

Luiz Henrique Botega Cervantes

bi3008479

Eng201

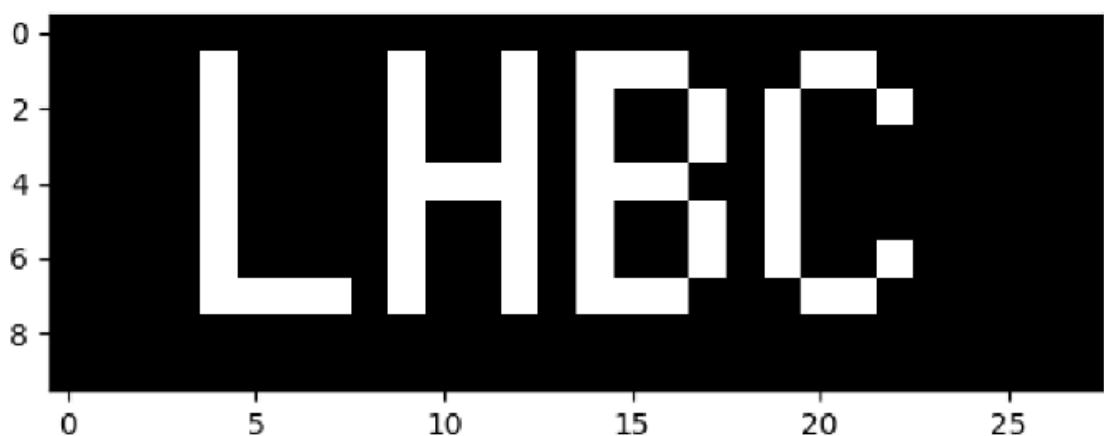
Compilado Trabalhos PDI

Código em Python e iniciais do nome geradas com NumPy

Código:

<https://github.com/LuizHenriqueCervantes/pdi/tree/95390658441a8a5e60ab14df1dfe3eb6901213f1/Pixels>

Resultado:



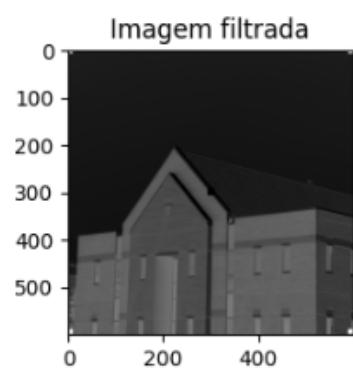
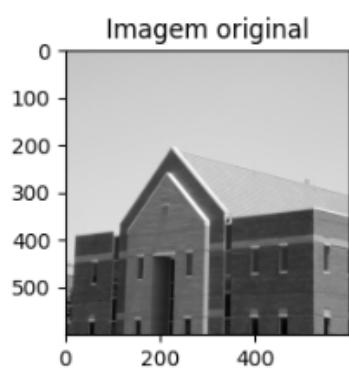
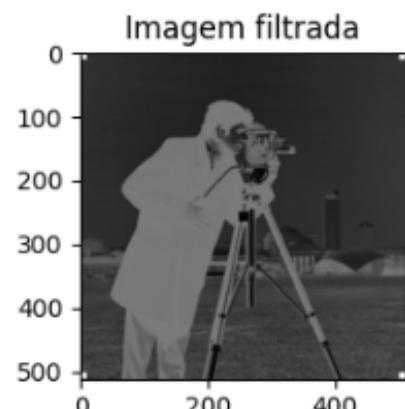
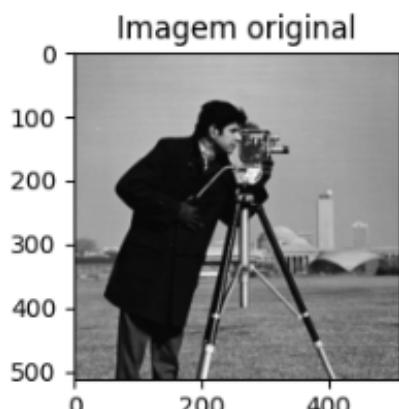
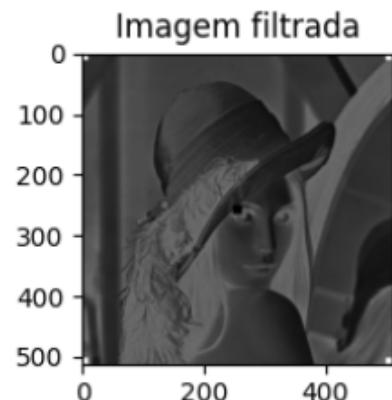
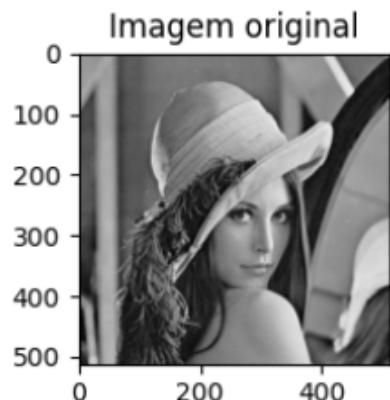
Fundamentos

- 1) Operação ponto a ponto.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosPontoaPonto.py>

Resultados:



Operações ponto a ponto em Processamento Digital de Imagens (PDI) são aquelas em que cada pixel na imagem de entrada é processado independentemente dos outros pixels. Isso significa que o resultado da operação em um determinado pixel não depende dos valores dos pixels vizinhos. As operações ponto a ponto são fundamentais no PDI e são frequentemente utilizadas para realizar manipulações simples nos valores dos pixels de uma imagem.

São eficientes e, muitas vezes, são implementadas usando operações vetorizadas em bibliotecas como NumPy para melhorar o desempenho. Elas são a base para muitos algoritmos de processamento de imagens mais complexos.

2) Operação por vizinhança.

Código:

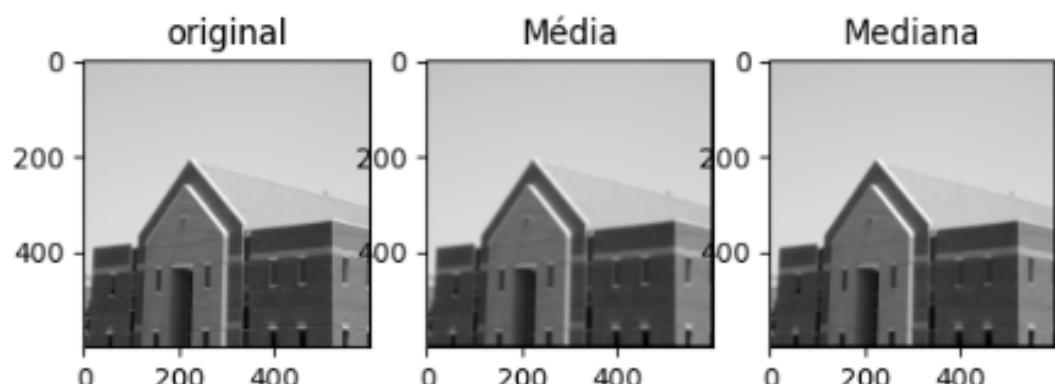
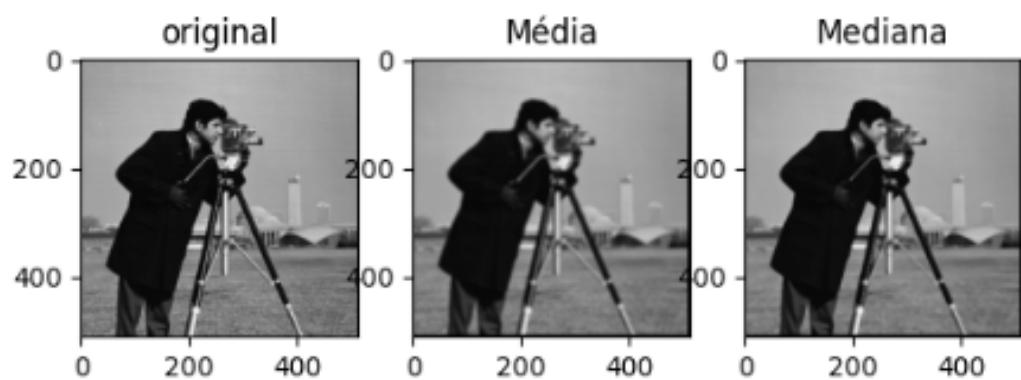
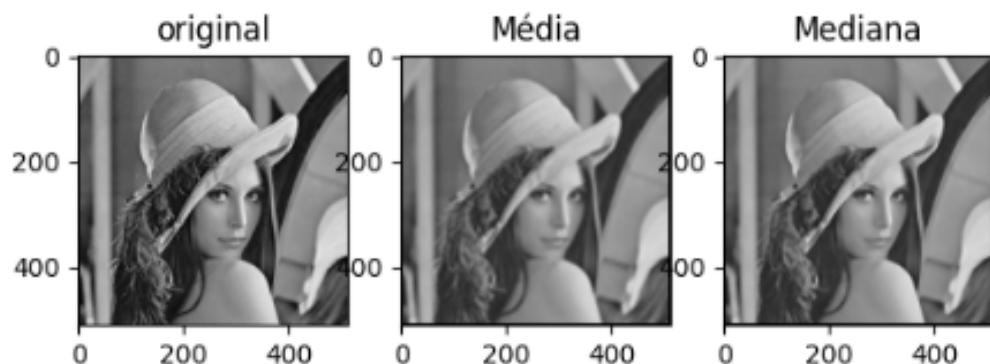
<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosOpPorVizinhanca.py>

Nas operações por vizinhança o valor de um pixel em uma imagem pode ser alterado com base nos valores dos pixels vizinhos. A vizinhança de um pixel refere-se aos pixels que o rodeiam em uma determinada região. Existem várias operações por vizinhança comuns, sendo algumas delas:

- Filtragem Média:
- Filtragem de Ponto Médio:
- Detecção de Bordas:
- Operações de Máscara:
- Erosão e Dilatação:
- Segmentação:

A vizinhança pode ser definida de várias maneiras, como 4-vizinhança (considerando pixels acima, abaixo, à esquerda e à direita) ou 8-vizinhança (considerando também os pixels nas diagonais).

Resultado:



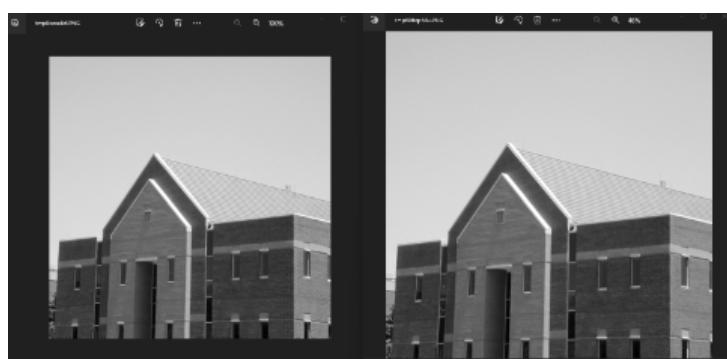
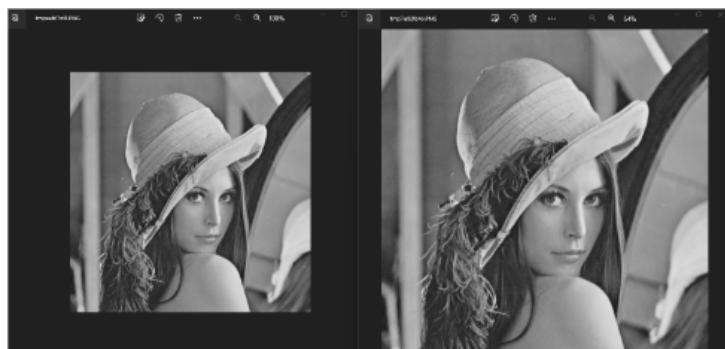
3) Zoom in

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosZoomIn.py>

O *Zoom In* em uma imagem digital é um processo que amplia uma parte específica da imagem, tornando-a mais visível e ocupando uma porção maior da tela. Existem várias maneiras de implementar o *zoom in* em imagens digitais sendo a escolha do método de zoom dependente das necessidades específicas da aplicação e da qualidade desejada na imagem ampliada.

Resultado:



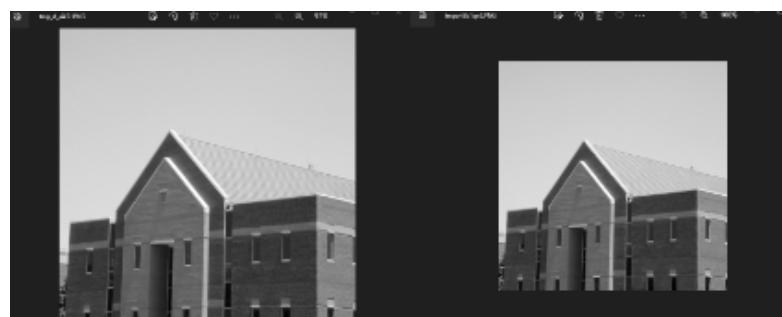
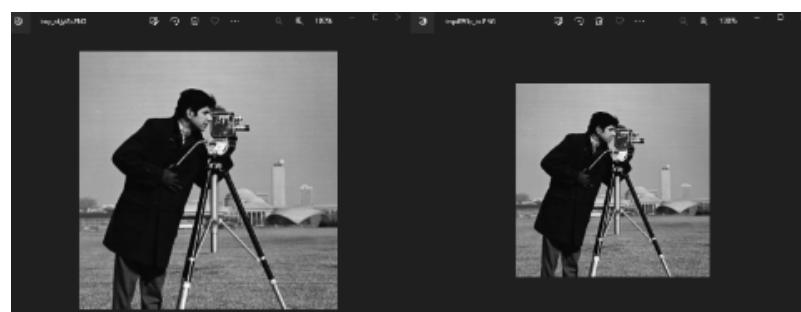
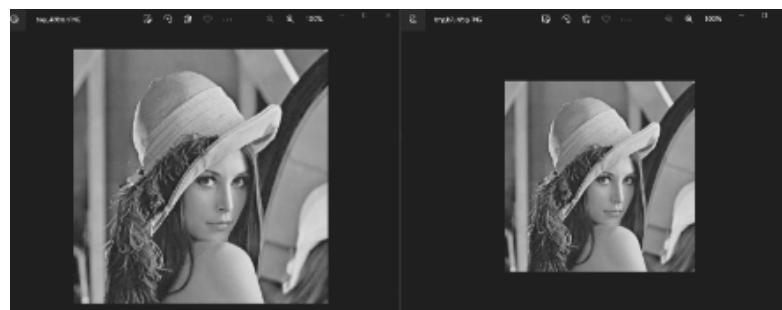
4) Zoom out

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosZoomOut.py>

Sendo o processo inverso do *Zoom In*. Ele envolve a redução do tamanho da imagem, mostrando a imagem original em uma porção menor da tela. Assim como no caso do zoom in, há várias abordagens para implementar o zoom out e o método escolhido também depende da aplicação.

Resultado:



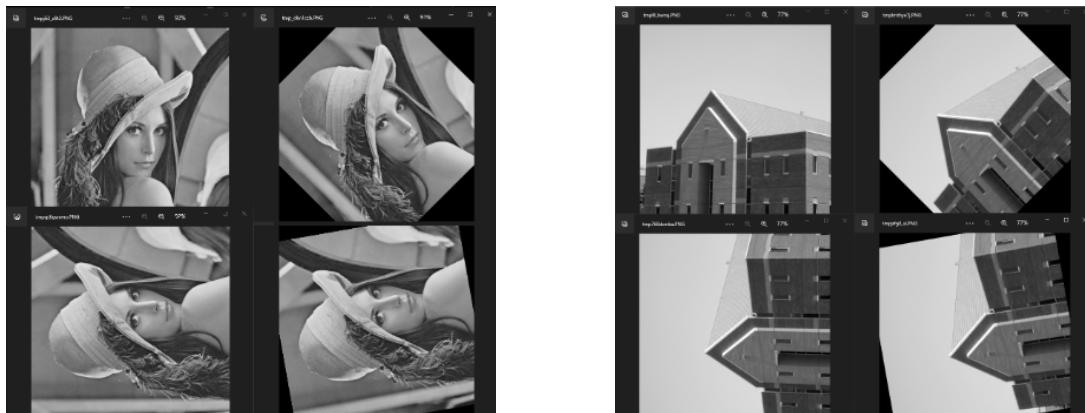
5) Rotação

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosRotacao.py>

A rotação de uma imagem envolve a transformação dos pixels da imagem para mudar sua orientação angular sendo a escolha do método dependente dos requisitos específicos da aplicação. Rotações pequenas podem ser realizadas eficientemente por métodos simples, enquanto rotações maiores ou mais complexas podem exigir técnicas mais avançadas.

Resultado:



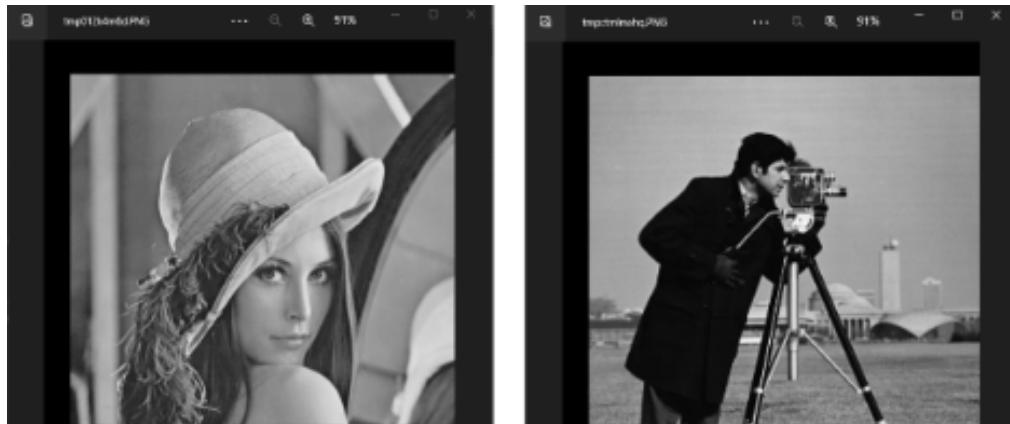
6) Translação

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosTranslacao.py>

Envolve o deslocamento de todos os pixels de uma imagem em direções específicas ao longo dos eixos x e y. A ideia básica é mover a imagem de uma certa quantidade de pixels em uma direção específica, criando a ilusão de que a imagem foi movida.

Resultado:



Intensidade

A intensidade se refere à luminosidade ou brilho de um pixel em uma imagem. Ela é uma característica fundamental e é comumente representada por valores numéricos que indicam a quantidade de luz ou cor em uma determinada posição na imagem.

Histograma

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/777092ce58863cfa535aa5733725028a4910ac69/Intensidade/histograma.py>

Resultados:

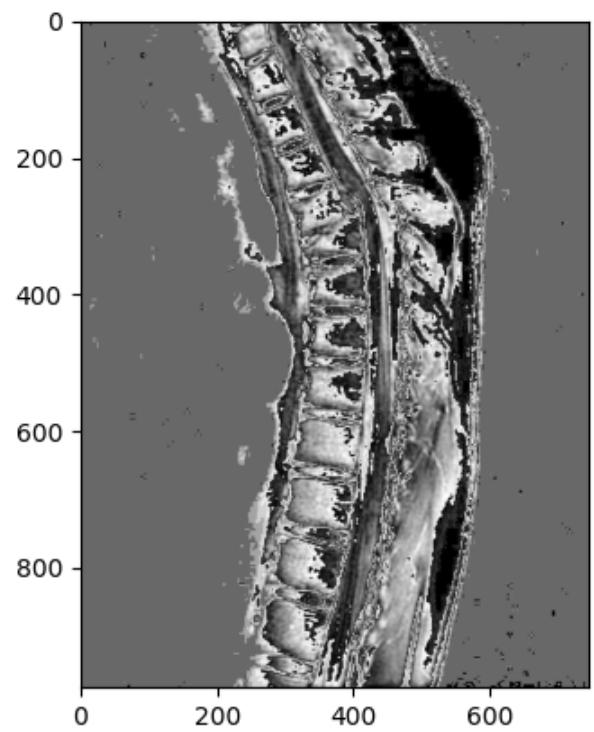


Logaritmica

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/777092ce58863cfa535aa5733725028a4910ac69/Intensidade/logaritimica.py>

Resultados:

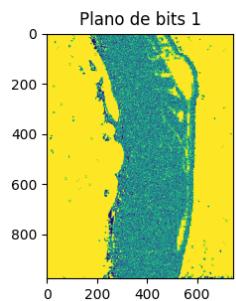
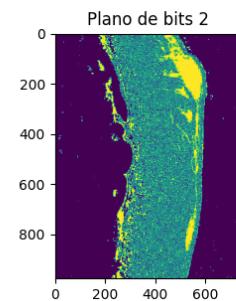
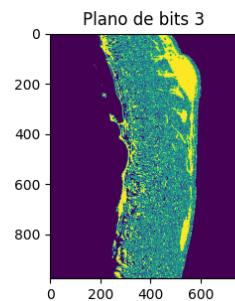
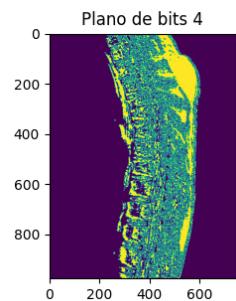
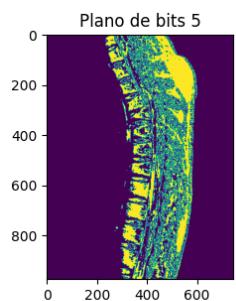
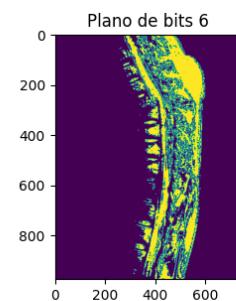
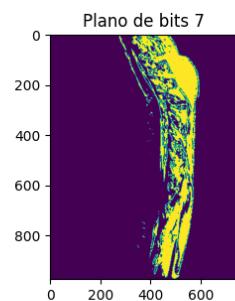
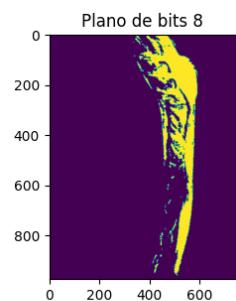


Plano de Bits

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/777092ce58863cfa535aa5733725028a4910ac69/Intensidade/planoBits.py>

Resultados:



Potência

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/777092ce58863cfa535aa5733725028a4910ac69/Intensidade/potencia.py>

Resultados:



Filtragem Espacial

A filtragem espacial envolve a manipulação de pixels em uma imagem com base em suas posições espaciais. A seguir será mostrado um exemplo feito manualmente e outro com o auxílio da biblioteca OpenCV.

1) Manual

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/FiltragemEspacial/filtragemEspacialManual.py>

Resultados:

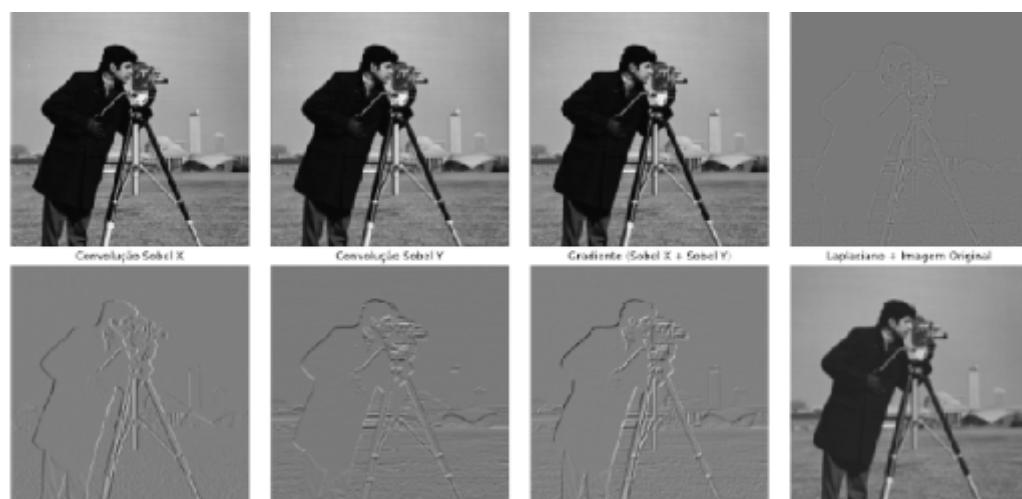
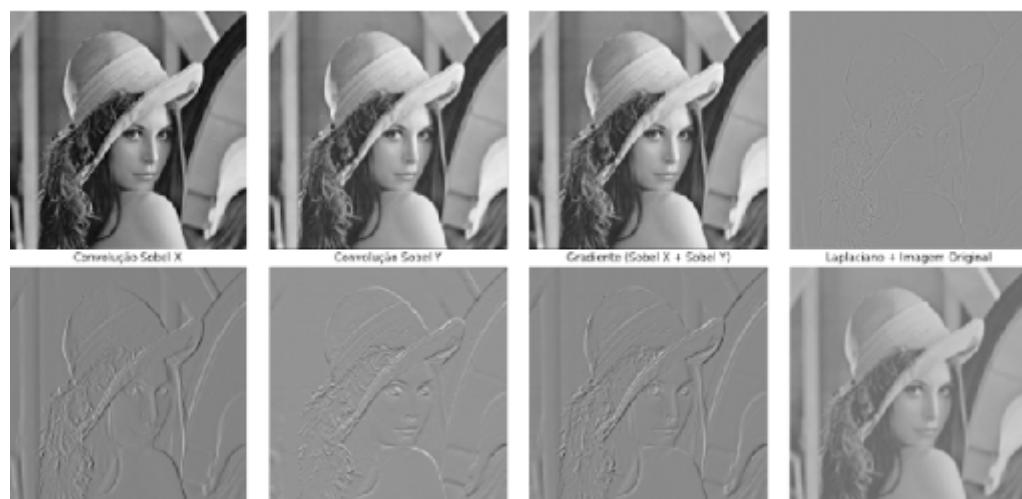


2) OpenCV

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/FiltragemEspacial/filtragemEspacialOpenCV.py>

Resultados:



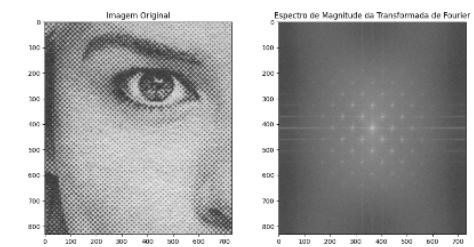
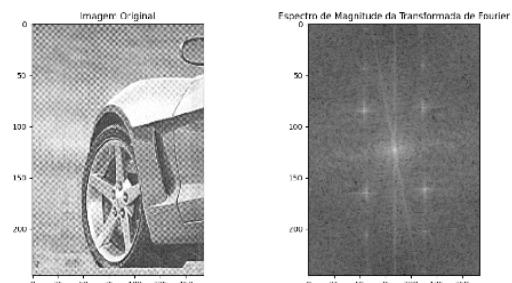
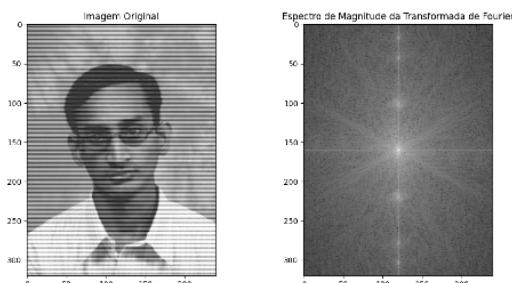
Fourier

A Transformada de Fourier é uma técnica poderosa em Processamento Digital de Imagens (PDI) para analisar e representar uma imagem no domínio da frequência, seu funcionamento se baseia na conversão de uma imagem do domínio espacial para o domínio da frequência, onde a informação é representada por suas componentes de frequência.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fourier/fourier.py>

Resultados:



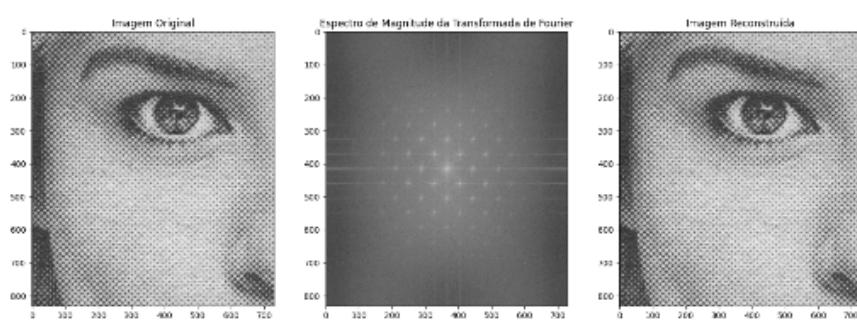
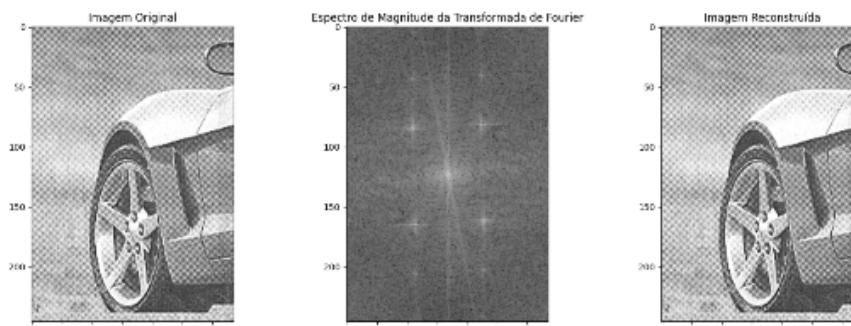
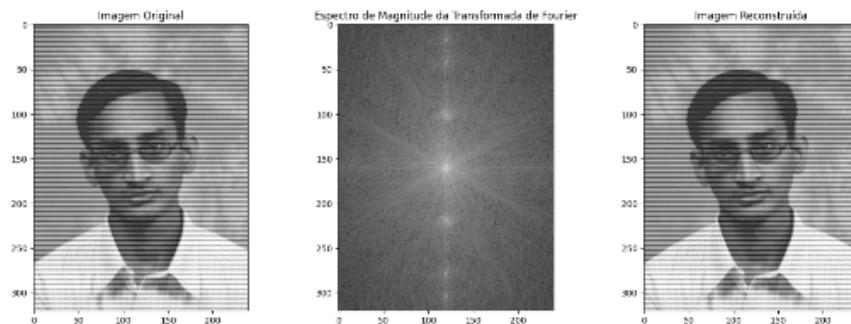
Transformada inversa de Fourier

Sua função é a oposta da transformada de Fourier, ou seja, ela pega uma imagem no domínio da frequência e converte para a imagem original

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fourier/fourierInversa.py>

Resultados:



Transformada de fourier mostrando os espectros de magnitude e fase

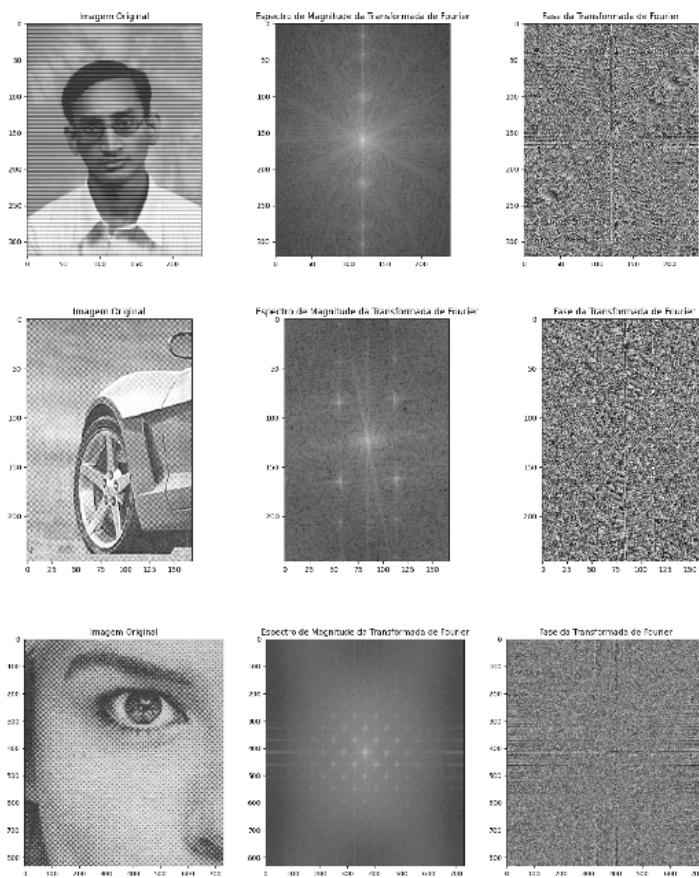
O espectro de magnitude exibe a magnitude (ou amplitude) das componentes de frequência da imagem no domínio da frequência. Em outras palavras, ele mostra quanta energia está presente em diferentes frequências.

O espectro de fase exibe a fase angular das componentes de frequência da imagem no domínio da frequência. A fase é a informação de deslocamento ou posição das ondas senoidais que compõem a imagem.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fourier/fourierMagnitudeFase.py>

Resultados:



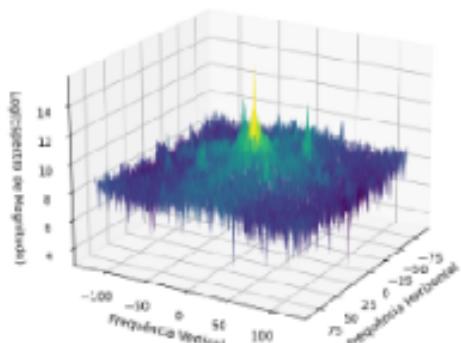
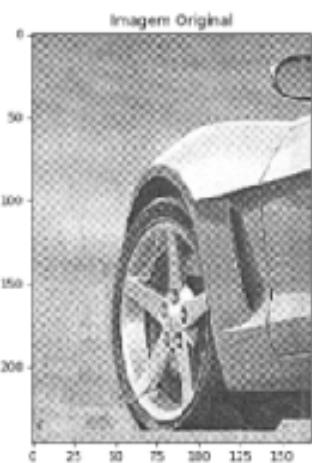
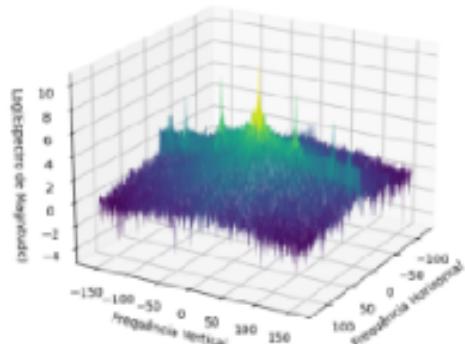
Plot 3D do espectro de Fourier

O plot 3D do espectro de Fourier é uma representação gráfica tridimensional das componentes de frequência de uma imagem no domínio da frequência. Essa representação pode ser útil para visualizar a distribuição de energia em diferentes frequências em uma imagem.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fourier/fourierPlot3D.py>

Resultados:



Sync em Fourier

A sincronização de frequência envolve a correspondência precisa de frequências, enquanto a sincronização de fase refere-se ao alinhamento das fases de dois sinais.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fourier/fourierSync.py>

Resultados:



Desafio de Stanford

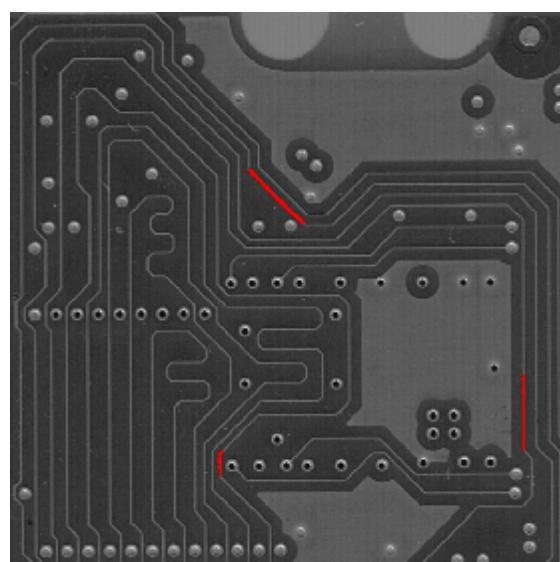
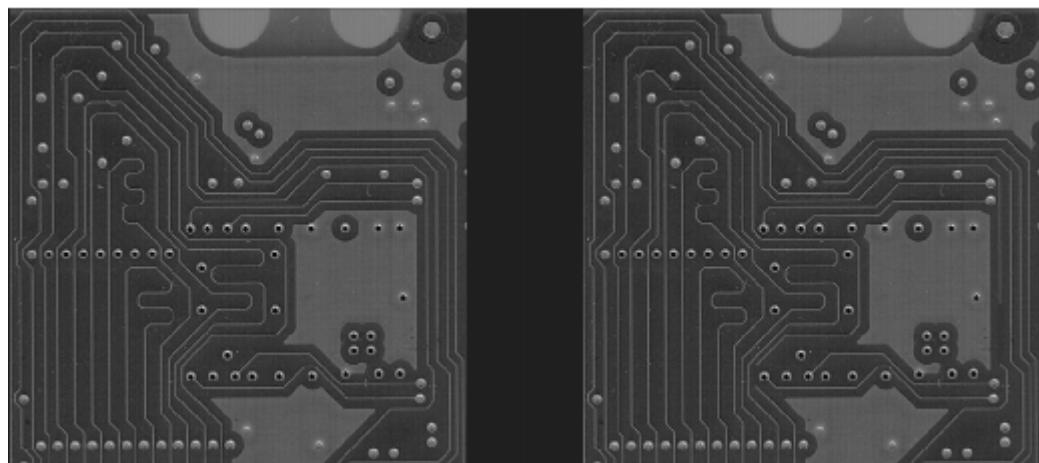
O principal propósito do desafio de Stanford é detectar alterações em vídeos em imagens, será mostrado ele aplicado em uma placa de circuito para detectar falhas e também para detectar movimentos em vídeos

1) Placa de circuito

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Stanford/stanfordPCB.py>

Resultados:

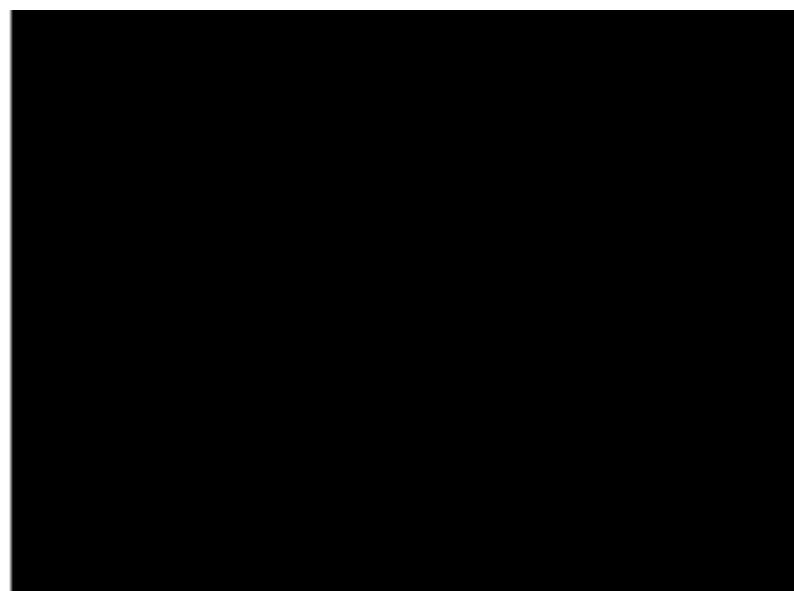


2) Vídeo

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Stanford/stanfordVideo.py>

Resultados:



Frequência

Consiste na transformação da imagem do domínio espacial para o domínio da frequência por meio da Transformada de Fourier. A ideia é analisar a imagem em termos de suas componentes de frequência e aplicar operações de filtragem neste domínio antes de retornar à imagem original por meio da Inversa da Transformada de Fourier.

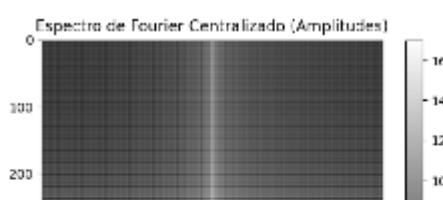
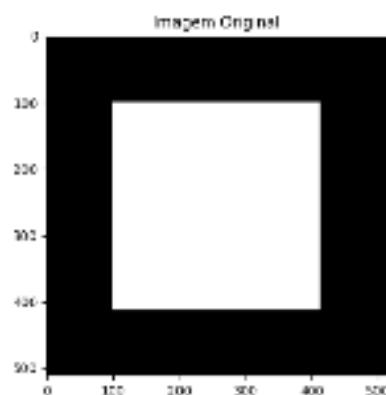
1) Manual

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Frequencia/frequencia.py>

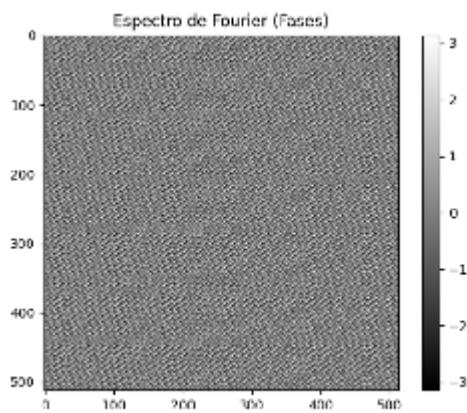
Resultados:

1) a)

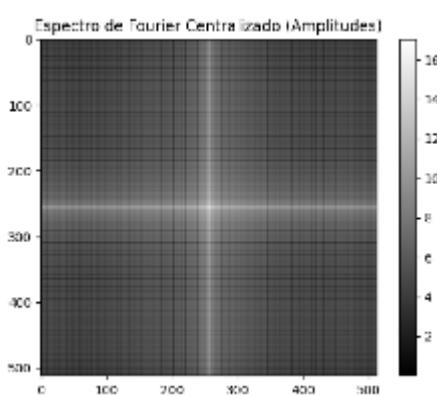


b)

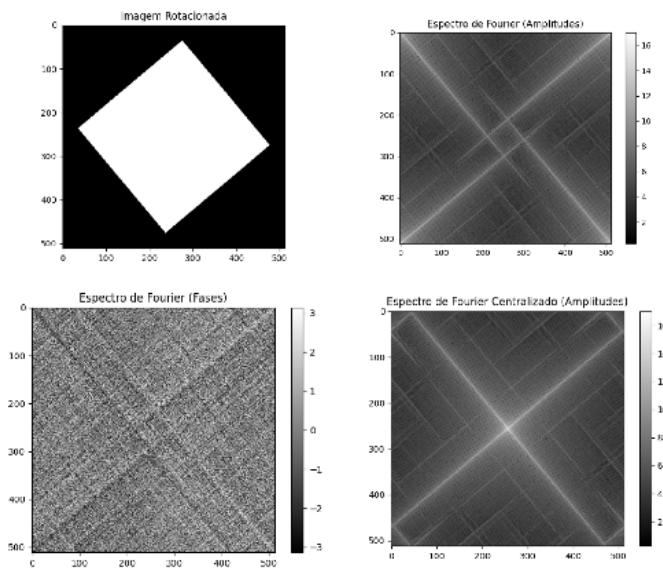
c)



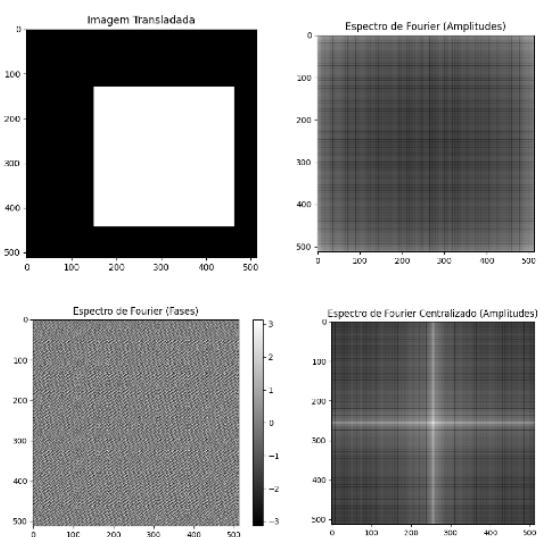
d)



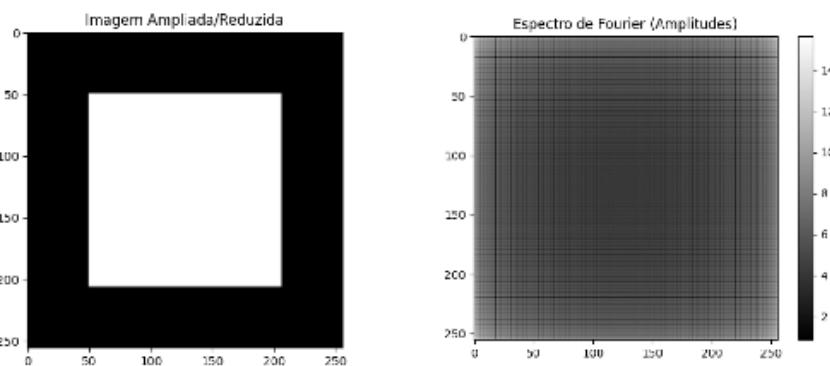
e)



f)



g)



Morfologia

Erosão e dilatação

A erosão e a dilatação são operações morfológicas básicas em processamento de imagem. Ambas operam em conjuntos de pixels para modificar ou realçar características específicas. Essas operações são comumente usadas em tarefas como remoção de ruído, realce de bordas, segmentação e análise de formas.

Erosão: reduz a região dos pixels brancos em uma imagem binária. O efeito da erosão é remover pequenos detalhes, afinar objetos e separar regiões conectadas.

Dilatação: é usada para aumentar a região dos pixels brancos. Ela tende a preencher lacunas e conectar regiões próximas. A operação também é realizada percorrendo a imagem com um elemento estruturante.

1) a)

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia1A.py>

Resultados:

EROSÃO



DILATAÇÃO



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

b)

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia1B.py>

Resultados:

EROSÃO



DILATAÇÃO



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

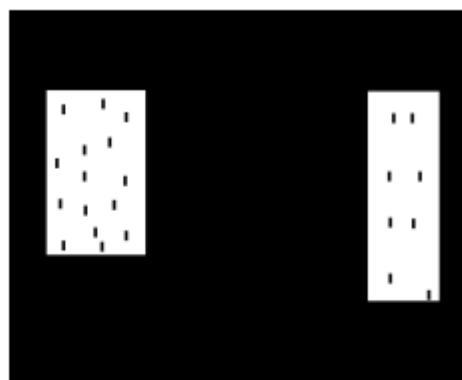
c)

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia1C.py>

Resultados:

EROSÃO



DILATAÇÃO



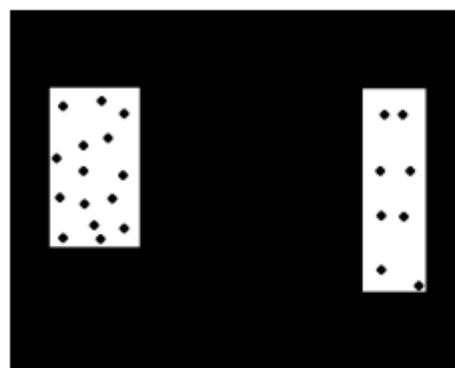
d)

Código:

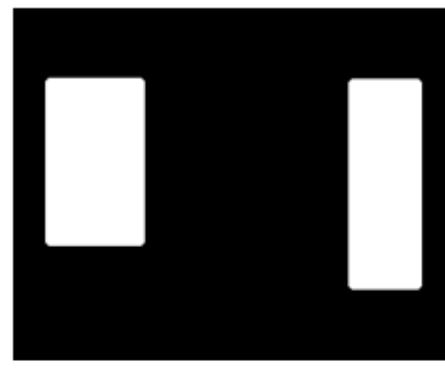
<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia1D.py>

Resultados:

EROSÃO



DILATAÇÃO



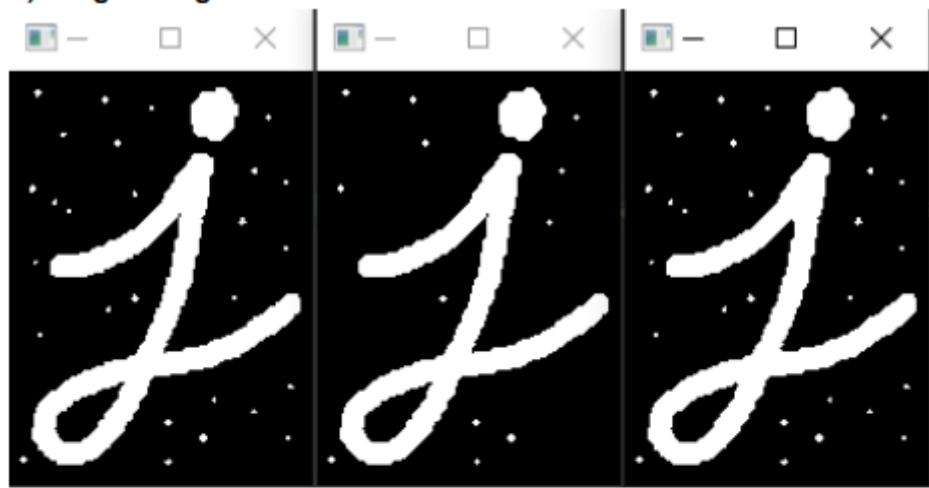
2)

Código:

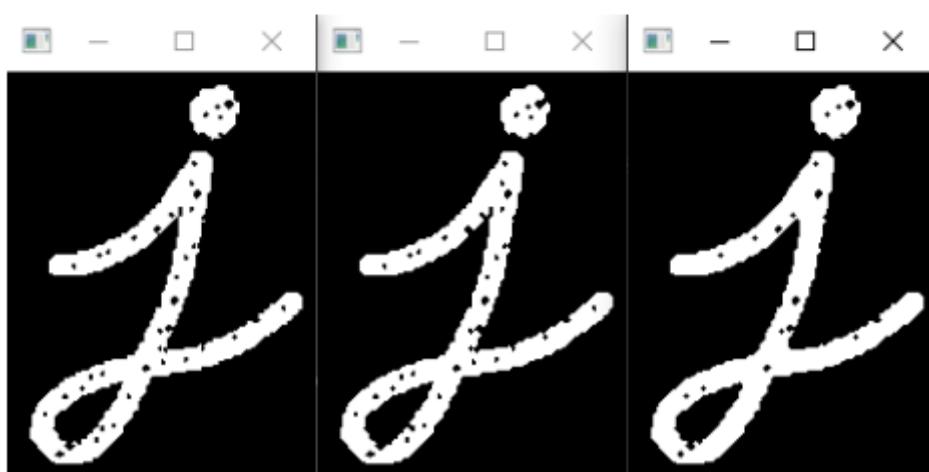
<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologiaAbrFecha.py>

Resultados:

B) Imagem original/aberta/fechada



C) Imagem original/aberta/fechada



3) Para se aproximar mais de um retângulo branco, seria necessário uma dilatação e fechamento.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia3.py>

4) Utilizar os métodos de dilação se mostrou eficiente

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia4.py>

3) Realizando a operação de erosão para reduzir o objeto branco à sua borda e subtraindo a imagem resultante da imagem binarizada original para obtemos apenas o contorno do objeto.

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Morfologia/morfologia5.py>