



Universidade Federal de Pernambuco

Centro de Informática

Relatório de Projeto

Pedro Nogueira Coutinho - pnc
Riei Joaquim Matos Rodrigues - rjmr
Victor Hugo Meirelles Silva - vhms
Victor Miguel de Moraes Costa - vmmc2
Zilde Souto Maior Neto - zsmn

Recife, 23 de Agosto de 2021

Professor: Divanilson Rodrigo de Sousa Campelo

Índice

1	Introdução	1
2	Descrição do problema	2
3	Metodologia	4
3.1	Pré-processamento da imagem com filtros estatísticos	5
3.2	Transformadas de Fourier para obtenção dos contornos	9
3.3	Deteção e processamento dos contornos	12
4	Resultados	15
5	Conclusão	17

1 Introdução

O objetivo deste relatório é apresentar detalhes e etapas da construção do projeto da disciplina de Sinais e Sistemas, que tem como finalidade a aplicação de filtros estatísticos e transformadas de Fourier para resolver um problema relacionado ao processamento de sinais.

O relatório está dividido em três etapas: inicialmente apresentaremos o problema a qual nos dispusemos a resolver, em seguida apresentaremos as metodologias que foram aplicadas, descrevendo todo o processo para a construção da solução, onde, posteriormente, apresentaremos os resultados da nossa abordagem em comparação com abordagens do estado-da-arte.

2 Descrição do problema

O monitoramento de veículos que trafegam em vias públicas vem ganhando bastante atenção por ser uma necessidade constante e que cada vez mais vem apresentando desafios mais complexos. De longe, o maior destes desafios é o constante aumento da frota de veículos, que, segundo o Denatran¹ em maio de 2021 no Brasil a quantidade de automóveis já passa dos 100 milhões, ou seja, aproximadamente 1 carro para cada 2 brasileiros.

Esses números são tão expressivos que acabam comprometendo a eficácia de métodos de fiscalização, como, por exemplo: *blitzes* e inspeção manual dos documentos dos veículos. Uma boa alternativa que tem sido aplicada para melhorar a eficácia destes tipos de monitoramento é a automação desse serviço. Isso é feito através do uso de tecnologias que capturam imagens em vias de tráfego, em seguida, realizam captura das placas veiculares em tais imagens e, por fim, executam consultas no sistema de forma automática, de modo a cobrir uma quantidade muito mais expressiva de veículos.



Figura 1: As vias de tráfego vem apresentando cada vez mais veículos, apresentando um desafio para os métodos de fiscalização tradicionais

Apesar da alta eficiência destes sistemas autônomos, existem diversos obstáculos nesta tecnologia que, constantemente, são alvos de atenção e melhorias. Um dos maiores deles, sem dúvidas, é em como realizar a seg-

¹<https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/frota-de-veiculos-2021>

mentação minuciosa das placas dos veículos que trafegam na via somado ao problema de como reconhecer cada caractere que compõe a placa detectada.

Tendo isso em vista, a proposta do projeto elaborado é apresentar uma maneira de realizar a segmentação das placas veiculares em imagens. Para isso, fez-se o uso de filtros estatísticos em conjunto com a transformada de Fourier. Ao final, comparamos a performance da metodologia proposta com a de metodologias tradicionais.

3 Metodologia

O reconhecimento de placas de veículos em uma imagem envolvem duas etapas complexas e que demandam uma alta eficiência. Primeiro é necessário selecionar a região que contém a placa e depois a partir dessa seleção é preciso reconhecer qual é a numeração de identificação da placa.

Segmentar imagens é um problema amplamente estudado, sua complexidade vem principalmente dos ruídos que podem ser inseridos em uma ambiente real e dinâmico como as vias de estradas, onde temos uma grande variação de iluminação e reflexos nos veículos. Além disso, o design dos veículos geralmente possuem diversos formatos geométricos em sua composição que muitas vezes confundem e atrapalham o processo de segmentação dos algoritmos, que precisam filtrar e/ou até mesmo testar a maioria das segmentações em busca de uma placa.



Figura 2: Exemplo de como a segmentação pode ser dificultada: existem muitos contornos a serem analisados, principalmente do próprio automóvel

Durante nossas pesquisas notamos que muitas aplicações tradicionais utilizam detectores de bordas que, em sua composição, utilizam filtros gradientes para obter os contornos da imagem. Esses tipos de filtros são excepcionalmente úteis para a aquisição de contorno, mas acabam por detectar muitas vezes o contorno de objetos ruidosos no ambiente, gerando impactos negativos para o processamento e procura dos segmentos de interesse.

Por isso, é natural que muitas vezes sejam executados etapas de pré-processamento na imagem, tais como a aplicação de filtros de *blurring* para

borrar a imagem, e filtros lineares realçadores de contorno de modo a, não apenas eliminar o ruído das imagens, como também realçar o contorno dos objetos, melhorando a posterior etapa de segmentação. Esses filtros são excepcionalmente úteis porque mesmo que eles não forneçam um ganho visual aparente, ajudam bastante, principalmente, durante a detecção das bordas com o uso dos contornos, permitindo que o *shape* dos objetos fique devidamente contornado.

Com isso em mente, em nossa abordagem, tentamos fazer uso de Transformadas de Fourier em conjunto com filtros estatísticos de modo que pudéssemos realizar essas etapas de pré-processamento na imagem para o realçamento de contornos, eliminação de ruídos indesejados e aplicação das transformadas para a obtenção destes contornos através do deslocamento do espectro de magnitude e transformadas inversas. A seguir, mostraremos como que essas etapas de pré-processamento e aplicação de transformadas funcionam na prática.

3.1 Pré-processamento da imagem com filtros estatísticos

De início através do *opencv*, selecionamos uma imagem, usamos de exemplo a imagem a baixo (*como o tamanho original das imagens estapulam o limite do PDF, usaremos um tamanho padrão para todas, logo a diferença do tamanho da foto original para as filtradas apenas se fará perceptível ao usar o código de fato*):



Figura 3: Imagem original sem nenhuma alteração ou pré-processamento.

Em seguida padronizamos o seu tamanho através de um metodo implementado na biblioteca *imutils*, fazemos isso para evitar o alto custo temporal de pré-processamento e segmentação em imagens muito grandes. Após isso, usamos o *opencv* novamente para alterar a imagem selecionada para escala de cinza, pois detectar bordas de uma imagem em escala cinza é um processo mais facil e confiavel de obter bons resultados. Abaixo temos a imagem após essas alterações:



Figura 4: Figura 3 com o tamanho alterado e em escala cinza.

Tendo a imagem em escalas de cinza, utilizamos uma função para remover algumas reflexos de luzes nas superfícies do carro. Para isso, observamos a tonalidade de cada pixel e, dependendo do seu valor, esclarecemos ou escurecemos ele de forma a evitar diferenças bruscas de tonalidade da foto.



Figura 5: Figura 4 com o filtro de anti-reflexo aplicado

Usaremos agora um filtro estatístico para aumentar a nitidez da imagem, para o nosso projeto, isso significa que as bordas da imagem vão ficar mais enfatizadas, logo melhorando a detecção das mesmas. Esse filtro se trata de uma combinação linear do impulso de Dirac com filtro passa-alta: $\text{Dirac} + 0,9 \times \text{passa-alta}$. Para fazer isso primeiro precisamos de um kernel, que será uma matriz 3×3 :

$$\text{kernel} = \frac{1}{10} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 18 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Com o kernel definido, faremos uma convolução entre ele e a imagem fornecida.

$$\text{output}(x, y) = \sum_{x'=0}^3 \sum_{y'=0}^3 \text{kernel}(x', y') * \text{input}(x + x' - 2, y + y' - 2)$$

No output temos a imagem de entrada com o filtro aplicado, podemos vê-la abaixo:



Figura 6: Figura 5 com o filtro enfatiza borda aplicado

3.2 Transformadas de Fourier para obtenção dos contornos

Uma vez com a imagem pré-processada, podemos partir para a aplicação da transformada de Fourier. Em linhas gerais, essa transformada funciona da seguinte forma: Recebe como entrada um sinal no domínio do tempo e, por meio de operações matemáticas, é capaz de "quebrá-lo" em seus blocos componentes no domínio das frequências. Embora uma imagem não seja um sinal no domínio do tempo (e sim um sinal no domínio espacial), a transformada pode ser aplicada do mesmo jeito. Sendo assim, depois de sua aplicação, obtemos uma nova imagem que contém as frequências que compõem a imagem inicial. Nessa nova imagem, as frequências mais baixas estão presente na parte central da imagem, já as frequências mais altas estão ao redor dessa parte central (em direção às extremidades da imagem).

Em nossa abordagem, optamos pelo uso da FFT (Transformada Rápida de Fourier), devido principalmente à sua vasta aplicabilidade no ramo de processamento de imagens já que sua complexidade é de $O(N \cdot \log(N))$ (onde N é o tamanho da imagem) em contrapartida a transformada de Fourier clássica, que possui complexidade $O(N^2)$, o que nos permite processar a transformada de imagens (que geralmente possuem muitos pixels) de maneira mais rápida e eficiente. Na figura abaixo é possível ver o espectro de magnitude (à direita)

gerado pela FFT da imagem pré-processada (à esquerda).



Figura 7: Espectro de magnitude gerado a partir da imagem pré-processada

Uma vez com o espectro de magnitude podemos notar que as frequências ficam dispostas a partir do centro da imagem, ou seja, as menores frequências são as mais próximas ao centro e as maiores frequências são as mais distantes. Outro ponto muito importante é que a frequência de um pixel é, na verdade, o quão **homogêneo** ele é em comparação à seus vizinhos, ou seja, o quão semelhante em tonalidade ele é se comparado a sua vizinhança. Isso nos implica dizer que componentes de **alta frequência** são, na verdade, pixels muito distintos em tonalidade de seus vizinhos e componentes de **baixa frequência** são, portanto, pixels muito semelhantes em tonalidade a seus vizinhos.

Com essa propriedade em mãos, podemos aplicar um **HPF** (Filtro de Passa Alta), ou seja, um filtro que nos permita eliminar as baixas frequências e manter as maiores. Com isso, estaremos eliminando as regiões de preenchimento da imagem, ou seja, as regiões onde os pixels uma vez convertidos para o domínio da frequência apresentam baixa frequência. Podemos fazer a aplicação desse filtro simplesmente eliminando um raio a partir do centro no espectro de magnitude gerado, o que nos fará, portanto, eliminar uma certa quantidade de componentes de baixa frequência.

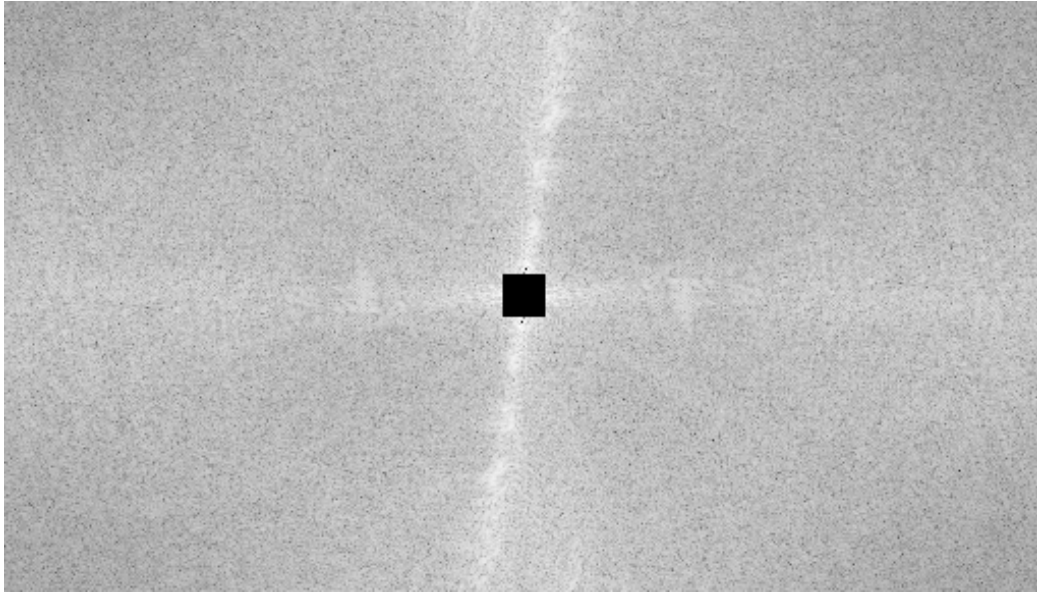


Figura 8: Espectro de magnitude após aplicação do filtro de passa alta

Uma vez feito a aplicação do HPF, podemos aplicar a **IFFT** (transformada rápida inversa de Fourier) para voltarmos ao domínio espacial através do domínio da frequência. Como fizemos a eliminação de uma certa quantia de elementos de baixa frequência, é esperado que tenhamos como resultado uma imagem com o preenchimento eliminado e contornos mantidos e destacados, já que estes últimos são representados pelas componentes de alta frequência que foram mantidas pelo filtro.



Figura 9: Resultado do processamento com **IFFT**

3.3 Detecção e processamento dos contornos

Para esta etapa de detecção de contornos, utilizaremos a função **findContours** da biblioteca **OpenCV** em **Python**. Um ponto importante desta função é que para seu funcionamento correto ela necessita de uma imagem que esteja normalizada e com valores de pixel que possuam valor 0 ou 255 (exclusivamente).

É importante destacar que, após a aplicação da FFT e da IFFT, a imagem recebida no domínio espacial pode apresentar alguns pixels com valor negativo e, conseqüentemente, a detecção de contornos não ocorreria de forma adequada. Sendo assim, é feita uma normalização de todos os pixels presentes na imagem para que o valor de cada um deles esteja dentro do intervalo $[0, 255]$. Após isso, precisamos garantir que os pixels possuam apenas valor 0 ou 255, exclusivamente. Para isso, aplicamos um thresholding nos pixels de forma que forcamos eles a assumir apenas esses dois valores.

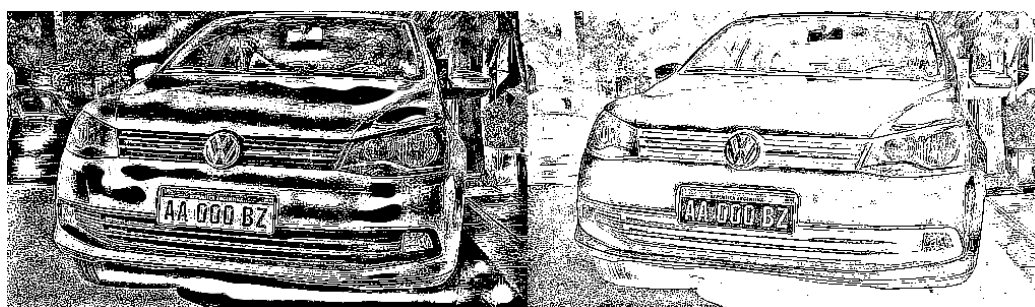


Figura 10: Retorno da imagem ao domínio espacial (à esquerda) e aplicação da normalização e thresholding (à direita)

Uma vez feita a passagem da nossa imagem processada para a função **findContours** do **OpenCV**, receberemos uma lista contendo os pontos que compõem cada um destes contornos. Com isso em mãos, decidimos realizar uma aproximação retangular no contorno, ou seja, gerar para cada contorno na imagem um retângulo aproximado que contenha seus pontos. Como sabemos, o formato das placas de automóveis é sempre retangular, onde a base do retângulo é maior do que sua altura com uma certa proporção. Diante disso, podemos filtrar os retângulos gerados para os contornos, eliminando os que definitivamente não possuem o formato da nossa placa procurada.

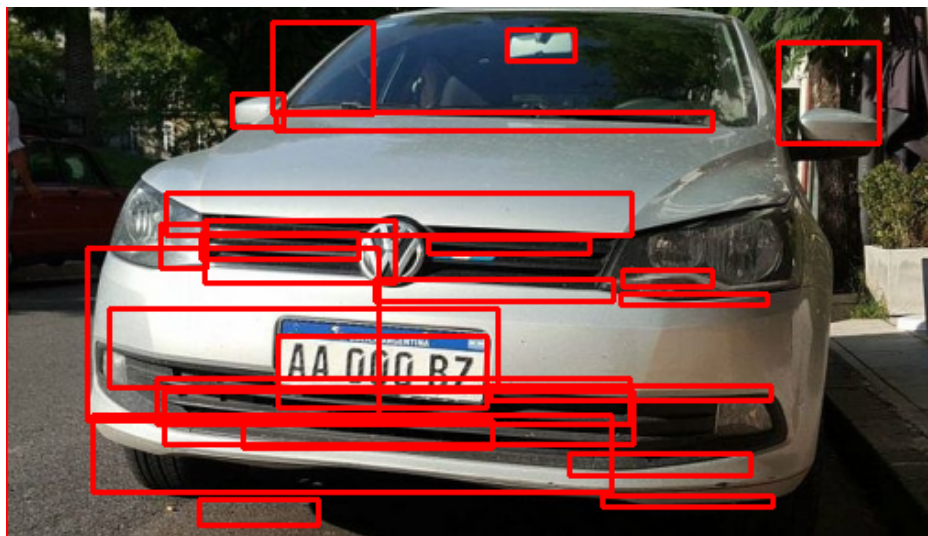


Figura 11: Regiões de interesse encontradas

Apesar disso, como visto na imagem acima, muitos retângulos obedecem a essa condicional (incluindo o retângulo que contém a placa). Infelizmente, não existem formas (apenas usando segmentação) de descobrir qual retângulo contém a placa, mas podemos fazer uso de redes neurais que são capazes de detectar caracteres em uma imagem. Não obstante, apenas o fato de termos gerados diversos segmentos a partir da imagem já nos dá uma clara vantagem de não termos que realizar o processo de busca na imagem completa, pois agora possuímos regiões de interesse nas quais podemos aplicar qualquer técnica que seja capaz de reconhecer caracteres.

Portanto, usando a biblioteca **easyocr**, que tem como finalidade a detecção de caracteres em uma imagem, passamos todos as regiões de interesse geradas pela nossa filtragem e mantivemos na imagem apenas as regiões de interesse onde foram detectados caracteres. Apesar de em alguns casos algumas regiões além da região contendo a placa serem marcadas, seja devido à logomarca do veículo, ou ao modelo (que geralmente é estampado na traseira), ou outro tipo de ruído, conseguimos garantir para uma grande porcentagem das imagens do dataset usado que a placa vai ser detectada em alguma região de interesse. Essa informação pode ser posteriormente filtrada observando os caracteres detectados e analisando se eles possuem o formato de uma placa válida de algum país de interesse.



Figura 12: Regiões de interesse validadas

4 Resultados

Para comparar os resultados da nossa abordagem primeiro precisaríamos de um conjunto de imagens. Por isso, fizemos uma busca por imagens de carros com placas visíveis no Google Images, onde conseguimos montar um dataset que contém **48 imagens** com carros de diferentes modelos, em diferentes condições de luminosidade e com diferentes modelos de placas.

Uma vez montado o dataset, realizamos uma procura por uma abordagem clássica do estado-da-arte de modo que pudéssemos comparar as duas abordagens. Aqui, fazemos uma análise qualitativa e uma análise quantitativa de modo a identificar quantas regiões de interesse marcadas englobam as placas dos automóveis. Na análise qualitativa, é verificado apenas se o nosso modelo proposto consegue construir regiões de interesse que englobam a placa (independente do formato/tamanho da região). Na análise quantitativa, são verificados 3 situações possíveis envolvendo as regiões de interesse construídas:

- **Situação 1:** são computados os casos em que a região de interesse apresenta um contorno coincidente com a placa do automóvel.

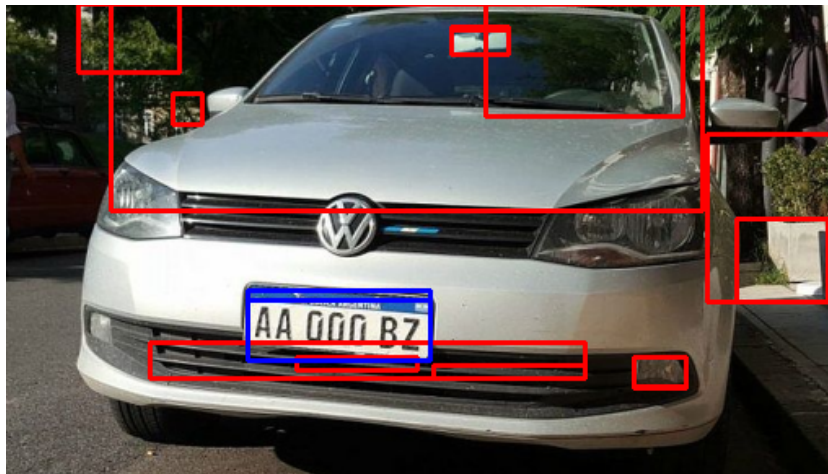


Figura 13: Exemplo de situação onde a placa é perfeitamente contornada

- **Situação 2:** são computados os casos em que a região de interesse engloba a placa mas possui um tamanho maior do que o da própria placa, mas que a placa ocupa pelo menos 20% da area selecionada.

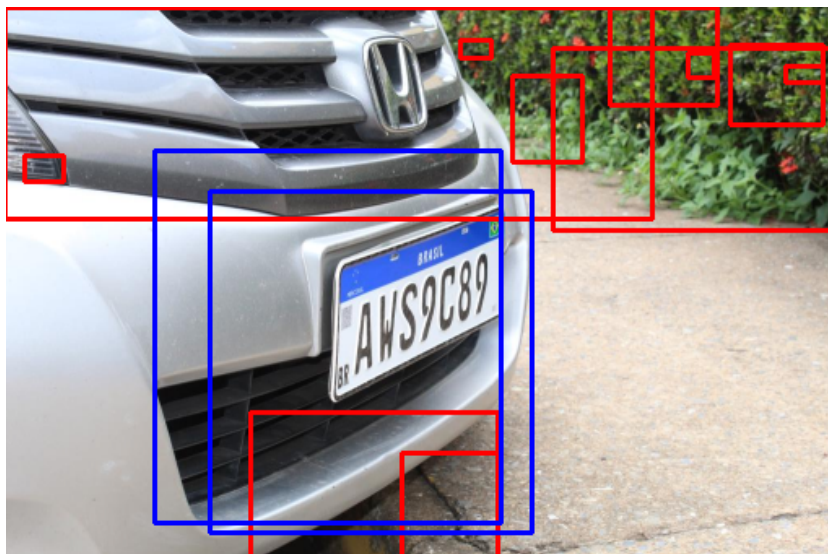


Figura 14: Exemplo de situação onde a placa é englobada, mas com uma área extra ao seu redor

- **Situação 3:** são computados os casos em que a região de interesse engloba parcialmente a placa automóvel ou que não foi possível reduzir a imagem a zonas menores de interesse, todo o carro foi selecionado.

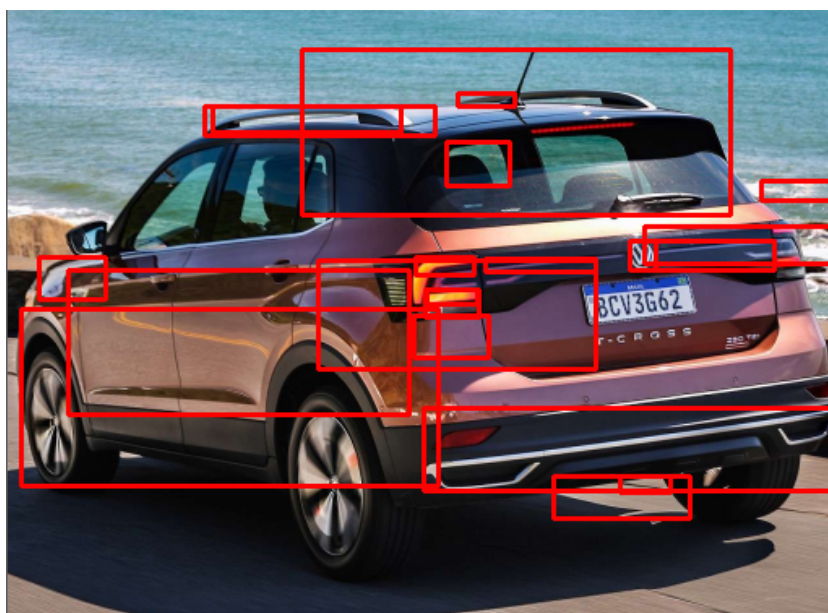


Figura 15: Exemplo de situação onde não foi possível segmentar a placa

Mostraremos agora os resultados obtidos utilizando as métricas apresentadas:

1. Análise Qualitativa

	Nossa abordagem	Abordagem OpenCV
Taxa de Reconhecimento	100,00% (48/48)	79,17% (38/48)

2. Análise Quantitativa

	Nossa abordagem	Abordagem OpenCV
Situação 1	77,08% (37/48)	60,42% (29/48)
Situação 2	22,92% (11/48)	18,75% (9/48)
Situação 3	0,00% (0/48)	20,83% (10/48)

5 Conclusão

Ao final do projeto, é perceptível que a abordagem de pré-processamento através de filtros estatísticos aliados às operações aritméticas sobre a matriz da imagem e da transformada de Fourier apresentada pelo grupo possui uma maior eficácia nos resultados, em comparação ao método tradicional do OpenCV. Dito isso, não há necessidade de partir para abordagens mais custosas, pois o método utilizado pela equipe gera uma imagem cujas bordas ficam mais destacadas, facilitando assim o trabalho do algoritmo de identificação de bordas, que usa uma conversão baseada em gradientes. Finalmente, é evidente a aplicabilidade da transformada de Fourier (e de sua transformada inversa) para o processamento de imagens, bem como a aplicabilidade de filtros estatísticos para resolver problemas desse mesmo tipo.