionado, sendo assim necessita de um conjunto de dados de treinamento. Esses dados foram gerados através da apresentação da imagem original da placa em uma janela, com a iteração através dos contornos, onde pelo teclado foi passado um tipo char referente a cada segmentação da imagem, tendo ao final deste processo dois arquivos .XML. Ao final do processo esses dois arquivos são carregados novamente no programa para a realização de testes nas amostras necessárias.

Um dos problemas enfrentados e que possivelmente teve impactos nos resultados é a limitação de dados disponíveis para a realização do aprendizado, uma vez que a associação de objetos ocorre através do agrupamento de padrões.

O numero de dados disponível para a utilização consistia em 18 imagens de veículos devidamente emplacadas. O treinamento dos padrões foram realizados nas 18 placas.

Após o treinamento das placas, foi testado as 18 imagens afim de se obter a eficiência do algoritmo desenvolvido. A tabela 01 expõe a quantidade de caracteres que foram reconhecidos e classificados corretamente, os lugares marcados com "X" significam que a placa não foi reconhecida.

Imagem	Quant. Caracteres Detectados	Quant. Caracteres Classificados Corretamente
1	7	7
2	7	6
3	X	X
4	7	7
5	7	7
6	7	7
7	7	7
8	7	7
9	X	X
10	7	7
11	5	5
12	7	7
13	X	X
14	6	6
15	X	X
16	7	7
17	6	6
18	X	X

Tabela I
TABELA1: RESULTADOS

Analisando a tabela 1 é possível perceber que 5 placas não foram reconhecidas, isso se deve principalmente a baixa qualidade dessas imagens. Outro fator que atrapalhou foi a iluminação, quando as placas estavam com alta incidência dos raios do sol, o algoritmo não conseguiu reconhecer.

O número de caracteres encontrados dentre as placas reconhecidas foi igual a 101, sendo que apenas 4 não foram identificadas, obtendo-se uma taxa de reconhecimento de 96,1%, e dentre eles 91 caracteres foram reconhecidos corretamente, ou seja, a taxa de acerto dentre todas as imagens foi igual a 72,2%.

V. CONCLUSÃO

Ao fim dos estudos foi possível constatar que o tratamento feito nas imagens foi satisfatório, pois após aplicar os filtros, e o threshold, a imagem resultante consegue ser segmentada de maneira satisfatória, sendo possível isolar a placa da imagem. Com a placa isolada, foram aplicados conceitos de inteligência artificial para a utilização do algoritmo Knearst que é treinado com uma fração das imagens das placas, para que em seguida fosse possível realizar a inferência dos caracteres presentes na placa. Desse modo o resultado obtido foi satisfatório, sendo que foi possível identificar os caracteres presentes em imagens de placas de veículos.

REFERÊNCIAS BIBLIOGRÁFICAS

CONCI, A; AZEVEDO, E.; LETA, F.R.; **Computação Gráfica: Teoria e Prática. V2.** Campus - Rio de Janeiro, 2008

DAMERA-VENKATA, N. et al. **Image quality assessment based on a degradation model. v. 9n. 4,** p. 636 – 650, April 2000.

NASCIMENTO, Jean Dias do. **Detecção e reconhecimento de placa automotiva com baixo custo.** 2012.

GONZALEZ, R.; WOODS, R. **Processamento digital de imagens digitais. 3a. ed.** São Paulo: Pearson Prentice Hall, 2010.

RIBEIRO, MADALENA GOMES, ABEL. **Adaptação de Cor para Dicromatas na Visualização de Imagens.** (2012).

SHINZATO, Patrick Yuri. **Sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais.** 2010. Tese de Doutorado. Universidade de São Paulo.

STEWART, James. **Cálculo, volume I,** 4a.edição. São Paulo: Pioneira Thompson Learning, 2002.

SCHMITZ, Marcelo; HÜBNER, Jomi Fred. Uso de SMA para avaliar estratégias de decisão no controle de tráfego urbano. **Seminário de Computação em Blumenau-SC,** p. 243-254, 2002.

Wu, Shin - Ting; **EA978 - Sistemas de Informações Gráficas,** Campinas - São Paulo, 2009