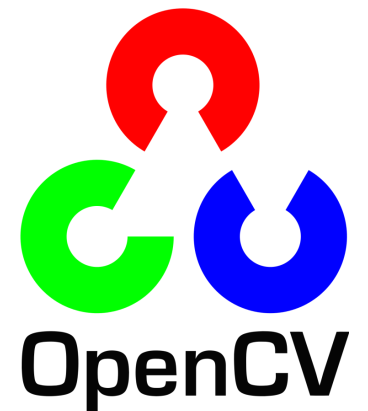
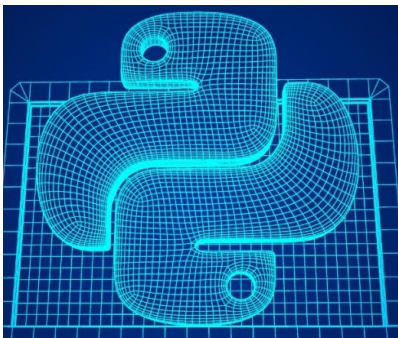


Processamento de Vídeo com Python e OpenCV

Prof. Maurício Acconcia Dias



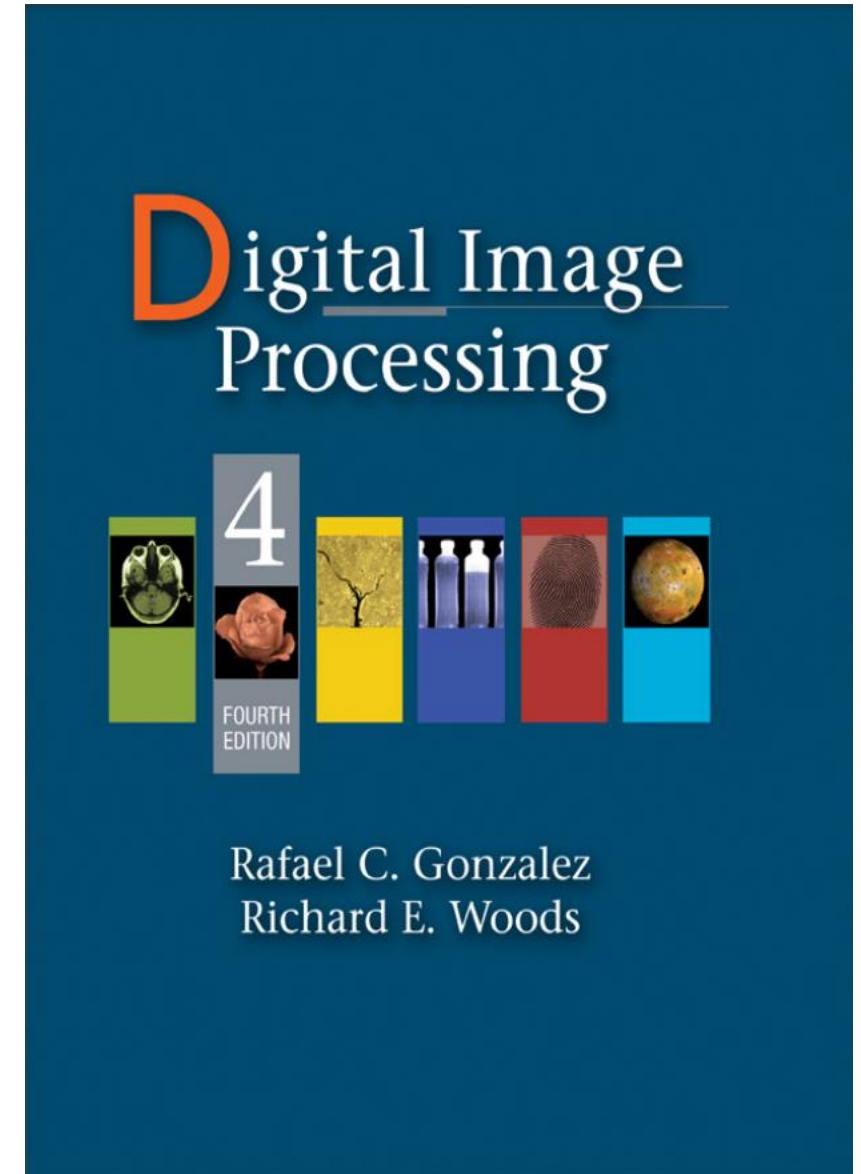
Agenda

- O que é o processamento de imagens
- Principais aplicações do processamento de imagens
- Básico da teoria do processamento de imagens
- A biblioteca OpenCV
- Qual o motivo de utilizarmos linguagem python?
- Processamento de vídeos – exemplo em python

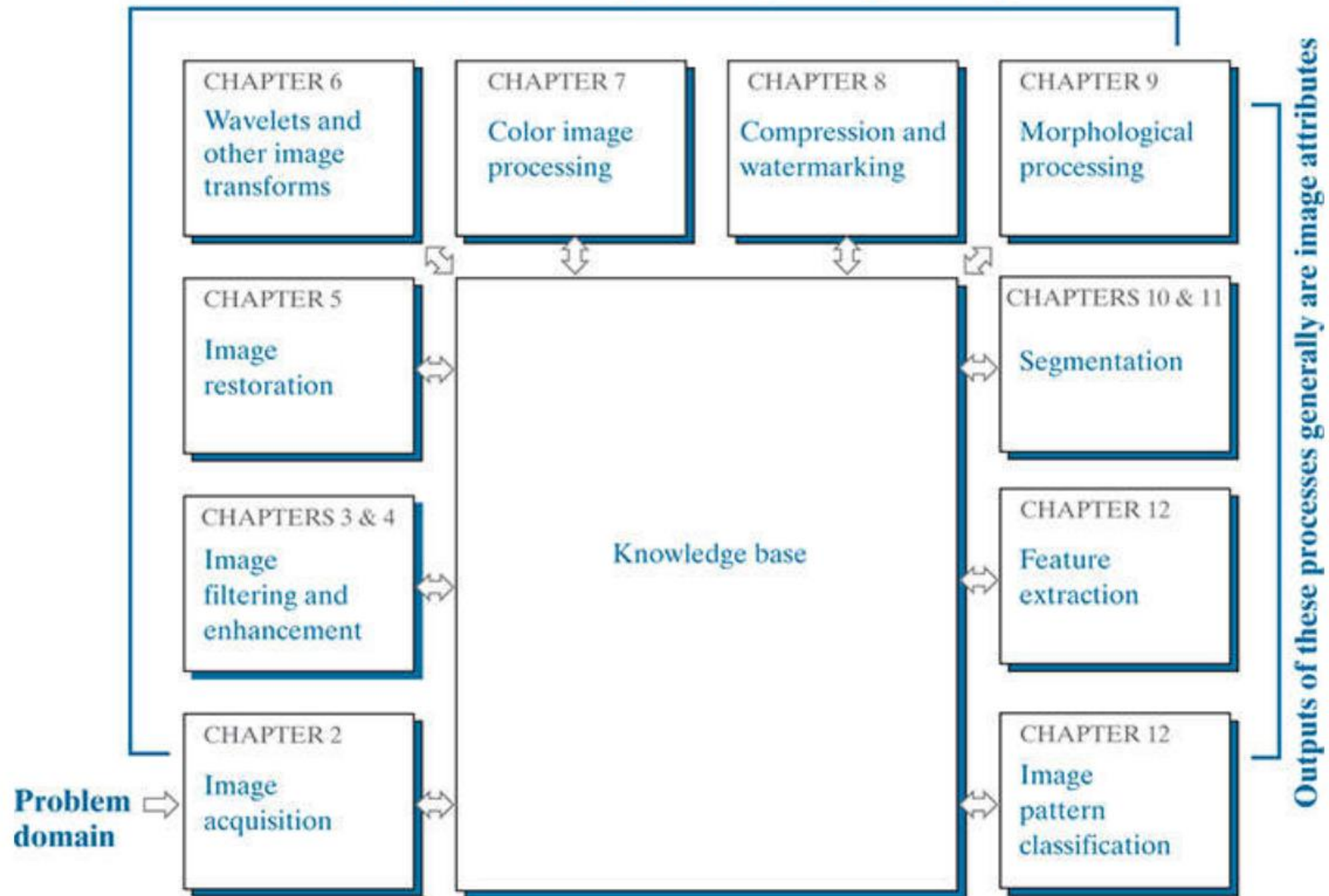
Processamento de imagens

- Principal referência da área (que foi utilizada como base para este conjunto de slides)

Gonzalez, Rafael C.; Woods, Richard E.
Digital Image Processing. 4th edition.
New York, NY. Pearson. 2018.



Outputs of these processes generally are images



Alguns exemplos...

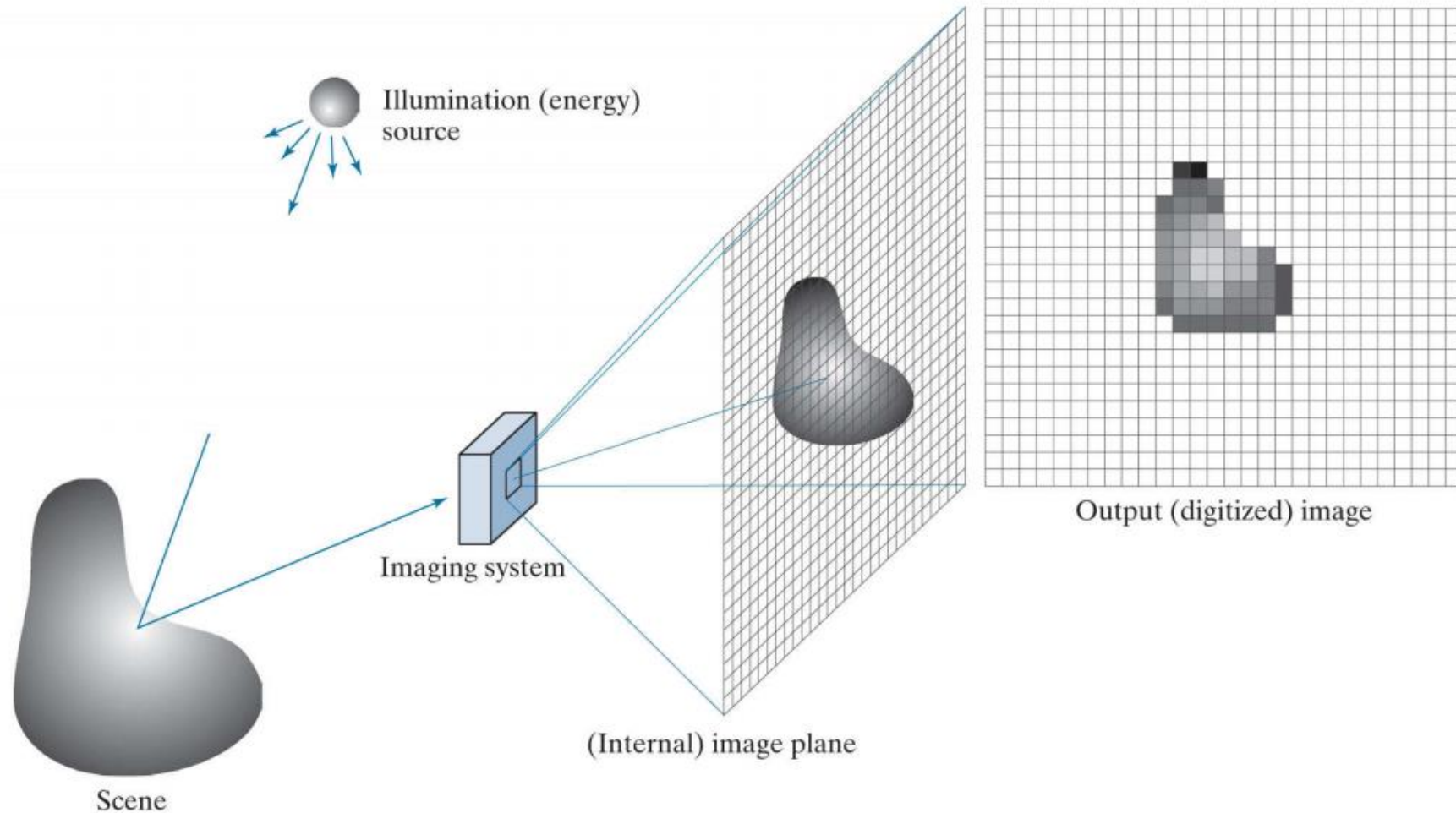


Aplicações de processamento de imagens

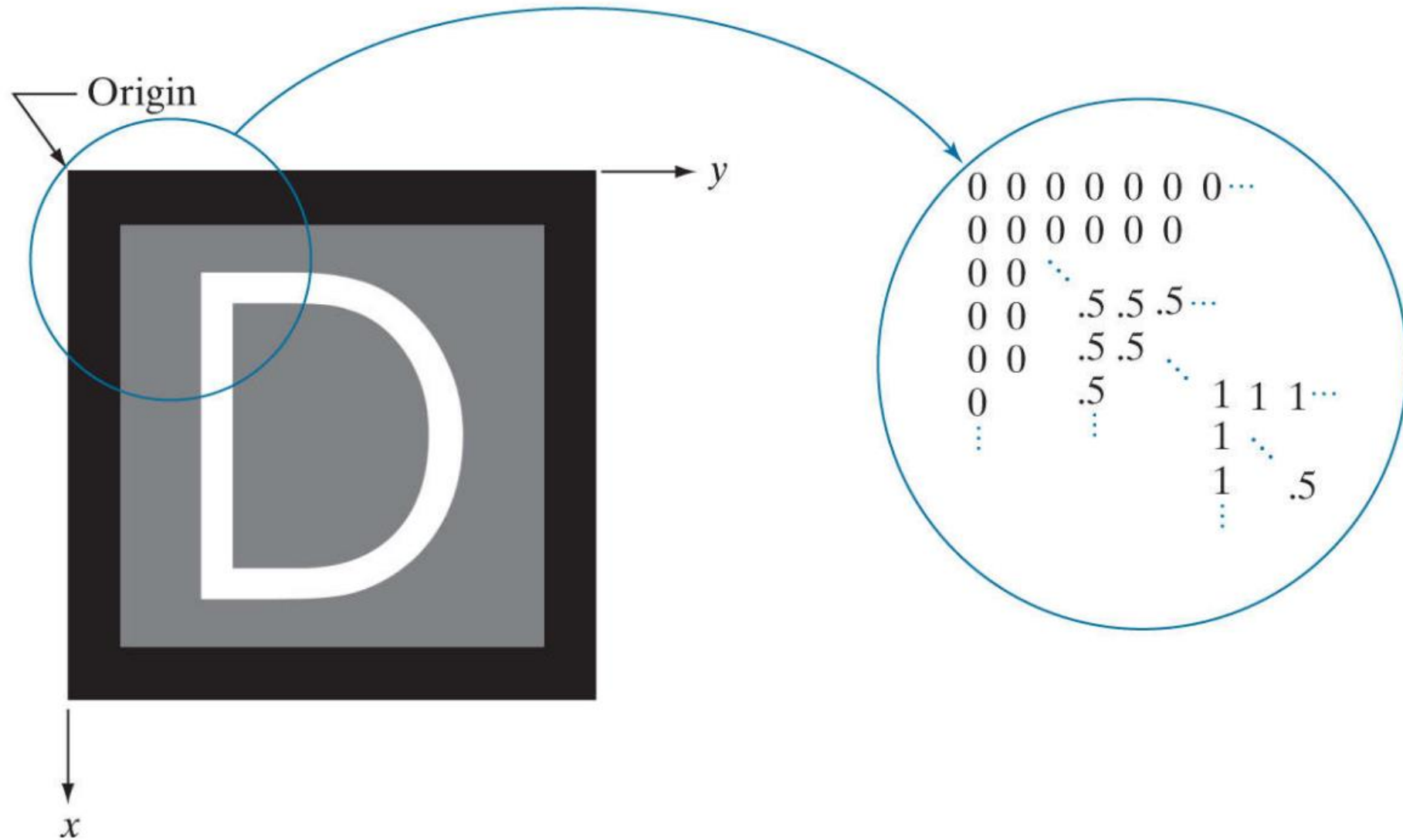
- Restauração de imagens
- Aplicações médicas
- Sensoriamento remoto
- Transmissão de imagens
- Visão computacional
- Robótica
- Criptografia
- Reconhecimento de padrões
- Imagens microscópicas



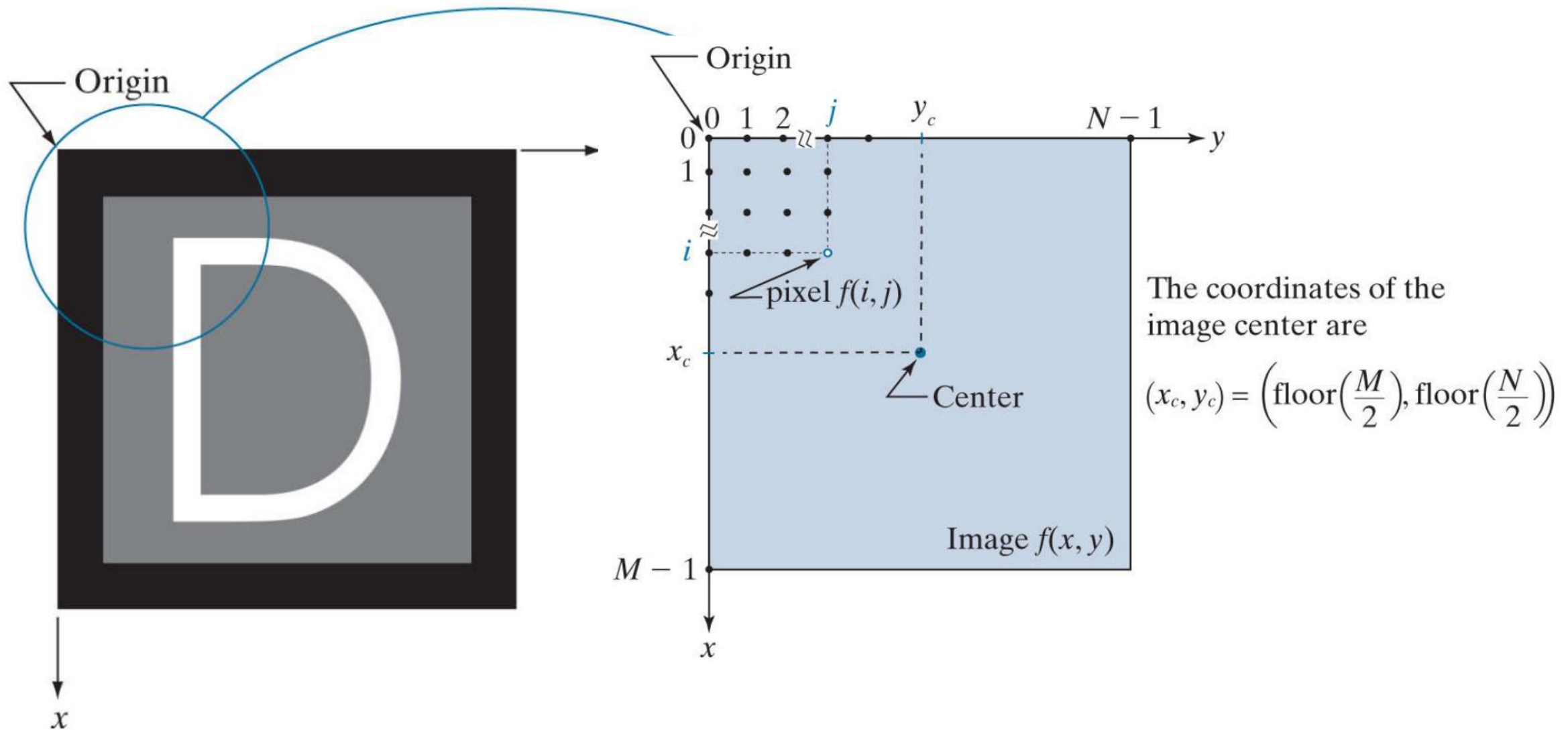
Processamento de Imagens



Representação Interna



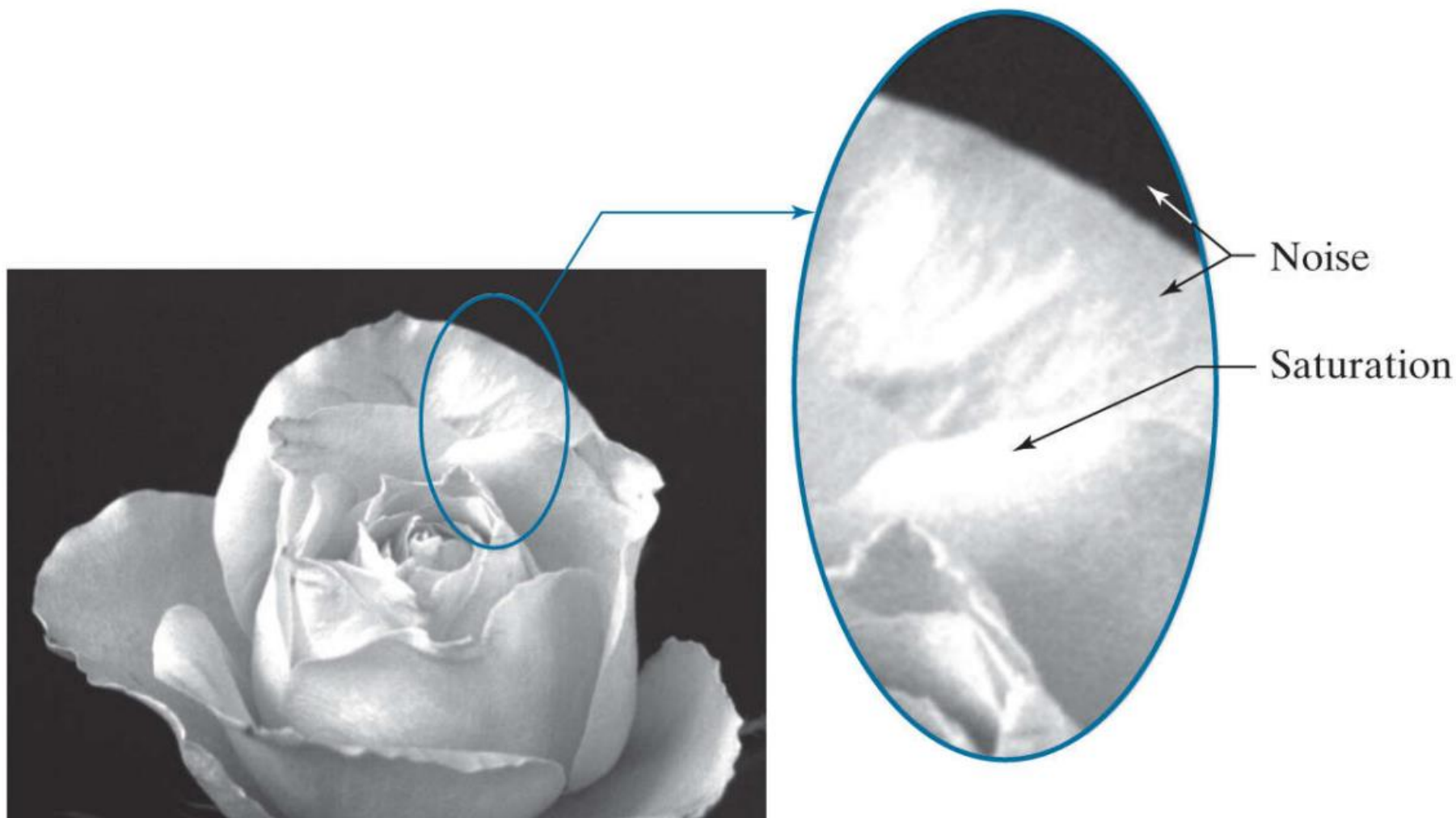
Representação Interna



Representação Interna

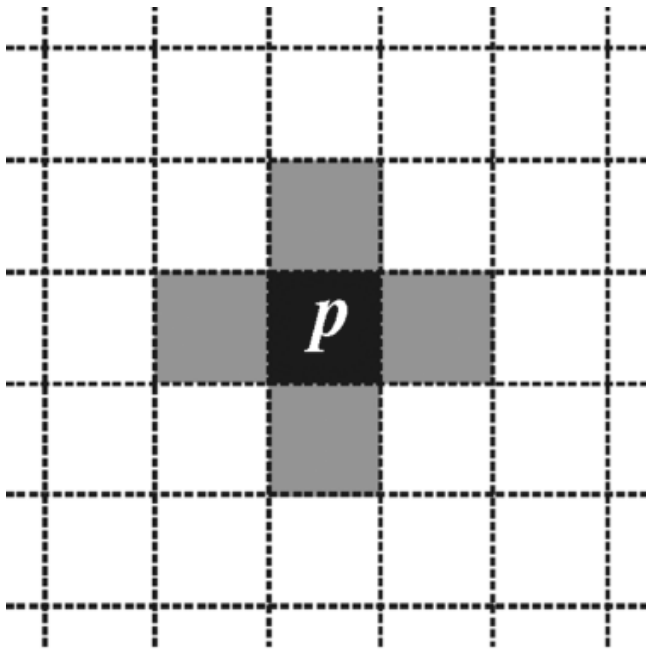
- Normalmente uma imagem é representada internamente em uma estrutura de dados
 - Um vetor ou uma matriz, estáticos ou dinâmicos.
 - Porém com o aumento do número de pixels é necessário otimizar a estrutura
 - Este é normalmente o motivo pelo qual aplicações que não utilizam bibliotecas gráficas não apresentam resultados tão bons

A captura de imagens apresenta problemas

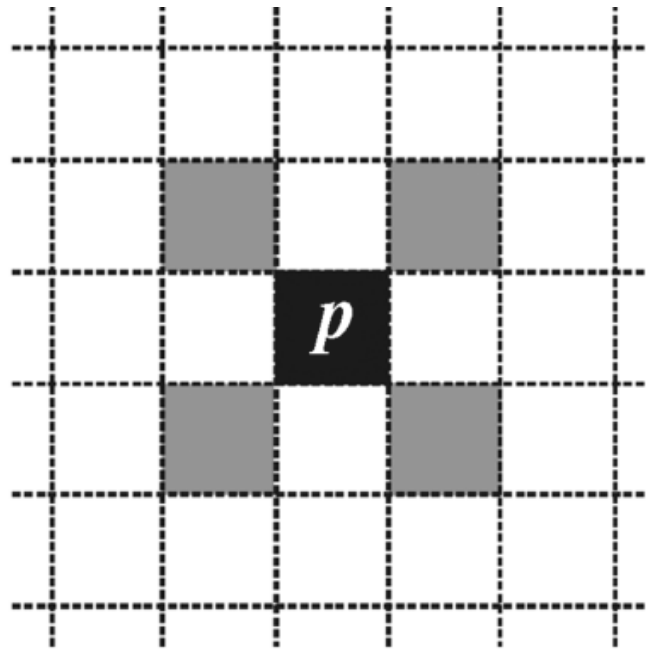


Para solucionar...

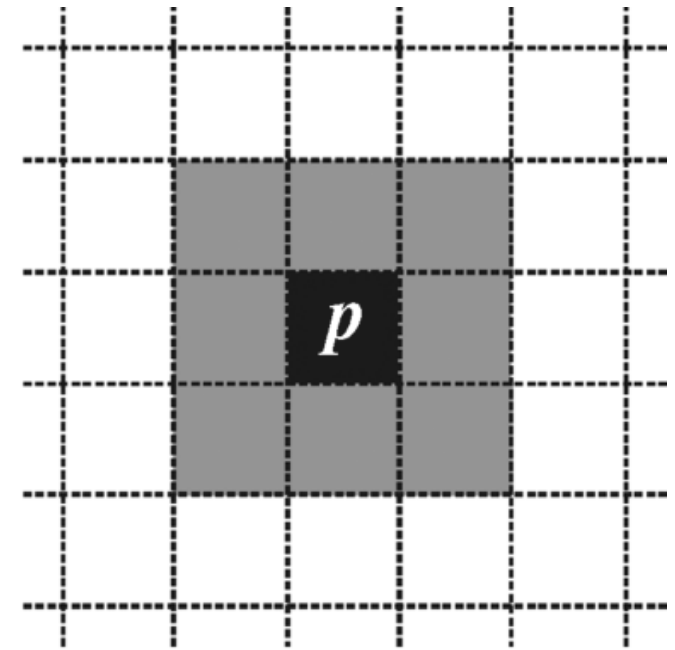
- Conceitos importantes
 - Vizinhança do pixel



(a)



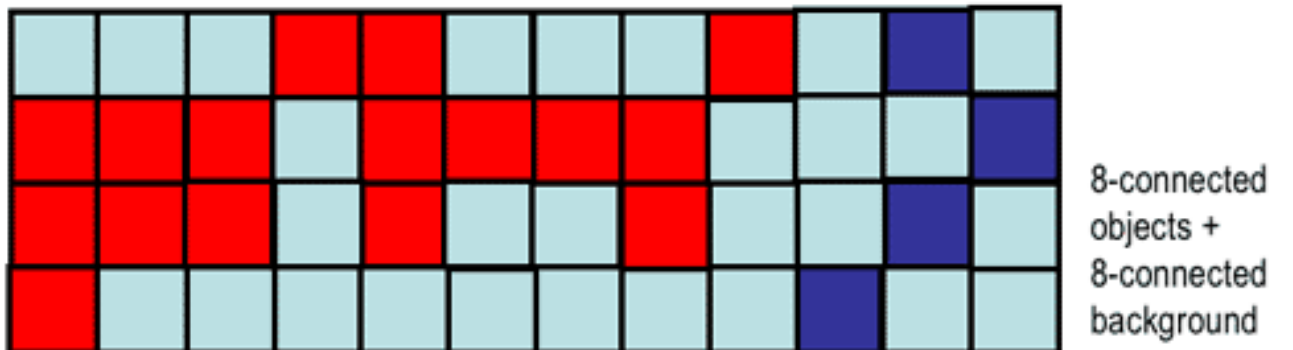
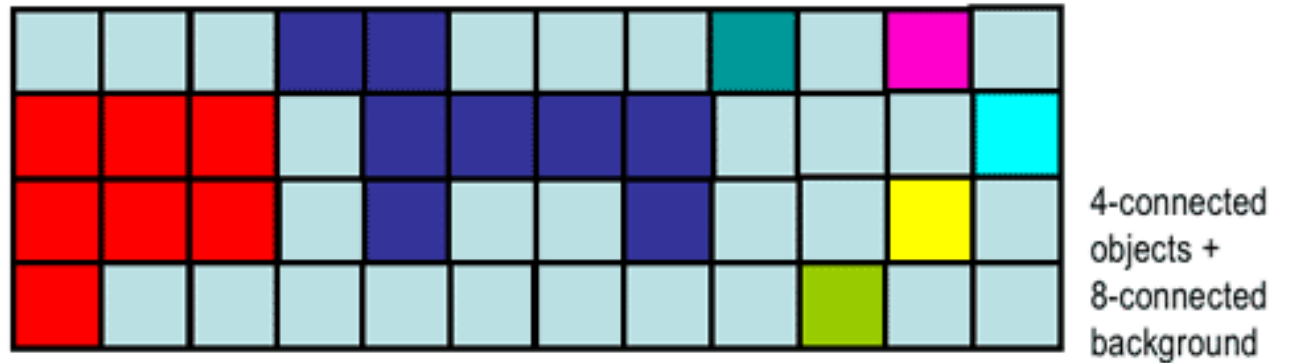
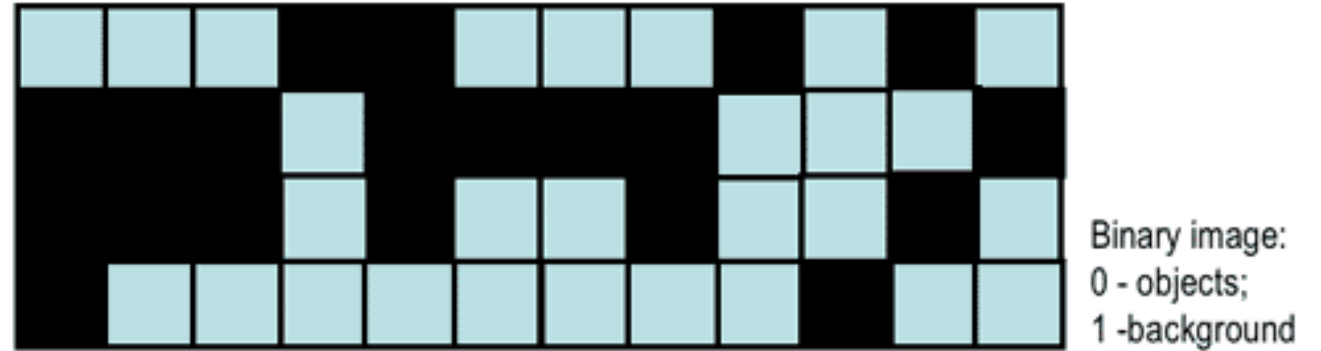
(b)



(c)

Para solucionar...

- O tipo de vizinhança define a distância entre objetos e também a forma como os pixels da imagem serão tratados



Para solucionar...

- Uma das técnicas mais utilizadas para melhorar a qualidade das imagens é a filtragem
- Existem diversos filtros que são utilizados como padrão, muitos deles seguem ideias de filtragem de sinais mais simples
 - Filtros passa-altas
 - Filtros passa-baixas
 - Filtros de média, mediana, gaussiano
 - Transformadas de haddamard, haar, Fourier

Para solucionar...

- Estes filtros são aplicados em imagens utilizando a operação de convolução

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

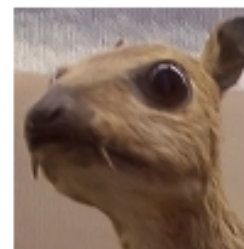
Image

4		

Convolved
Feature

Resultado da convolução

Input image







Convolution
Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map



<i>Original</i>	<i>Gaussian Blur</i>	<i>Sharpen</i>	<i>Edge Detection</i>
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

Então a imagem...

- Pode ser melhorada com relação a seus atributos
- Pode ser corrigida com relação à erros de captura
- Pode ser adaptada ao contexto necessário da aplicação
- Porém tudo isso envolve profundo conhecimento da área de processamento de imagens.

Imagens coloridas

- As imagens coloridas apresentam um problema mais complexo por possuírem componentes
- Inicialmente é preciso pensar que determinadas imagens coloridas são armazenadas de acordo com seu sistema de cores
- Alguns destes sistemas podem ser vistos no slide a seguir

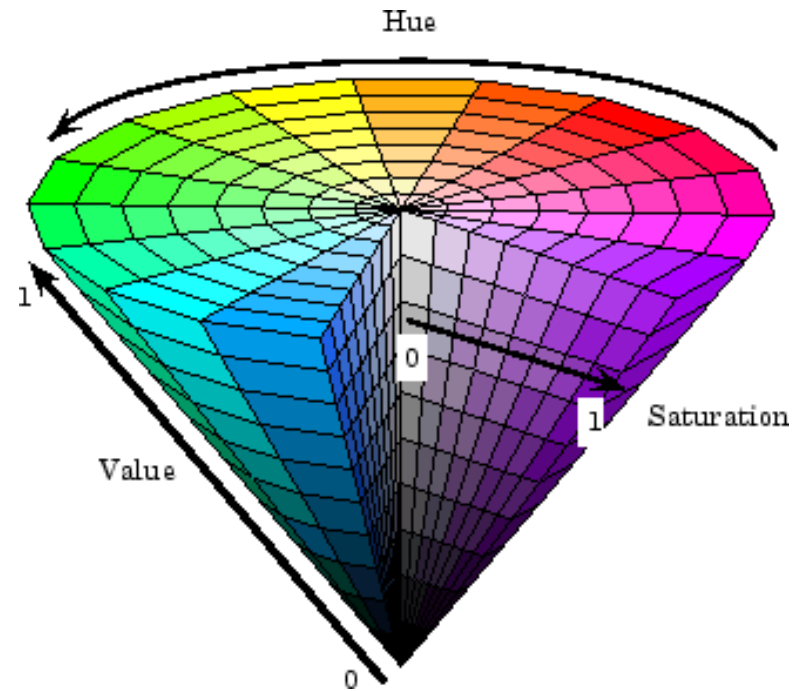
Sistemas de cores



CMYK - Subtractive Color



RGB - Additive Color



Sistemas de cores

- Nestes casos cada um dos sistemas possui um conjunto de informações a serem armazenadas
- Cada uma dessas informações é relativa a um pixel e deve ser armazenada na matriz de pixels correspondentes
- Além disso podem ser armazenadas informações de contraste, brilho, etc formando um conjunto de atributos para cada pixel da imagem

Imagens em geral

- Vários filtros obtêm diversas informações das imagens

Feature detector	Edge	Corner	Blob	Ridge
Canny[1]	Yes	No	No	No
Sobel	Yes	No	No	No
Harris & Stephens[2] / Plessey	Yes	Yes	No	No
SUSAN[3]	Yes	Yes	No	No
Shi & Tomasi[4]	No	Yes	No	No
Level curve curvature[5]	No	Yes	No	No
FAST[6]	No	Yes	Yes	No
Laplacian of Gaussian[5]	No	Yes	Yes	No
Difference of Gaussians ^{[7][8]}	No	Yes	Yes	No
Determinant of Hessian[5]	No	Yes	Yes	No
Hessian strength feature measures ^{[9][10]}	No	Yes	Yes	No
MSER[11]	No	No	Yes	No
Principal curvature ridges ^{[12][13][14]}	No	No	No	Yes
Grey-level blobs[15]	No	No	Yes	No

Imagens em geral

- Resultando em diversas análises



Perguntas relevantes

- Qual o número de pixels que devem ser processados?

- 720×480 : Widescreen DVD (anamorphic) (NTSC)
- 854×480 : EDTV (Enhanced Definition Television)
- 720×576 : D-VHS, DVD, miniDV, Digital8, Digital Betacam (PAL/SECAM)
- 720×576 : Widescreen DVD (anamorphic) (PAL/SECAM)
- 1280×720 : D-VHS, HD DVD, Blu-ray, HDV (miniDV)
- 1440×1080 : HDV (miniDV)
- 1920×1080 : HDV (miniDV), AVCHD, HD DVD, Blu-ray, HDCAM SR
- 1998×1080 : 2K Flat (1.85:1)
- 2048×1080 : 2K Digital Cinema
- 3840×2160 : 4K UHD TV, Ultra HD Blu-ray
- 4096×2160 : 4K Digital Cinema
- 7680×4320 : 8K UHD TV
- 15360×8640 : 16K Digital Cinema
- 61440×34560 : 64K Digital Cinema

2.123.366.400 pixels em 1 imagem

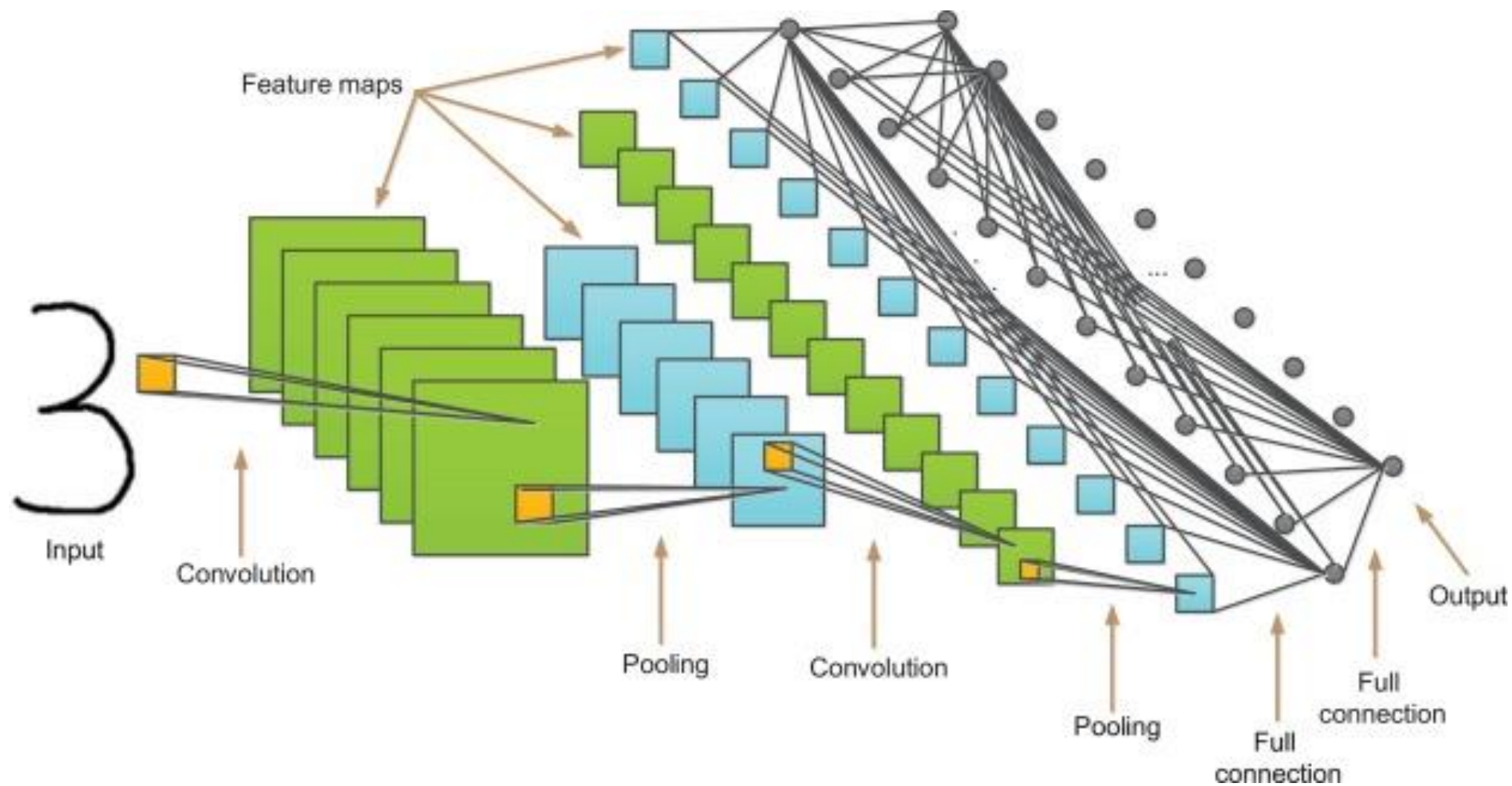


Como fazemos isso atualmente

- As técnicas de aprendizado profundo disponíveis atualmente ajudam no processo
 - Porém demoram dias, meses e anos para serem treinadas
 - Aquelas redes que vocês utilizam do google para gerar imagens na internet levaram meses/anos sendo treinadas antes de serem disponibilizadas para vocês

DNN

Milhões de neurônios e conexões!!!!

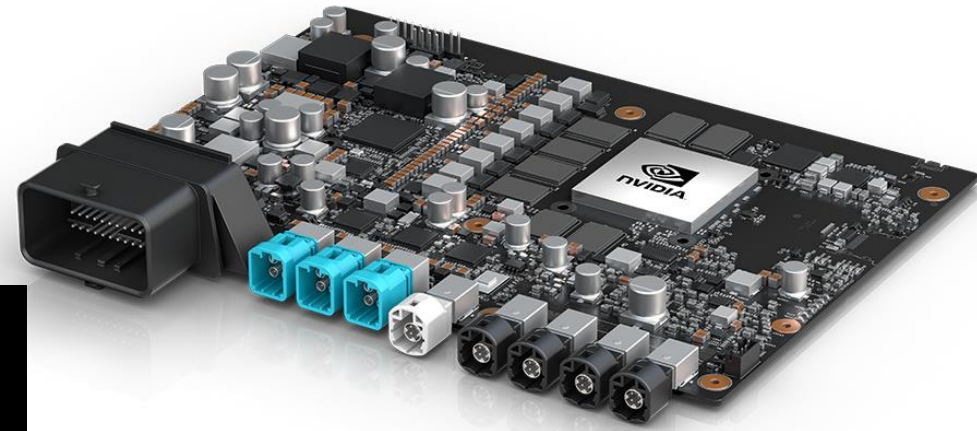


Certo e qual o problema?

- Aplicações de tempo-real que necessitam de processamento de imagens



Soluções atuais que prometem DL



NÃO FUNCIONAM, A MENOS QUE VOCÊ
TENHA UM DESENVOLVIMENTO ESPECÍFICO
PRA VOCÊ OK? TIPO A TESLA

Estamos ainda nas imagens

- Então, como fazemos com imagens em redes neurais e técnicas de IA que possam ser processadas em um veículo?
 - Dividimos as imagens em blocos
 - Processamos apenas algumas features não todas
 - Criamos hardware específico pra isso (tese do palestrante)
 - E mesmo assim não temos um veículo funcional, 100% confiável...
 - De qualquer forma os métodos podem ser aplicados assim.
 - Vamos entender uma imagem e aí passamos para o vídeo

O que ocorre num carro...



Certo, agora entendi a imagem...

- Só que uma imagem para um sistema destes não adianta nada certo?
- Você deve processar um vídeo... Mas afinal de contas o que é um vídeo?
- Um câmera de vídeo parte do princípio que se forem capturadas X imagens por segundo e projetadas na mesma velocidade seu olho e seu cérebro serão enganados com a ideia de movimento
 - E funciona...

Então um vídeo é?

- Um conjunto ENORMEEEEEE de imagens que devem ser processadas muito rapidamente
- O olho humano na sua melhor forma consegue ser enganado a partir de 50 fps (quadros por segundo que em inglês chamamos de *frames per second*).
- Para efeito de entendimento iremos considerar 64 ok?

Qual a meta de processamento

- Para que um veículo possa rodar autônomo ele tem que no mínimo ter o mesmo tempo de reação de um ser humano ok?
 - Este é o parâmetro mínimo utilizado por ser plausível e de certa forma atingível
- Isso faz com que um computador precise processar o dobro de quadros por segundo do que o olho humano processa (uma estimativa)
 - Ou seja, 128 fps
 - Mas estamos falando de processamento pesado em 2 bilhões de pixels por imagem...

Existe tecnologia pra isso hoje?

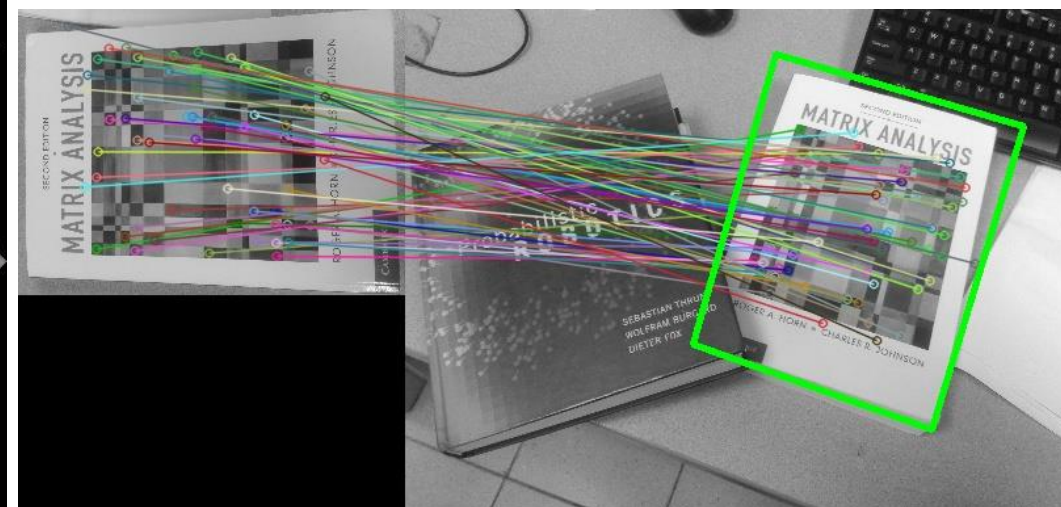
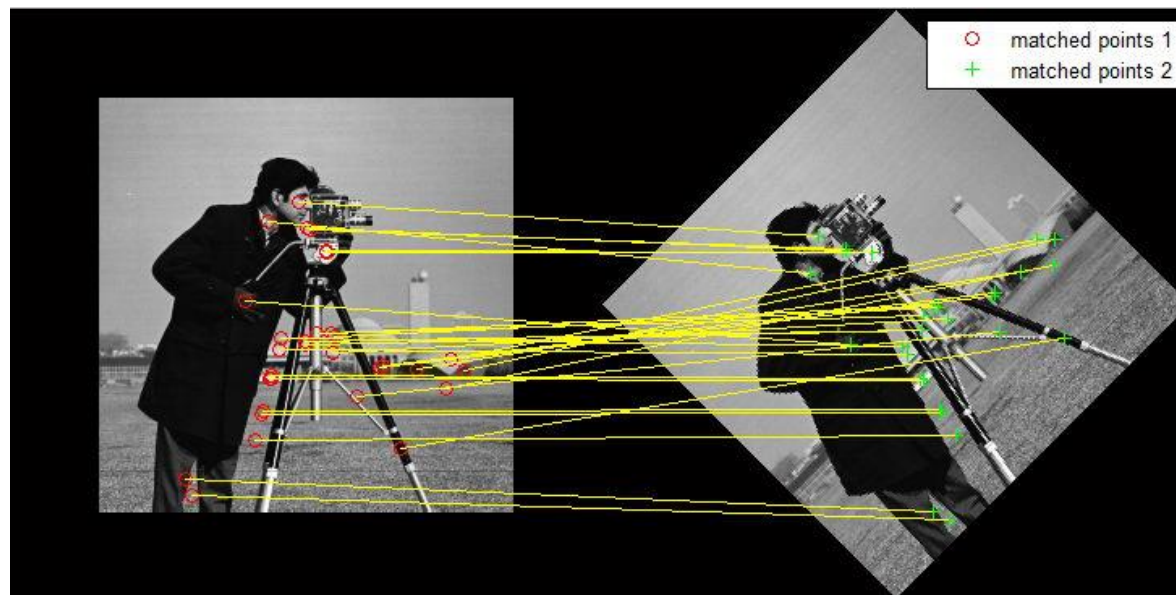
- Não existe, ao menos comercialmente
- Então como é feito
 - Fusão de várias câmeras em paralelo e processamento inteligente para compensar a incompetência sensorial
 - A tesla (quase) consegue fazer se a rua não tiver buracos, tiver sinalização, se as pessoas não correrem muito com o carro. Enfim, roda lá nos EUA e nunca rodaria aqui no Brasil...
- Já viram o tamanho do problema?
 - Por isso temos um grupo de pesquisa na FHO o SIVA que tenta solucionar isso de algumas formas.

Ok!!! Voltando para a palestra

- Agora que temos o ferramental vamos entender a palestra
- Processamento de vídeo usando OpenCV e Python
- Vamos entender como é feito o processamento de vídeo com algumas operações simples usando o OpenCV e a linguagem Python

O que podemos fazer em vídeos?

- Identificar coisas/pessoas
- Ver o que mudou entre imagens (mestrado do palestrante)
- Fazer o tracking de objetos, ou seja, achar algo na imagem e seguir enquanto o objeto estiver na imagem
- Identificar pontos de referência em uma imagem e seguir os pontos de referência
- Isso tudo permite várias maneiras de dirigir veículos por imagem!
 - Correlação de imagens, SURF, SIFT, Follow-me, etc..



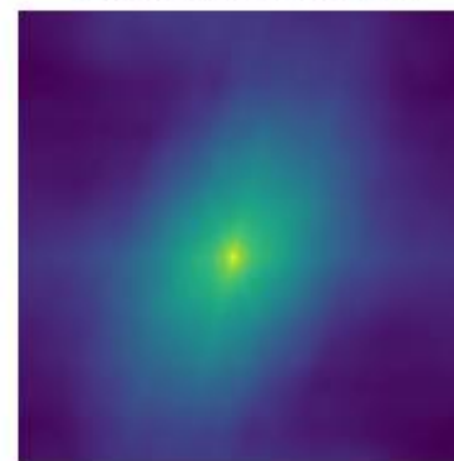
Reference image

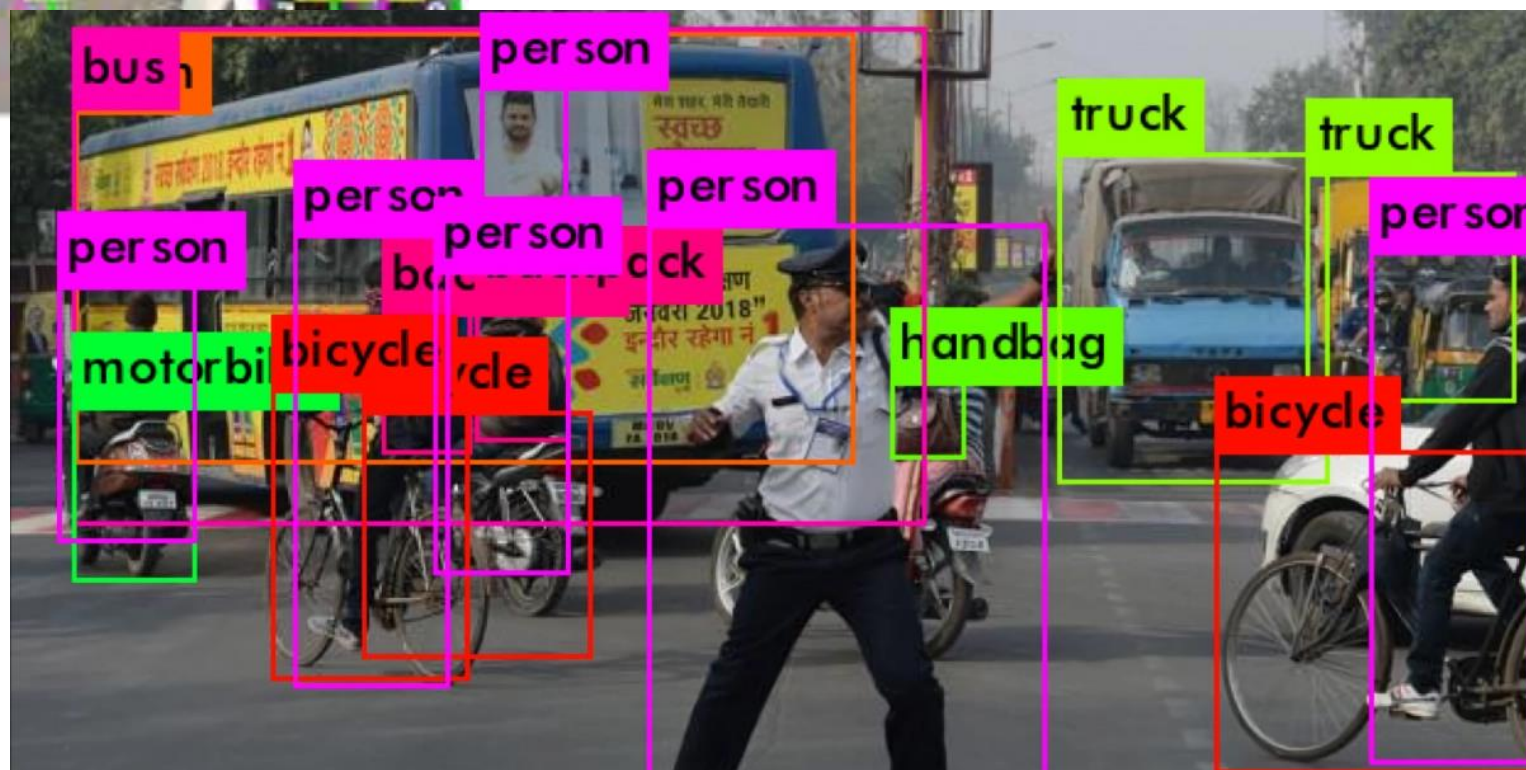


Offset image



Cross-correlation





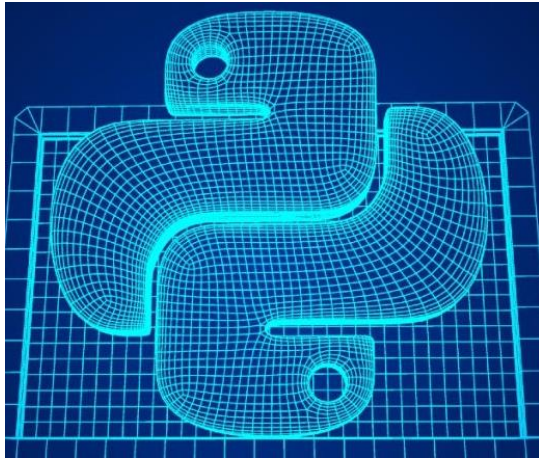
O carro da tesla....



Processamento de vídeo

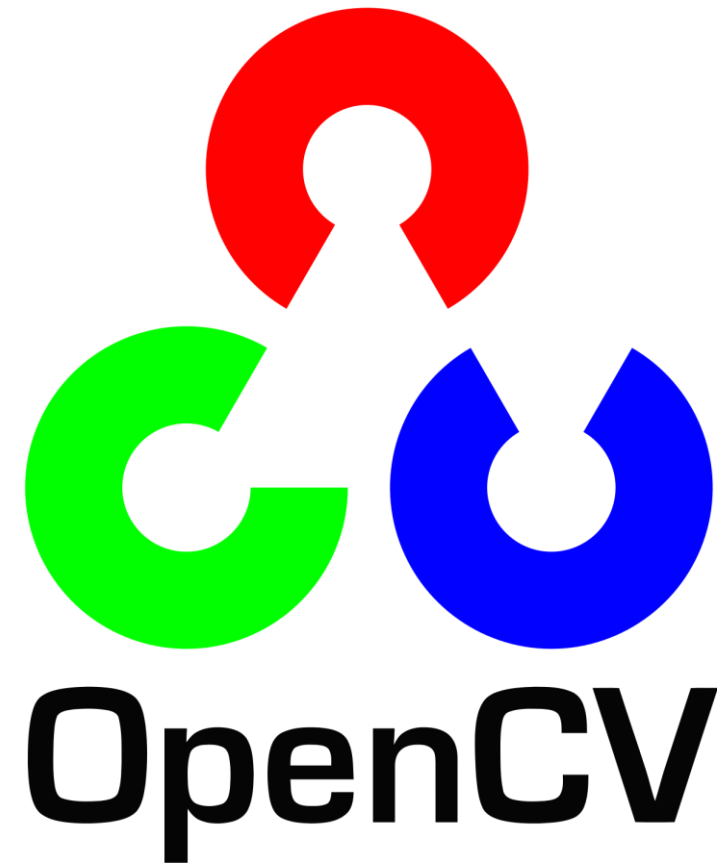
- Este é então o que podemos considerar o estado da arte
- Certo mas eu quero fazer isso em casa prof, como faço?
- Você deve utilizar as ferramentas corretas e iniciar com as expectativas bem baixas!

Linguagem e ambiente



Biblioteca de processamento

- Biblioteca de processamento de imagens
- Aberta
- Constantemente atualizada
- Com ferramentas de imagem e vídeo
- C/C++/Python
- Rápida e eficiente
- Possui algoritmos de IA internos
- Integrável com outras soluções em Python



Como proceder então

- Muita documentação online – M U I T A
- Este minicurso será um teaser e só o começo da sua jornada
- Muito do que você precisa/quer vai ter que ser aprendido por conta
- Alguns passos podem te ajudar muito, e o principal é entender o que você tem que fazer em cada caso
- Vamos analisar o nosso:

O que precisamos fazer:

- Escolher um vídeo de algo que queremos processar
- Descobrir como carregar este vídeo para o OpenCV (na linguagem escolhida e isso envolve a instalação)
- Descobrir como aplicar algoritmos e filtros em vídeos em OpenCV que já estão prontos
- Iniciar o desenvolvimento de novos filtros e aplicações (esse não hoje)

Para você que esta apenas acompanhando

- Para reproduzir estes códigos você precisa:
 - Baixar o ANACONDA
 - Instalar o ANACONDA
- 5 dias depois...
 - Instalar o OpenCV – no prompt do anaconda pip install opencv
 - Sério é imbecil assim mesmo.
 - Mas professor quero C/C++ como fazer?
 - Vai pro Linux
 - “Mas eu tenho Windows...” -> D E S I S T A
 - Mas que queria muito
 - Segue o link e boa sorte porque eu não vou me meter com essas coisas
 - <https://towardsdatascience.com/install-and-configure-opencv-4-2-0-in-windows-10-vc-d132c52063a1>

Pausa para um café

- Pessoal vamos pausar para um café e voltamos com a codificação ok?
- Disponibilizarei o código pra vocês então quem quiser pode seguir comigo.

Obrigado!

Prof. Maurício Acconcia Dias

