Uso de transformada de Fourier e de métodos estatísticos em imagens segmentadas.

Felipe Jun Ting Lin

João Gabriel de Araujo Vasconcelos

Lucas Cavalcanti Calabria

Matheus Machado Guilhermino

Vinicius Revoredo de Albuquerque

1. Objetivo

A segmentação de imagens é o processo de dividir uma imagem digital em regiões distintas, destacando e simplificando objetos e formas para possibilitar análises e identificações.

Neste projeto será utilizado a detecção de bordas como tipo de segmentação, a partir das Transformadas de Fourier, filtros e o Método Estatístico Gaussiano com o objetivo de identificar placas de trânsito.

2. Metodologia

2.1 Ambiente de Desenvolvimento

A realização desse projeto foi feita no Google Colab, onde toda a parte do código está sendo executada nesse ambiente de desenvolvimento.

2.2 Bibliotecas Utilizadas

```
[ ]  import cv2
     import imutils
     import easyocr
     import numpy as np
     import matplotlib.pyplot as plt

     from os import listdir
     from math import sqrt,exp
     from google.colab import files
     from os.path import isfile, join
```

- cv2 - Serve para resolver problemas usando visão computacional.
- imutils - Tem funções para realizar processamento de imagem e ajuda na identificação de bordas e contornos.
- easyocr - Permite aos desenvolvedores de visão computacional realizar o reconhecimento óptico de caracteres sem muito esforço.
- numpy - Adiciona estruturas de dados potentes ao Python que garantem cálculos eficientes com array e matrizes e fornece uma enorme quantidade de funções matemáticas de alto nível que operam sobre esses componentes.
- matplotlib - Responsável por criar as visualizações estáticas e interativas de dados no Python.

2.3 Importação de Imagens

Nessa etapa foi feita a importação de imagens para popular o banco de dados e posteriormente, realizar os processamentos de imagem para a obtenção do resultado final.

2.4 Transformadas de Fourier Utilizadas

2.4.1 Transformada Discreta de Fourier

A Transformada Discreta de Fourier utiliza um conjunto limitado, porém suficiente, de frequências para descrever a imagem do domínio espacial para o domínio de frequência. A DFT é dada pela fórmula de:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i*2\pi(\frac{ki}{N}+\frac{lj}{N})}$$

Onde a equação descrita acima serve para uma imagem de tamanho NxN e a função $f(i,j)$ representa o valor de pixel em escala de cinza da coordenada da imagem.

2.4.2 Transformada Rápida de Fourier

A chamada Transformada Rápida de Fourier, FFT, é apenas uma otimização do método da Transformada Discreta de Fourier, que realiza os cálculos computacionais no tempo aceitável e menos custoso, sendo assim, uma ótima alternativa para a realização de tratamento de imagens.

2.5 Método de Estatística Gaussiana

Neste ponto já foram selecionadas possíveis regiões que correspondem à placa de trânsito. O método estatístico tem como objetivo determinar a região da placa.

O método estatístico de Gauss utiliza as propriedades dos pixels, como cor, intensidade, textura e continuidade para detectar discrepâncias e variações entre pixels próximos. Para cada região candidata o método é aplicado em blocos de pixel e o resultado é analisado. O tamanho do bloco escolhido foi 4x4.

A comparação entre os pixels é realizada pela fórmula:

$$T_{f_1}(p,q) = \sum_{(m,\,n)\in\Omega_1} [f_1(m,n) - f_1(m+p, n+q)]^2,$$
$$\text{for } (p,q) \in \{(0,1),(1,0),(1,1),(1,-1),\ldots\}$$

Onde, para este projeto, $m_{max} = n_{max} = 4$. O resultado é uma imagem mais simples e abstrata. Calcula-se então a média de proximidade dos pixels. A região correspondente à placa de trânsito tende a ter uma média sempre menor que as demais regiões. Isso ocorre devido ao fato da maioria das placas de trânsito possuírem uma cor de fundo constante em contraste apenas com a cor das palavras. Isso gera continuidade nas características dos pixels da placa. Por exemplo, escolhido um bloco de pixel, provavelmente muitos destes pixels vão representar o fundo constante da placa.

Portanto, a região escolhida será a de menor média.

2.6 Processo de filtragem (Filtros Passa-Alta)

Os filtros passas-altas, no qual, utiliza a técnica para realçar os detalhes e destacar as bordas de uma imagem (*Image Sharpening*), removendo os componentes de baixa frequência da imagem e preservando os componentes de alta frequência. Existem vários tipos de filtros passa alta, como por exemplo: filtro ideal passas-altas, filtro de Butterworth passa-alta e filtro de Gaussiano passa-alta.

2.6.1 - Filtro Ideal Passa-Alta

Dessa forma, a fórmula do filtro ideal passa-alta pode ser expressada na forma de:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Onde D0 é uma constante positiva que indica o limiar da nossa função que é um ponto de transição entre H(u,v) = 1 e H(u, v) = 0, denominado como frequência de corte, sendo assim, o *threshold* do filtro;

D(u,v) é a distância euclidiana da origem do plano da frequência até qualquer coordenada (u,v) da matriz.

2.6.2 Filtro de Butterworth Passa-Alta

A fórmula do filtro de Butterworth passa-alta pode ser expressada na forma de:

$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

Onde D0 é um limite definido como frequência de corte, sendo assim, o *threshold* do filtro;

D(u,v) é a distância euclidiana da origem do plano da frequência até qualquer coordenada (u,v) da matriz;

n é o número de elementos do filtro.

2.7 Transformações morfológicas

Essas operações fazem manipulações na forma da imagem que normalmente é executada em imagens binárias. Para a utilização dessa função, ela precisa de dois parâmetros, o primeiro é a imagem original e o segundo é a chamada de elemento estruturante, no qual, vai decidir a natureza das operações morfológicas. Existem 2 principais operadores morfológicos:

- erosão
- dilatação

Também tem outros operadores secundários:

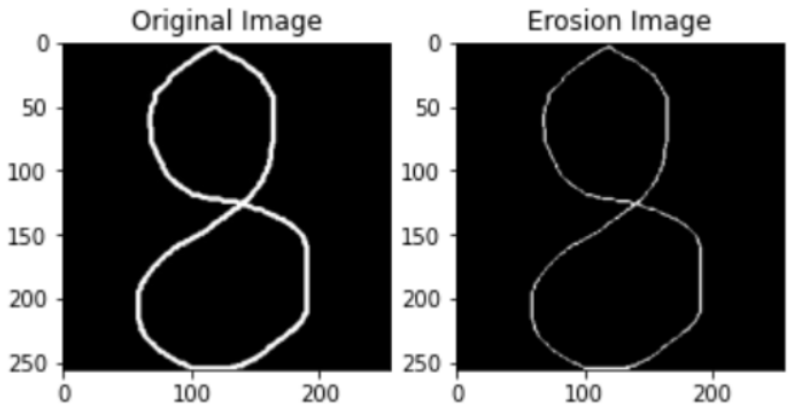
- abertura
- fechamento

2.7.1 Operações Morfológicas

1 - Erosão

A ideia dessa operação é desgastar os limites do objeto que estão em primeiro plano. Sendo assim, o que acontece nesse elemento estruturante performa na imagem toda, deslizando da esquerda para a direita e de cima para baixo sobre a imagem.

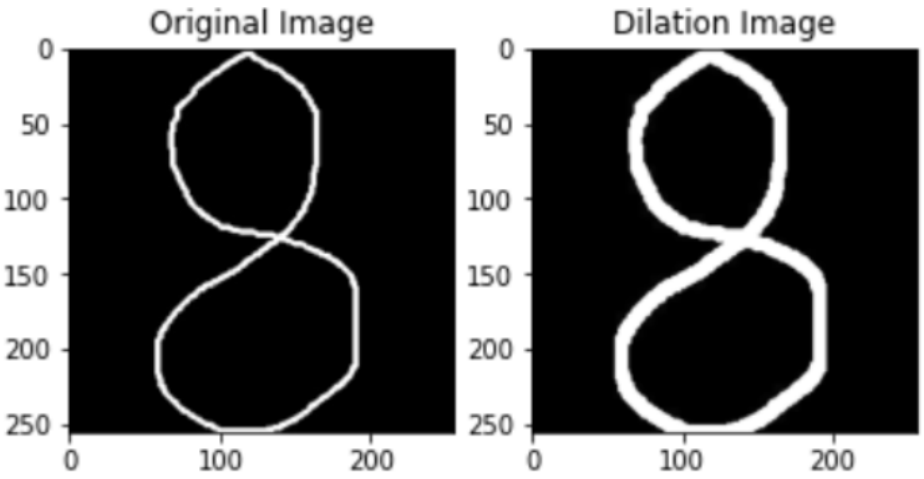
O pixel de primeiro plano (1 ou 0) na imagem original será mantido se todos os pixels dentro do elemento de estruturação forem 1 (objeto da imagem estruturada) . Caso contrário, os pixels serão definido como 0 (fundo da imagem)



Logo, a aplicação desse operador é muito útil para remover os ruídos brancos ao redor da forma e realçar os limites de dois objetos conectados. Para utilizar esse operador, usa-se a função cv2.erode().

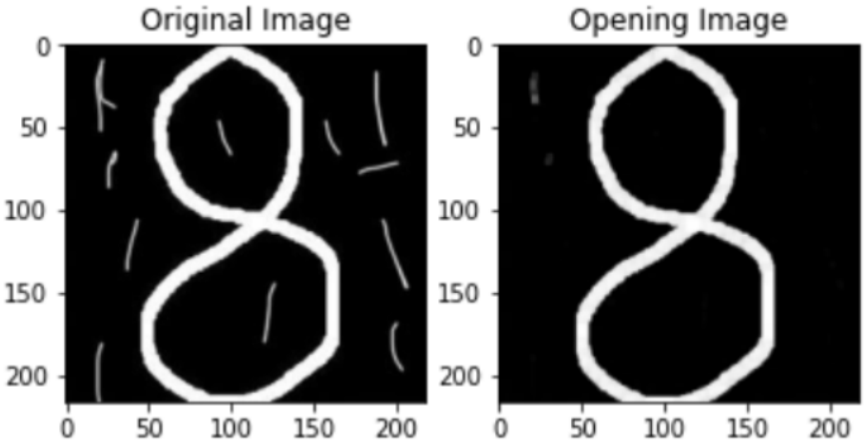
2 - Dilatação

Esse operador é o contrário da erosão. Devido ao fato de que o propósito do uso da erosão é para remover os ruídos da imagem, isso inevitavelmente vai diminuir o tamanho da imagem. Sendo assim, o uso de dilatação é aumentar o tamanho do objeto e como os ruídos já foram removidos previamente, o uso desse operador é geralmente seguido da erosão. Para utilizar esse operador, usa-se a função cv2.dilate().



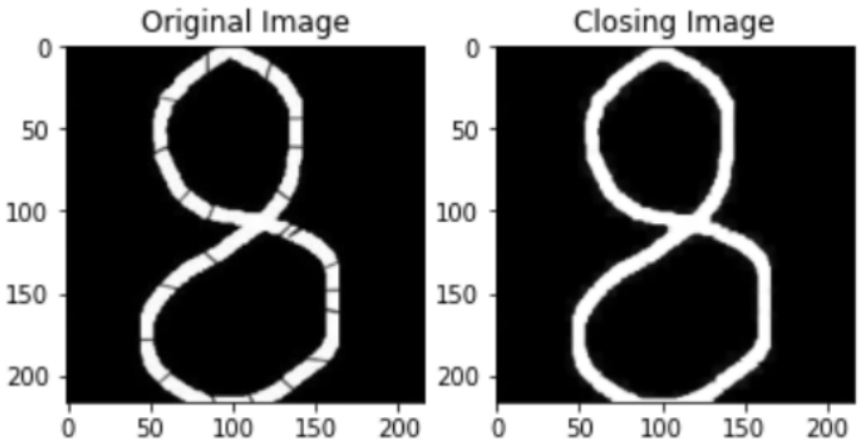
3 - Abertura

Essa operação é apenas um outro nome para erosão seguida de dilatação, cujo propósito é remover os ruídos. Para utilizar essa operação, usa-se a função cv2.morphologyEx(cv2.MORPH_OPEN).



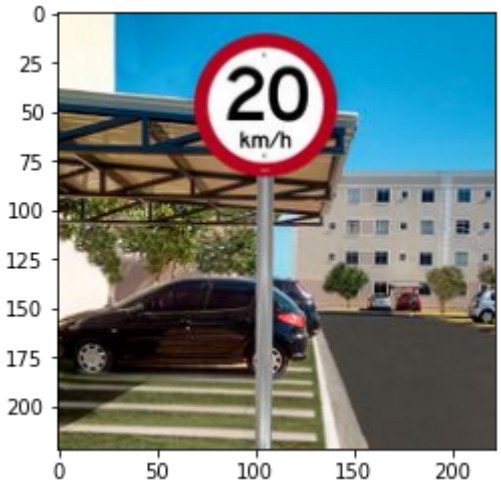
4 - Fechamento

Essa operação é o inverso da abertura, aplica-se o conceito de dilatação seguido da erosão. É bastante útil para fechar pequenos buracos nos objetos que estão em primeiro plano. Para utilizar essa operação, usa-se a função cv2.morphologyEx(cv2.MORPH_CLOSE).

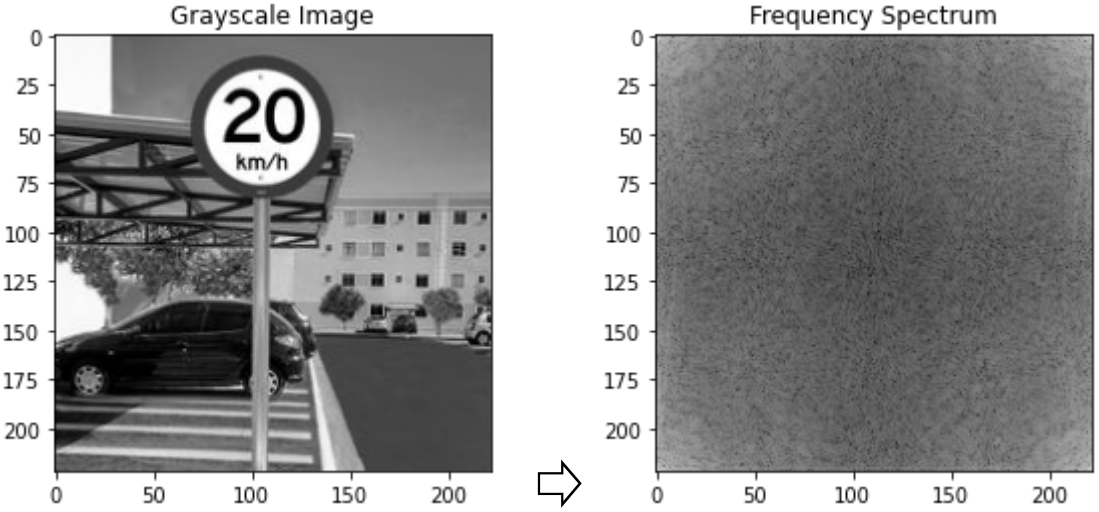


3. Resultados

A seguir são apresentados os resultados conforme cada passo é aplicado na imagem. Como exemplo vamos utilizar a seguinte imagem:

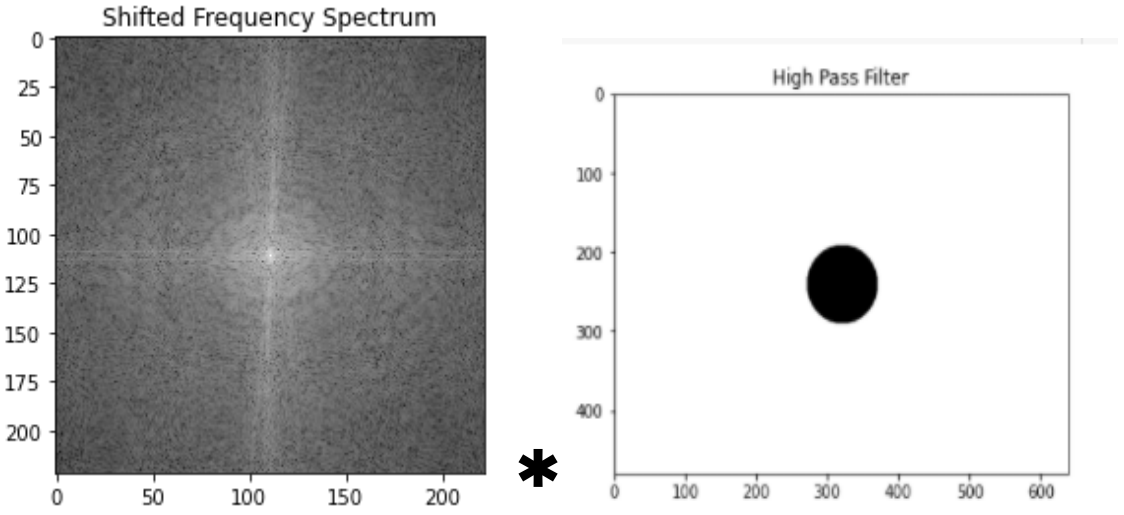


3.1 Aplicando a Transformada de Fourier

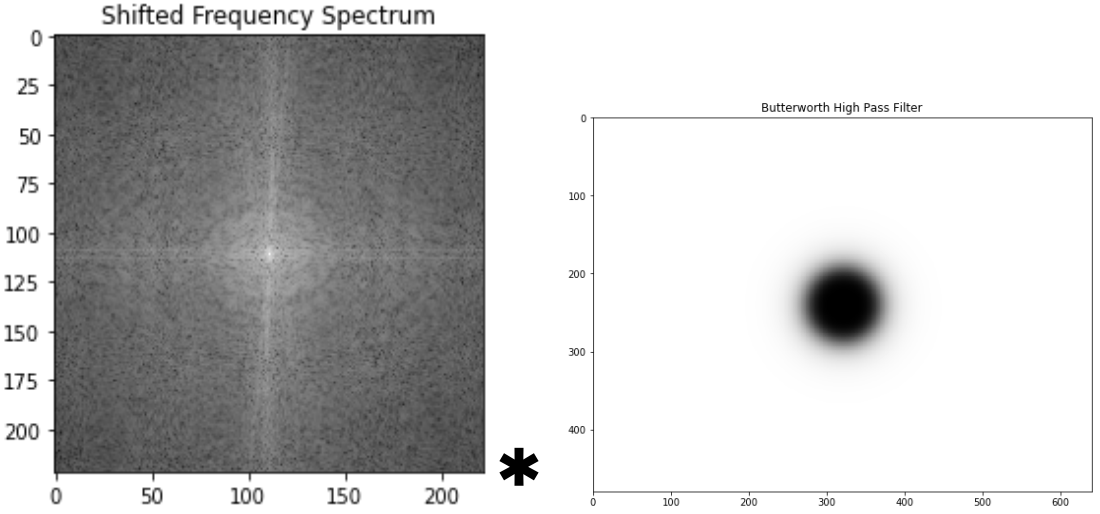


3.2 Preparando o Espectro e Aplicando os Filtros

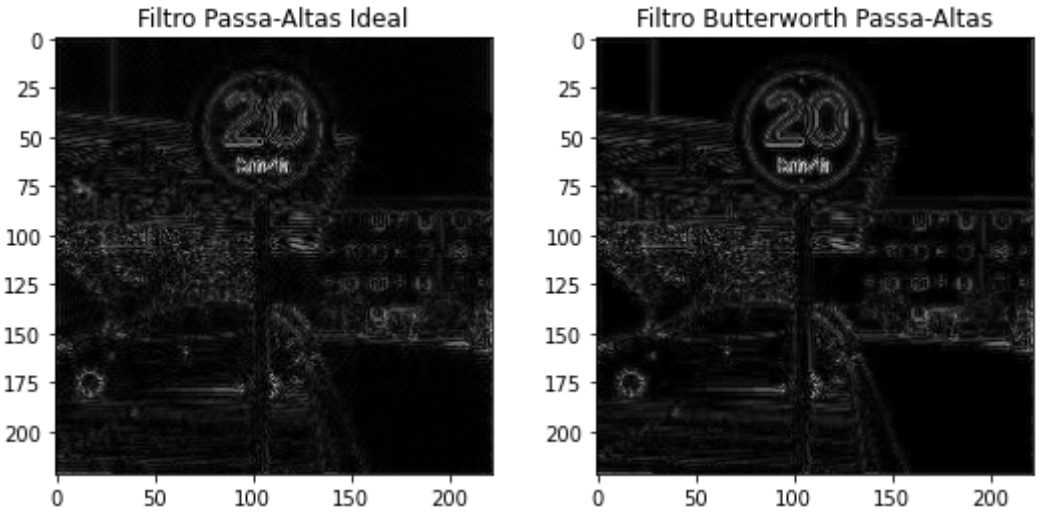
3.2.1 Filtro Ideal Passa-Altas



3.4.2 Filtro butterworth Passa-Altas

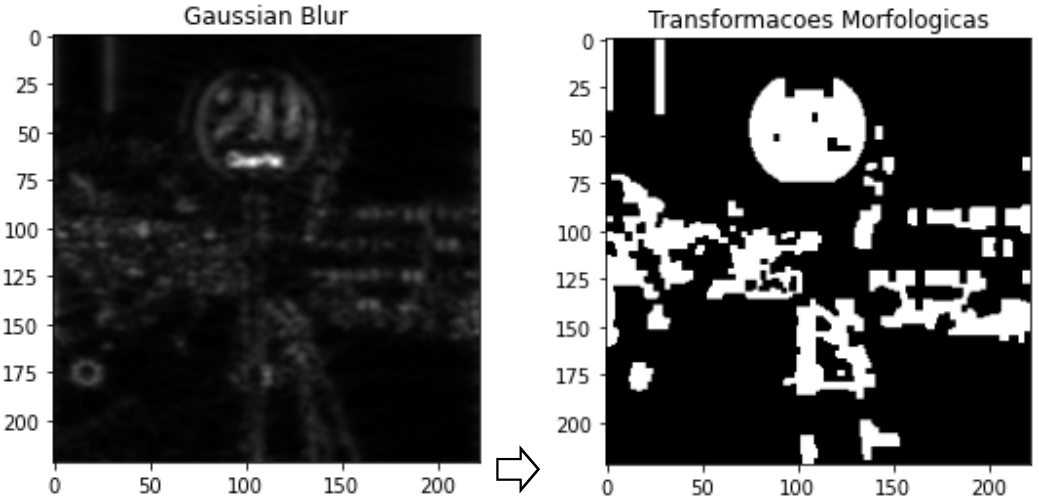


3.3 Descentralizando o resultado do filtrado e Aplicando a Transformada Inversa de Fourier

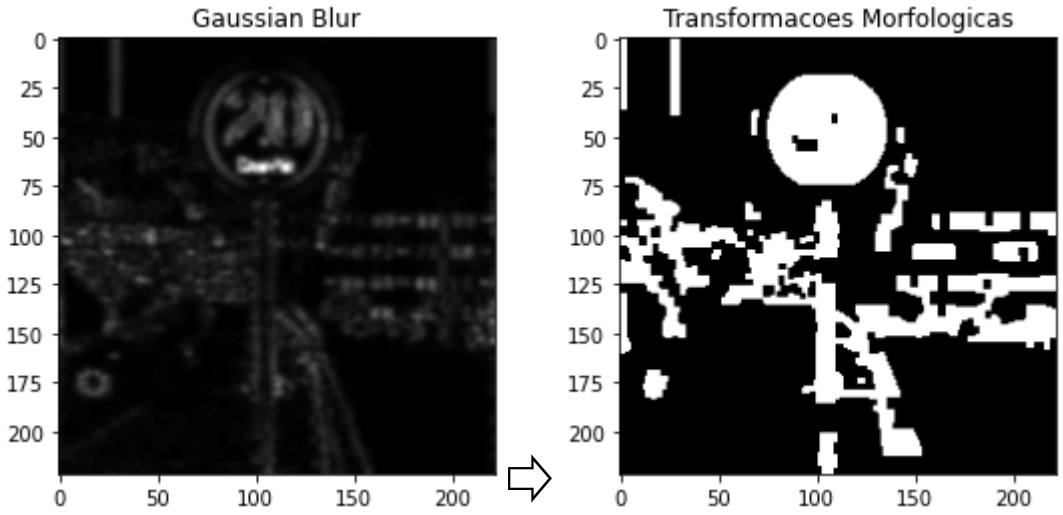


3.4 Aplicando o Gaussian Blur e realizando as Transformações Morfológicas

3.4.1 Filtro Ideal Passa-Altas

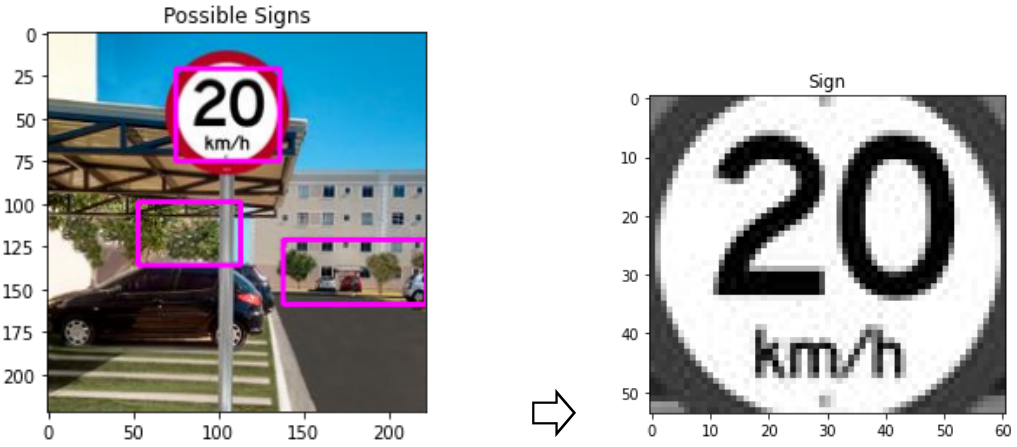


3.4.2 Filtro butterworth Passa-Altas



Neste ponto, as regiões que são possíveis placas são escolhidas.

3.5 Aplicando o Método Estatístico Gaussiano

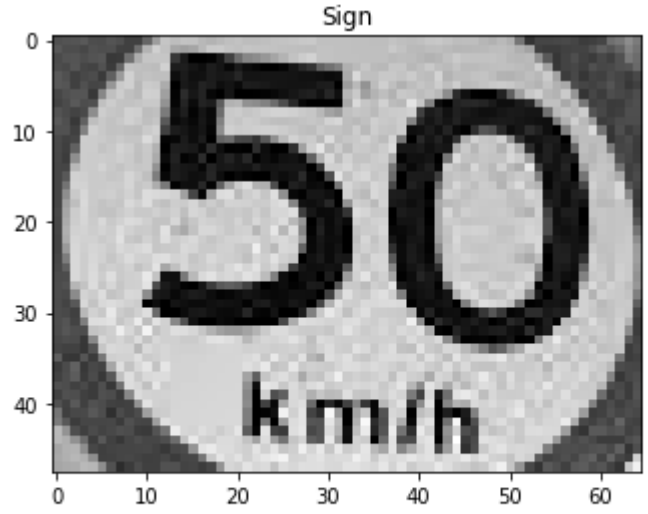


3.6 Leitura de caracteres da placa

Nessa etapa, foi utilizado o método reader.readtext() da biblioteca EasyOCR para identificar cada caractere da imagem da região

selecionada e formar o texto apresentado. Primeiramente, é necessário configurar o idioma para que essa função possa entender a linguagem, em seguida, a função vai receber a imagem final pós-processada como parâmetro para poder identificar o texto e transcrevê-lo.

```
24.jfif
```




```
Text: 50
Text: 50kmh
```

Como pode se observar no resultado acima, mesmo tendo vários candidatos para serem a parte da imagem desejada, através do método estatística gaussiana, o algoritmo conseguiu interpretar que a placa de “50 km/h” era a região de interesse para que o algoritmo da EasyOCR pudesse interpretar o texto e registrá-lo.

4. Conclusões

O projeto utiliza os conceitos da Transformada de Fourier, do Método Estatístico Gaussiano e técnicas como destaque de contornos e transformações morfológicas para determinar placas de trânsito em imagens.

Um ponto importante em relação à precisão é a forma escolhida para o funcionamento do método estatístico gaussiano. Uma vez calculada a média total das discrepâncias de pixels, para cada região de interesse, é selecionada a região de menor média. Entretanto, existem alguns cenários onde essa estratégia não é acurada. Por exemplo, para uma região de interesse formada majoritariamente por cores homogêneas, como o céu sem nuvem, a média de discrepâncias vai ser possivelmente menor do que a região contendo a placa. Para imagens compostas com mais estruturas, a região da placa será a região com a menor média.

Em relação aos filtros utilizados, o filtro ideal passa-alta apresentou maior acurácia em relação ao filtro Butterworth passa-alta para o banco de dados utilizado.

De maneira geral, um projeto estruturado em inteligência artificial e machine learning poderia aumentar a precisão dos resultados. Contudo, a precisão deste projeto é satisfatória levando em consideração o nível de sofisticação dos métodos utilizados e a arquitetura acessível.