

# **VISÃO COMPUTACIONAL**

**Webconferência**

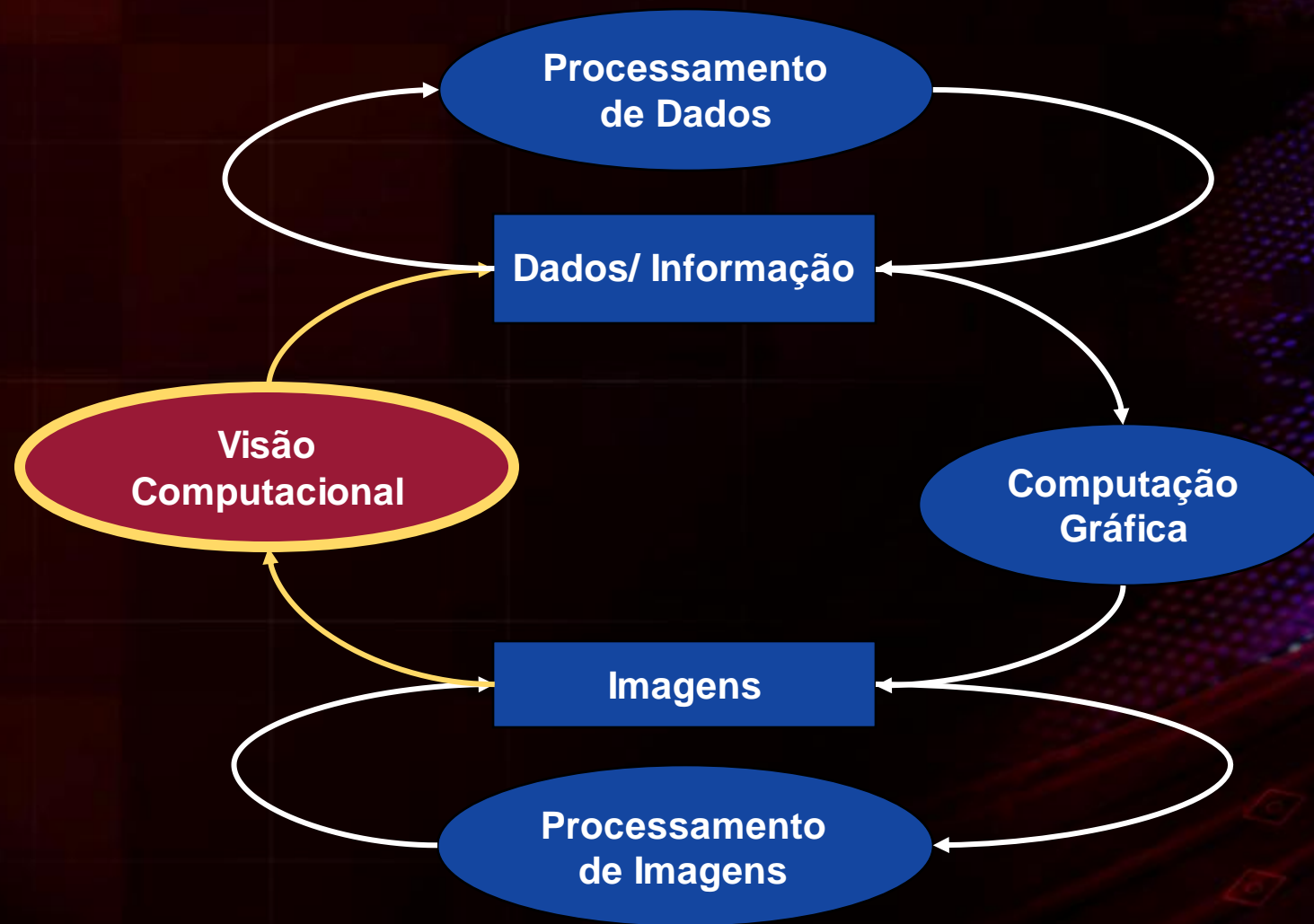
**Revisão**

**Prof. Márcio P. Basgalupp**

# Semanas

1. **Introdução à Visão Computacional**
2. **Pré-processamento de imagens**
3. **Imagens coloridas e câmeras**
4. **Aprendizado de Máquina**
5. **Segmentação de imagens**
6. **Classificação de imagens**
7. **Reconhecimento de objetos**

# VISÃO COMPUTACIONAL



# PROCESSAMENTO DE IMAGENS

## Processamento (baixo nível)

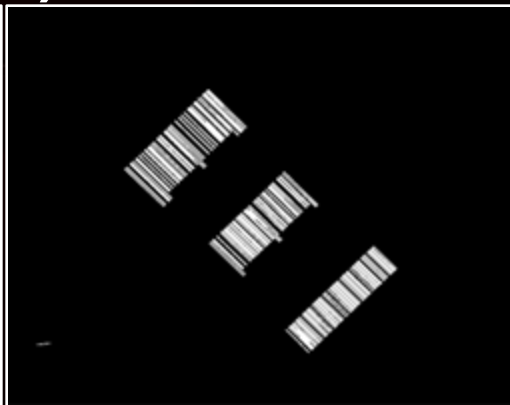
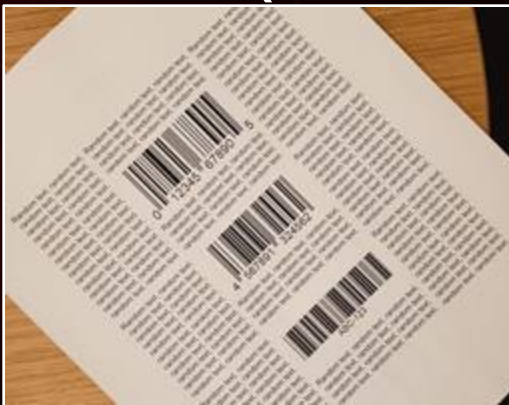


↑  
**ruídos**



↑  
**contraste**

## Análise (alto nível):



**Segmentação**



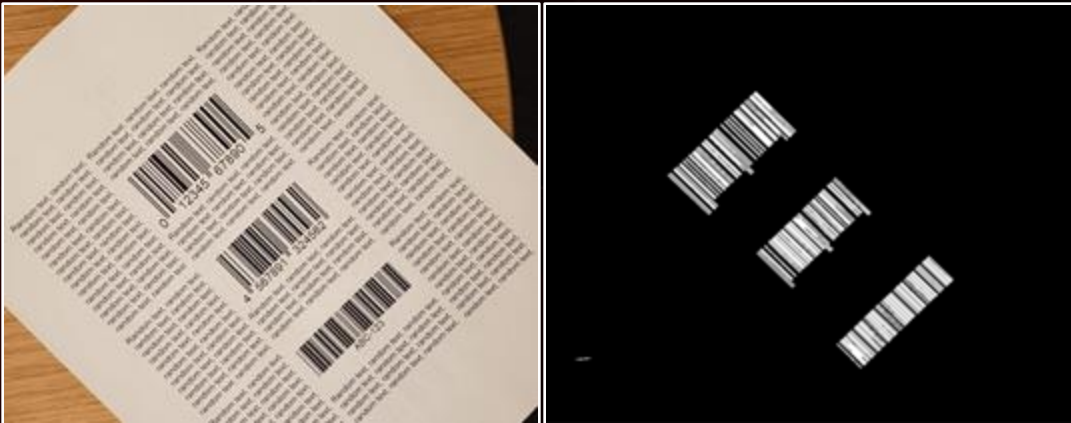
**Classificação**



# PROCESSAMENTO DE IMAGENS

**Análise de imagens (alto nível): segmentação em regiões de interesse, descrição dessas regiões para representar o conteúdo da imagem, e reconhecimento ou classificação.**

**Segmentação**



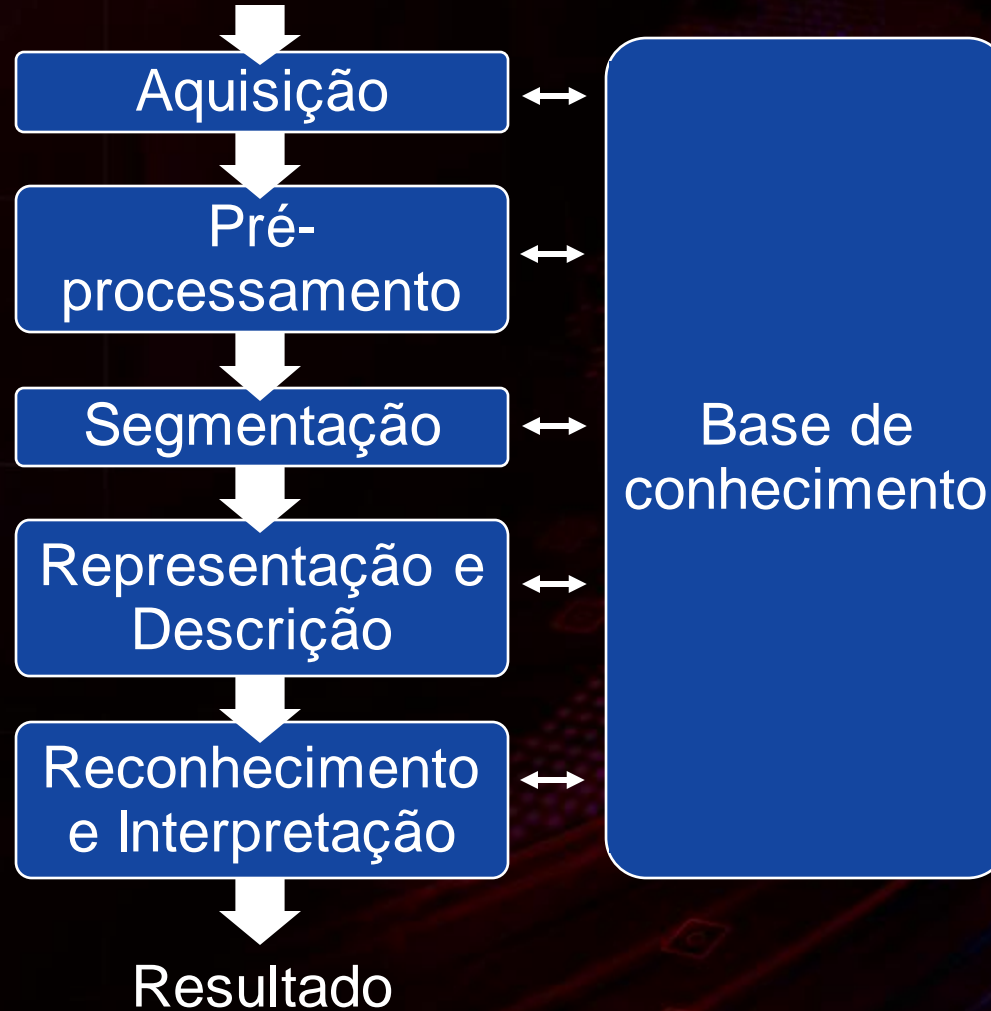
**Classificação**



# ETAPAS

## Etapas de um sistema de processamento de imagens

domínio do problema



# ETAPAS

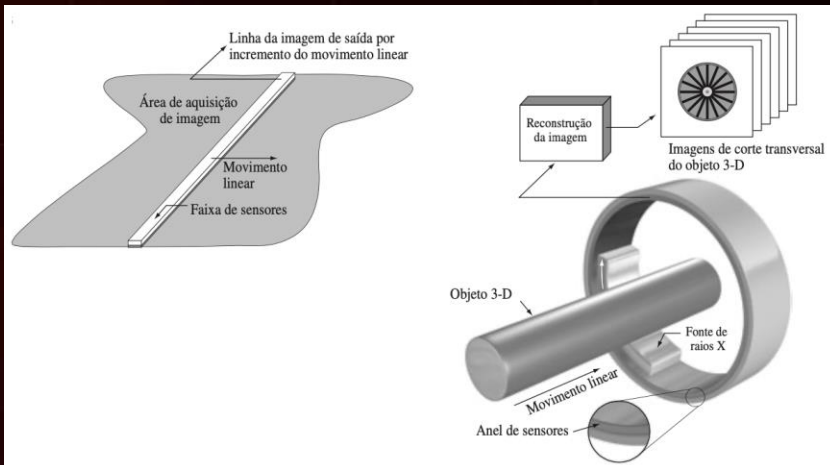
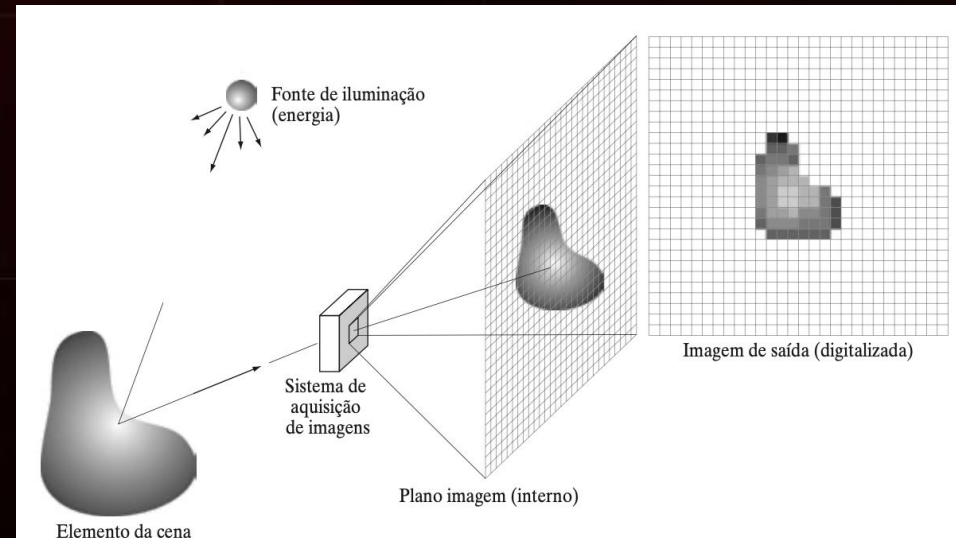
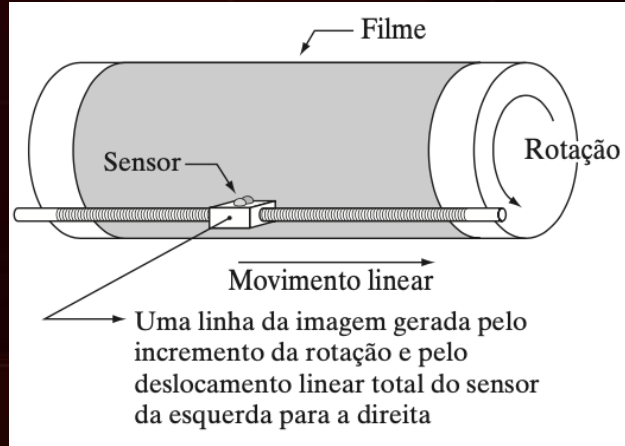
**Aquisição:** captura a imagem por um dispositivo ou sensor e converte essa imagem para uma representação adequada para o seu processamento.

domínio do problema





# AQUISIÇÃO DE IMAGENS – SENSORES





# AMOSTRAGEM E QUANTIZAÇÃO

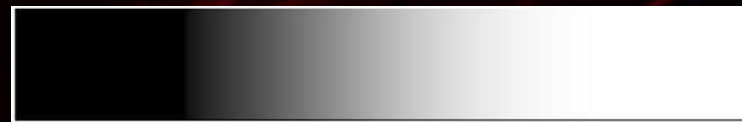
UMA IMAGEM DIGITAL PODE SER OBTIDA POR UM PROCESSO DENOMINADO DIGITALIZAÇÃO, QUE ENVOLVE OS PASSOS DE:

**Amostragem:** discretiza o domínio de definição da imagem nas direções  $x$  e  $y$ ;

**Quantização:** escolhe o número inteiro  $L$  de níveis de cinza permitidos para cada ponto da imagem.

Pixel

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(1,0) & \dots & f(M-1,0) \\ f(0,1) & f(1,1) & \dots & f(M-1,1) \\ \vdots & \vdots & \ddots & \vdots \\ f(0,N-1) & f(1,N-1) & \dots & f(M-1,N-1) \end{bmatrix}$$



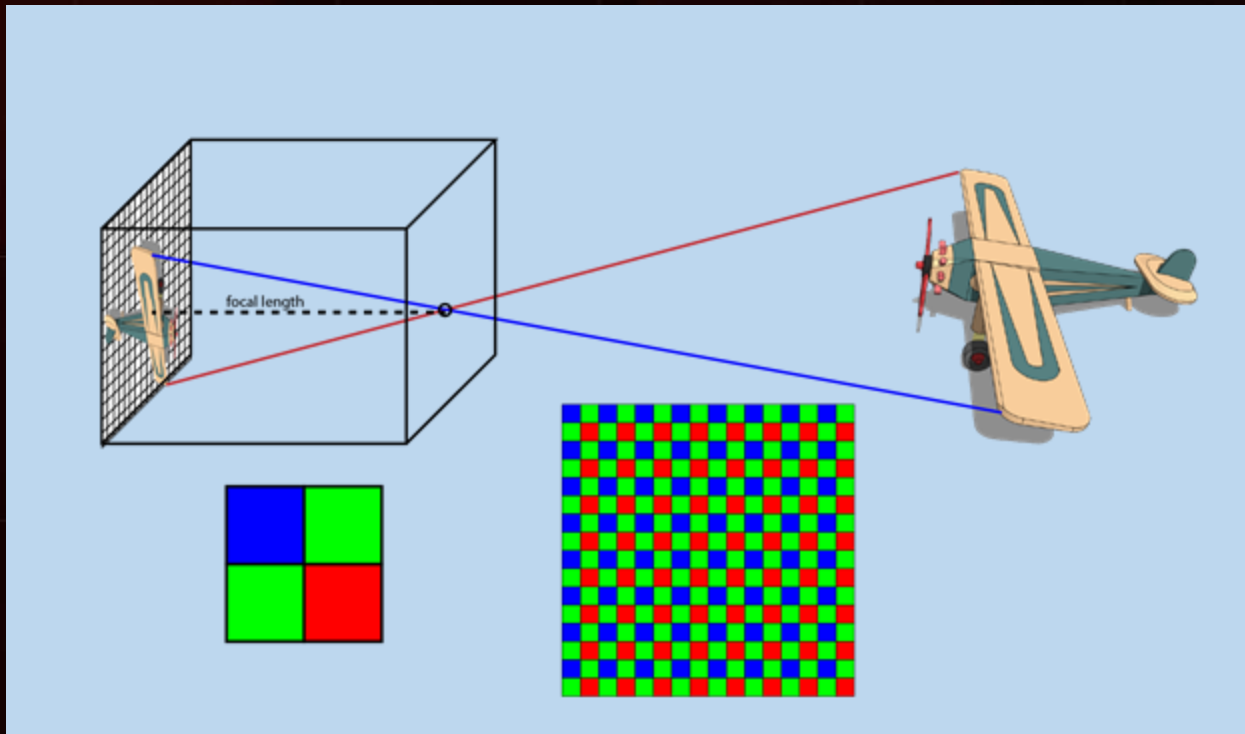
# TAMANHO DE IMAGENS

Quantidade de bits para armazenamento de imagens  $N \times N$  com  $L$  níveis de cinza e  $k$  bits.

$N/k$	1 ( $L = 2$ )	2 ( $L = 4$ )	3 ( $L = 8$ )	4 ( $L = 16$ )	5 ( $L = 32$ )	6 ( $L = 64$ )	7 ( $L = 128$ )	8 ( $L = 256$ )
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

# CÂMERA PINHOLE

Como registrar cada ponto

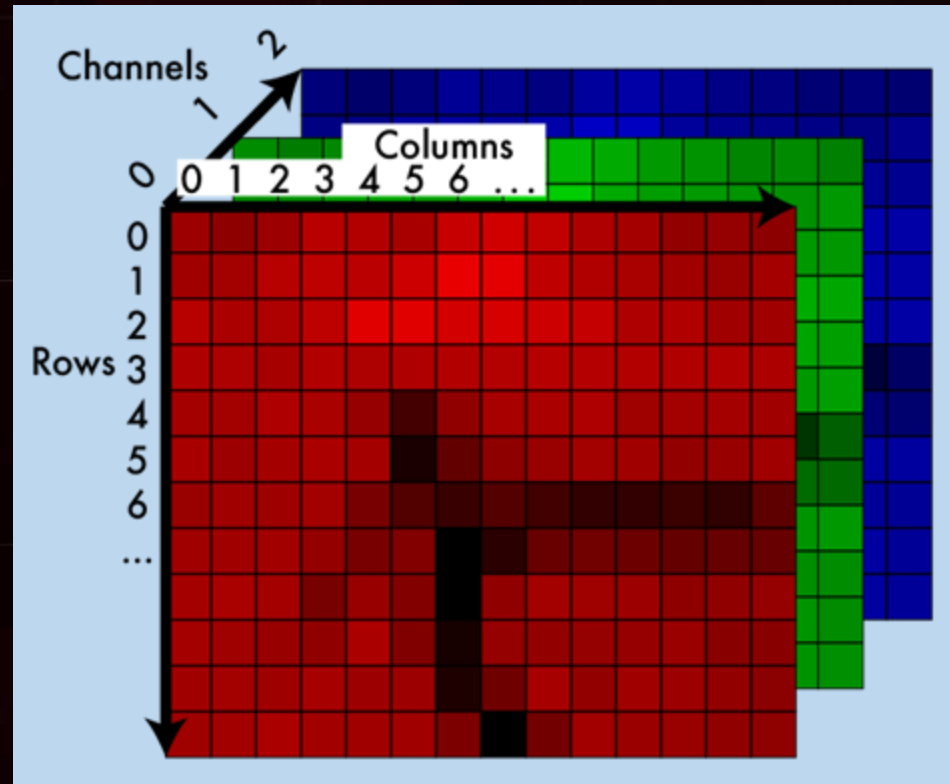


Sensores CMOS



# IMAGEM: QUANTIDADE DE LUZ

- **Pixels**
- $\uparrow$  = mais luz
- $\downarrow$  = menos luz
- mínimo: 0 = sem luz
- máximo: sensor
  - [0-255], 1 byte
  - [0-1], ponto flutuante
- E imagens coloridas?



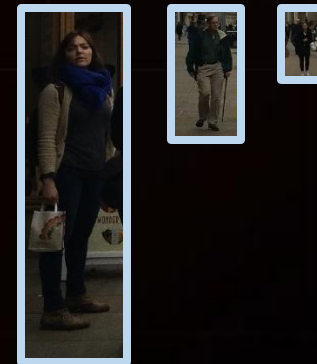
# ALGUNS FATOS SOBRE PROJEÇÃO



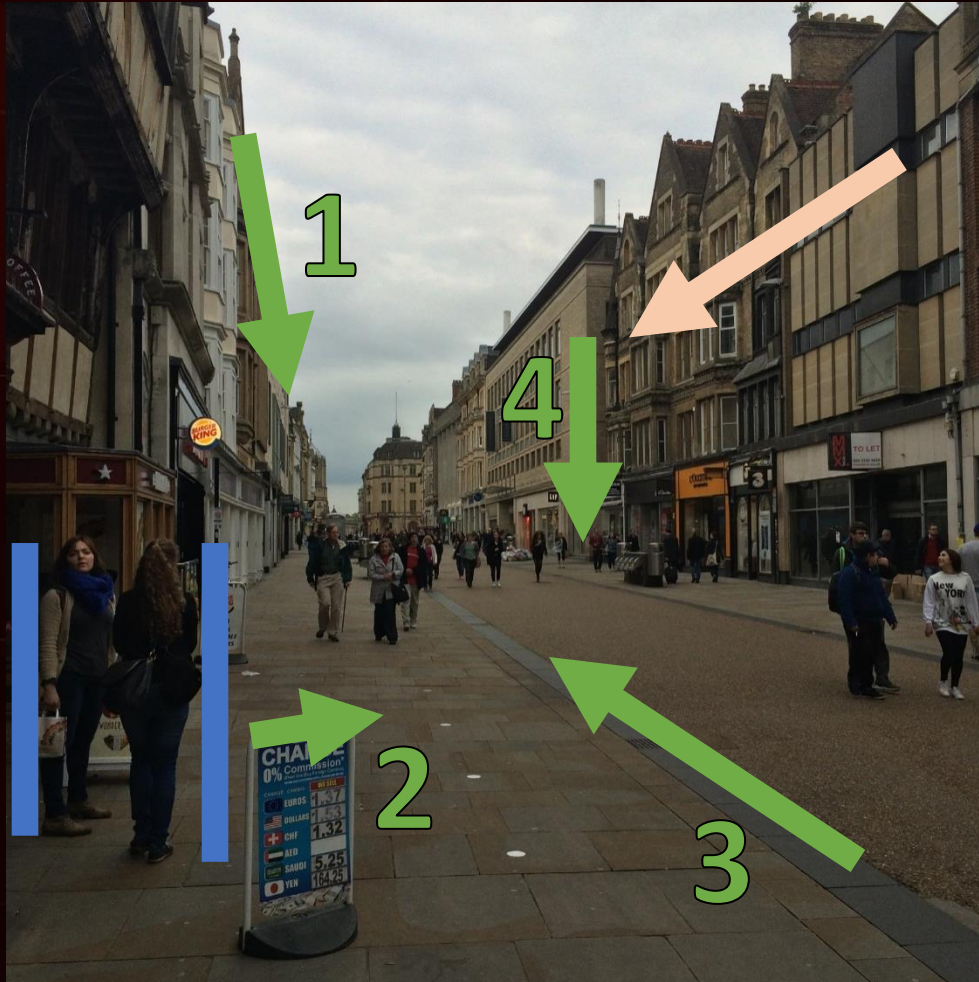
Linhas 3D projetadas para linhas 2D

A projeção de qualquer linha paralela em 3D converge para um ponto de desaparecimento

Objetos distantes são menores



# O QUE É PERDIDO?

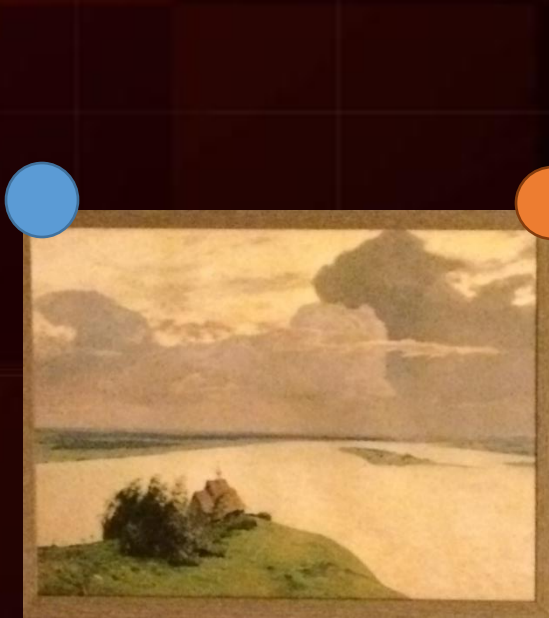


Está mais próximo ou distante?

As linhas **verdes** são paralelas / perpendiculares / nenhuma em relação à linha **laranja**?



# E A PERSPECTIVA?

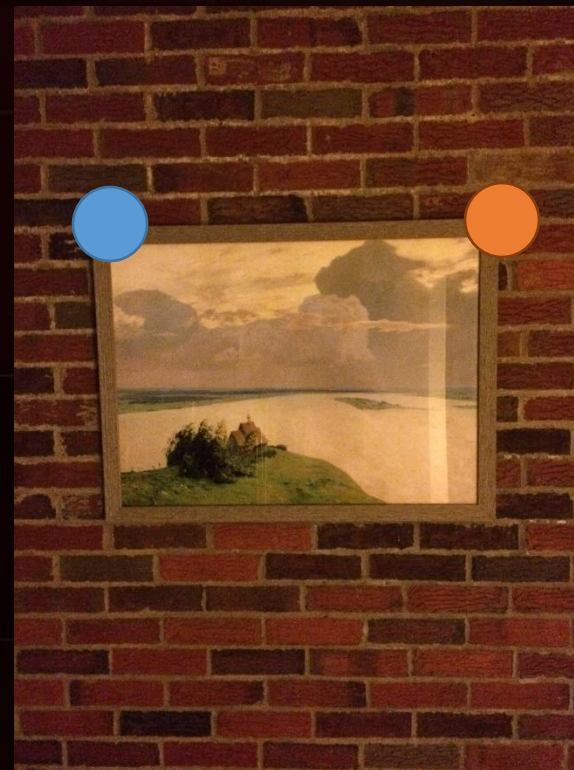


Y nos pontos  
Azul e Laranja  
na imagem



$$\frac{fy}{z_2}$$

$$\frac{fy}{z_1}$$



$$\frac{fy}{z}$$

$$\frac{fy}{z}$$

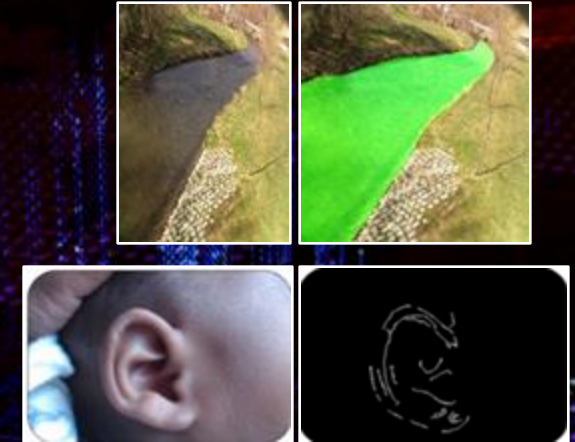
# COORDENADAS HOMOGÊNEAS

- Benefícios:
- Linhas (3D) e pontos (2D  $\rightarrow$  3D) agora estão na mesma dimensão.
- Usar *produto vetorial* ( $\times$ ) e *produto interno* para:
  - Intersecção de linhas  $l$  e  $m$ :  $l \times m$
  - Linha entre dois pontos  $p$  e  $q$ :  $p \times q$
  - Ponto  $p$  na linha  $l$ :  $l^T p$
- Linhas paralelas e verticais se tornam mais fáceis (comparado a  $y = mx + b$ )

# ETAPAS

**Pré-  
processamento:**  
melhora a  
qualidade da  
imagem.

domínio do problema





# TRANSFORMAÇÕES GEOMÉTRICAS

AS TRANSFORMAÇÕES GEOMÉTRICAS PODEM SER UTILIZADAS PARA:

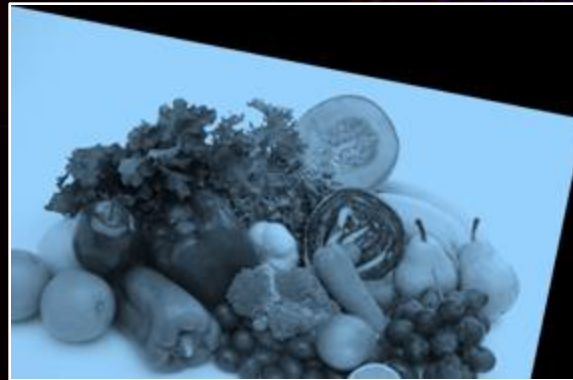
- **Editar imagens.** Ex. Formatação de publicações.
- **Alterar a orientação de cenários.** Ex. Fotografia.
- **No registro de imagens.** Ex. Imagens médicas.
- **Gerar imagem artística.** Ex. Montagem de comercial.
- **Inserir efeitos de distorção de objetos.** Ex. Face sorrindo.

## EX: ROTAÇÃO

Rotação por transformada inversa em ângulo  $\theta = 0.2$  radianos.

A rotação pode ser representada em forma de matriz de coordenadas homogêneas:

$$\begin{bmatrix} f \\ g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

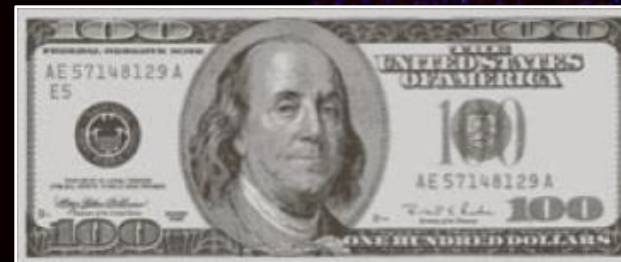




# TRANSFORMAÇÕES DE INTENSIDADE

## EX: INTENSIDADE POR *BITS*

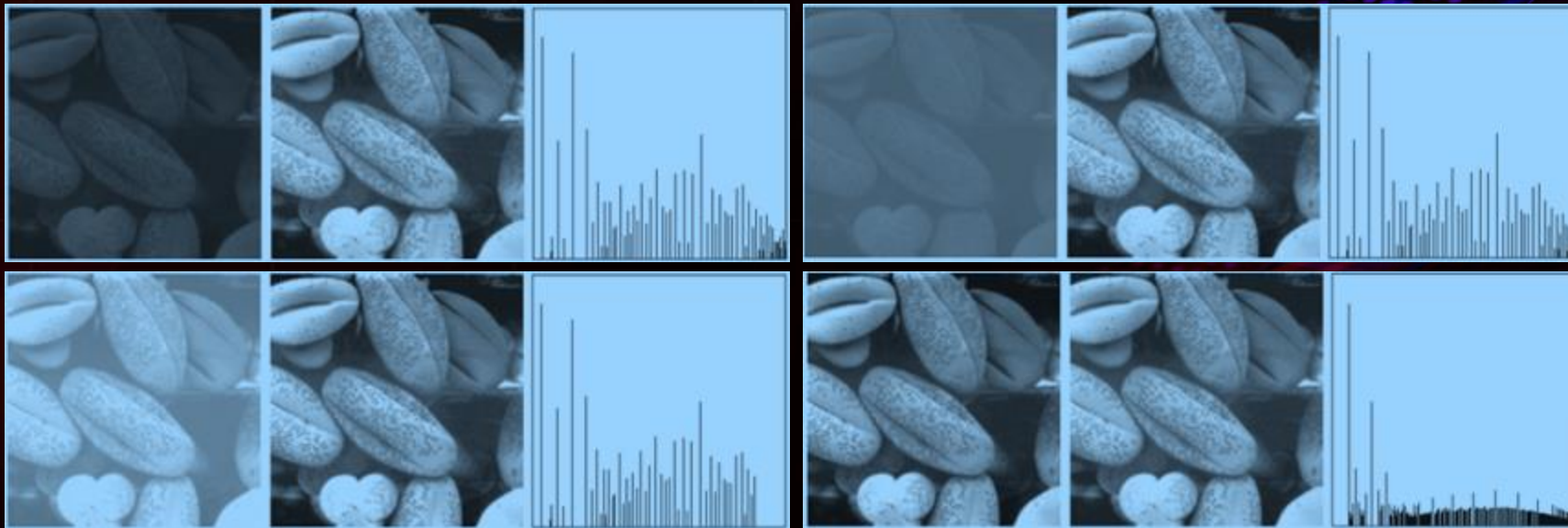
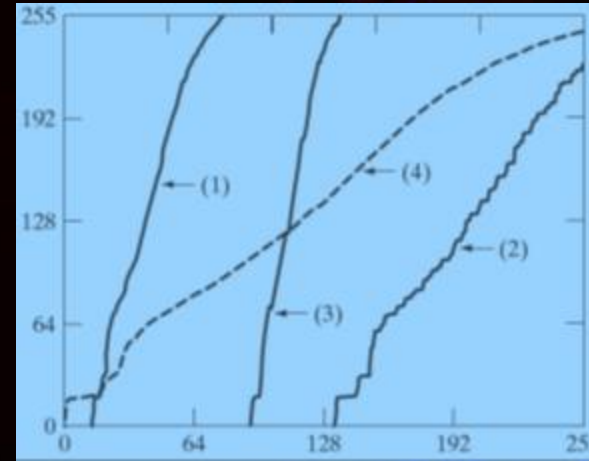
- Útil para verificar a contribuição de cada bit para a imagem e na compressão de imagens
- Reconstituição de imagem com apenas os 2, 3 e 4 últimos *bits*
- 1ª parece achatada
- 2ª possui falsos contornos de fundo
- 3ª está boa com apenas 50% da informação original





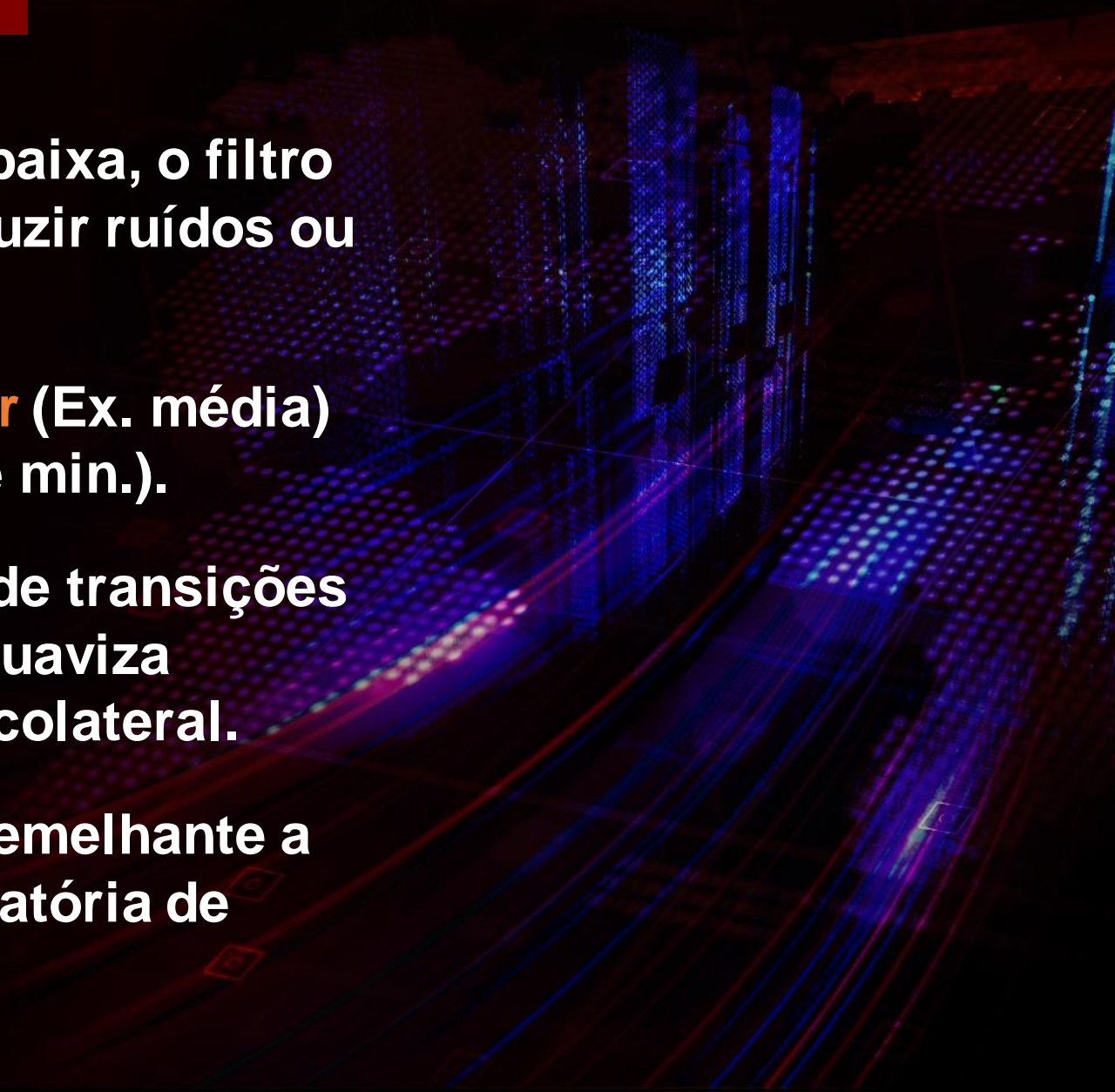
# HISTOGRAMA

**Exemplo:** equalização do histograma para realce de contraste



# FILTRO DE SUAVIZAÇÃO

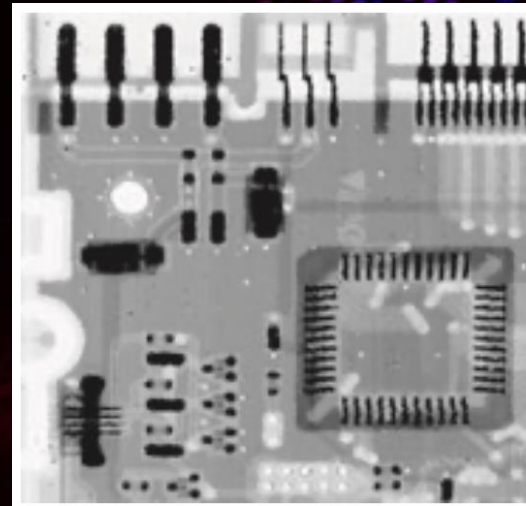
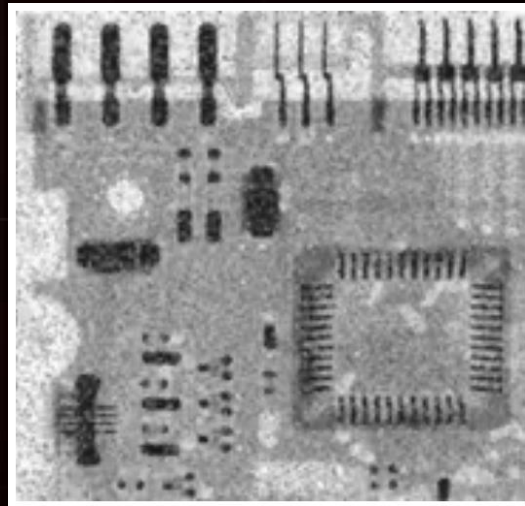
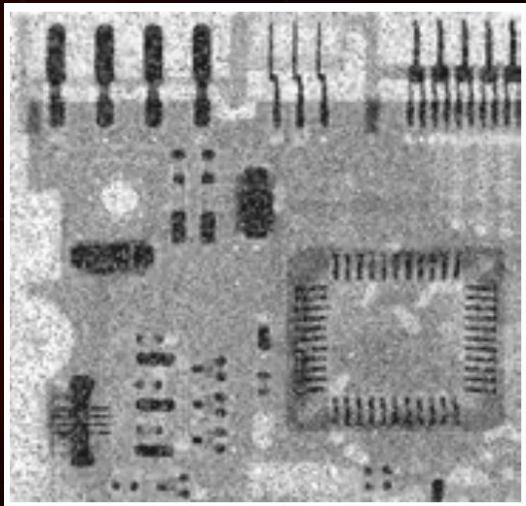
- Também chamado de filtro passa baixa, o filtro de suavização é utilizado para reduzir ruídos ou borrar a imagem original.
- Pode ser realizado com **filtro linear** (Ex. média) ou **não-linear** (Ex. mediana, máx. e min.).
- Apesar de reduzir ruídos que são de transições bruscas de intensidade, também suaviza arestas, o que pode ser um efeito colateral.
- Filtragem linear de suavização é semelhante a uma integração, ou seja, uma somatória de intensidades.





# EXEMPLO:

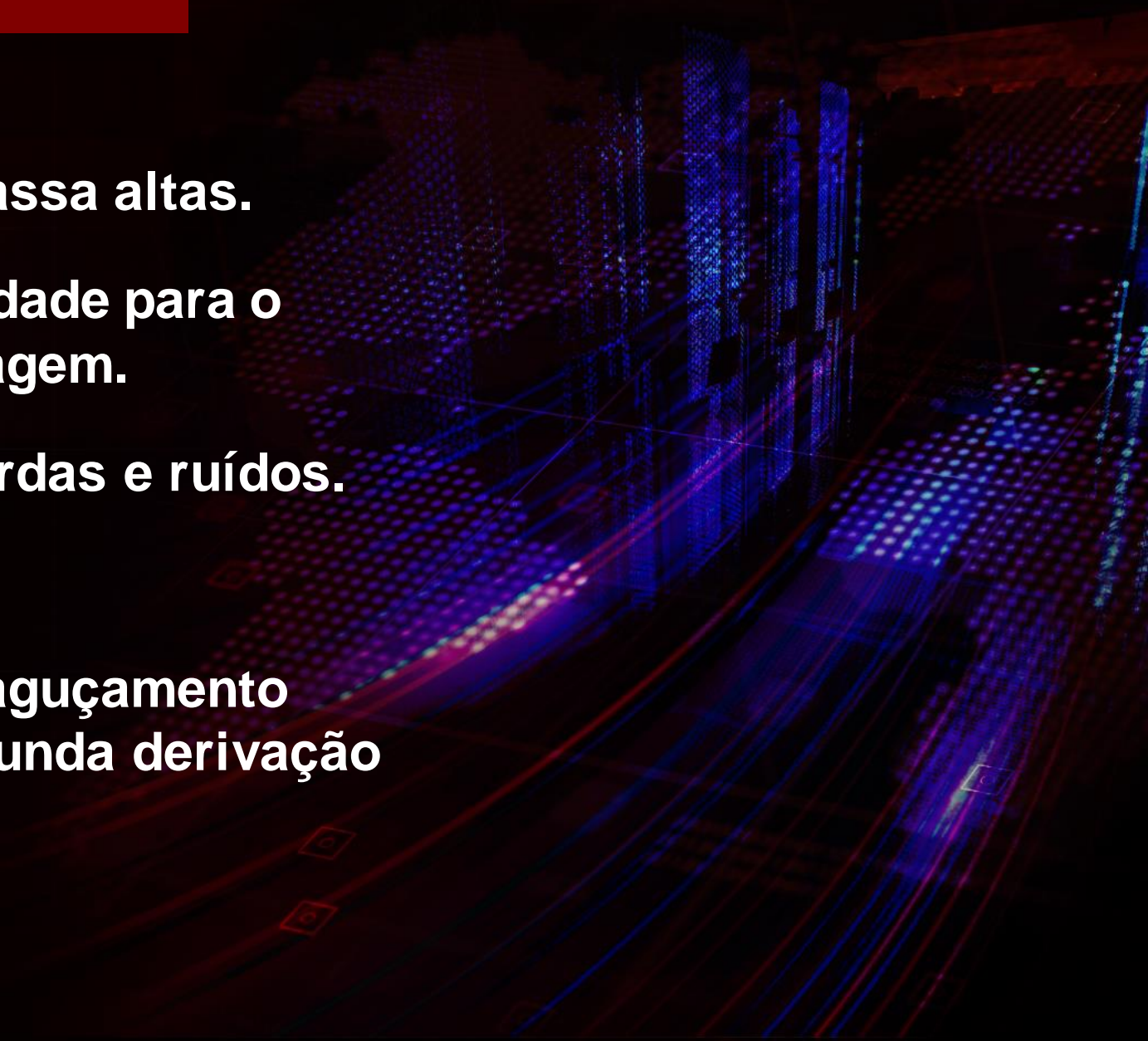
- a) Imagem corrompida pelo ruído sal e pimenta.
- b) Redução de ruído com um filtro de média 3x3.
- c) Redução de ruído com um filtro de mediana 3x3.





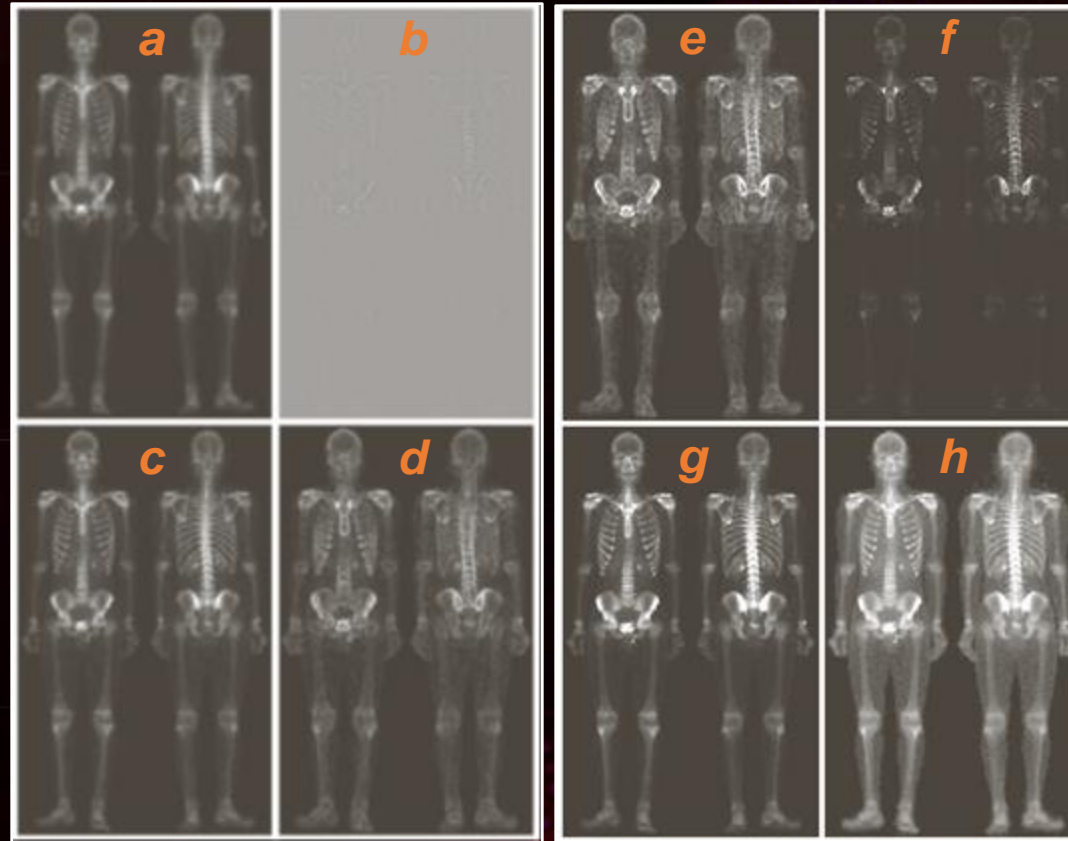
# FILTROS DE AGUÇAMENTO

- Também é chamado de filtro passa altas.
- Salientar transições de intensidade para o aumento da nitidez de uma imagem.
- Assim, diferenciação realça bordas e ruídos.
  - **Detentores de bordas**
- Serão considerados filtros de aguçamento baseados na primeira e na segunda derivação da imagem.



# APLICAÇÃO DE MÚLTIPLOS FILTROS

- a) Imagem original.
- b) Laplaciano de *a*.
- c) Aguçamento  $a+b$ .
- d) Sobel de *a*.
- e) *d* suavizada com filtro média 5x5.
- f) Máscara:  $c * e$ .
- g) Realçada:  $a+f$ .
- h) Transformação de potência em *g*.

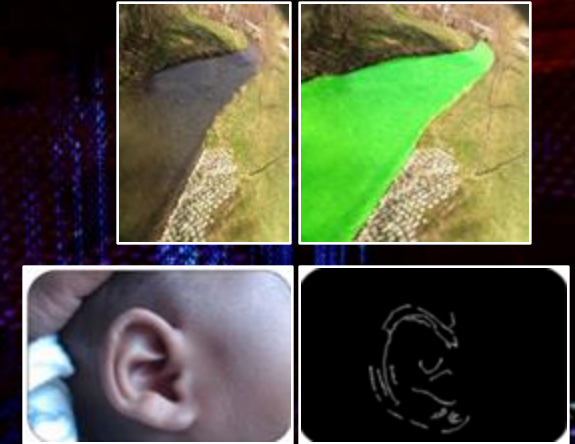
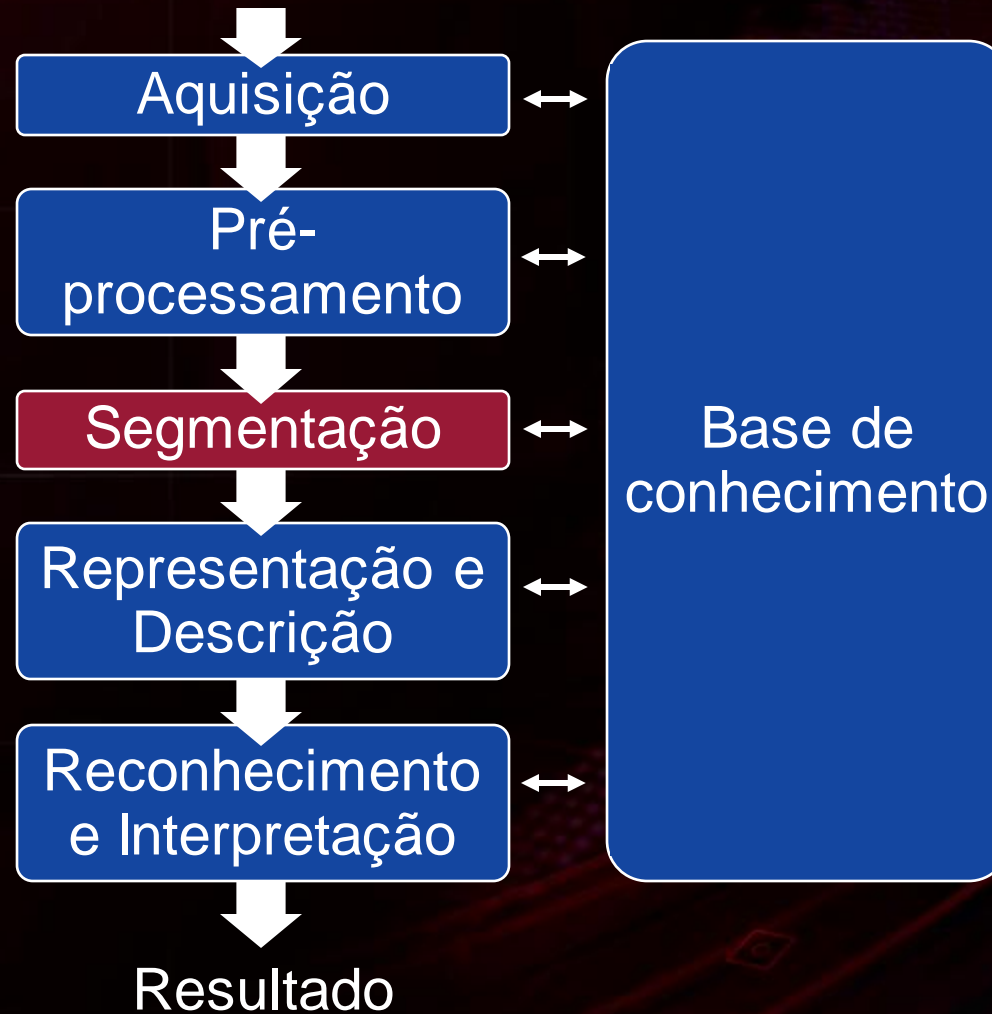




# ETAPAS

**Segmentação:**  
detecção de  
descontinuidades  
(bordas) ou de  
similaridades  
(regiões) na  
imagem.

domínio do problema





# TAREFAS DE VISÃO COMPUTACIONAL

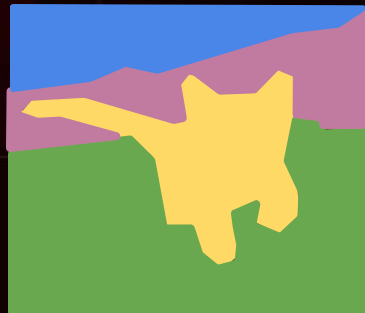
## Classificação



GATO

Sem  
extensão  
espacial

## Segmentação semântica



GRAMA GATO  
ÁRVORE CÉU

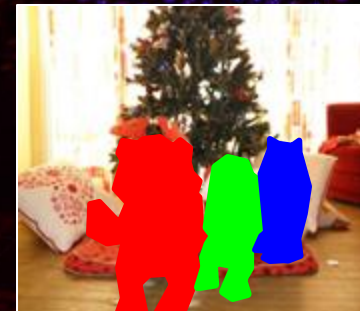
Nenhum  
objeto,  
apenas pixels

## Deteção de objetos



CÃO CÃO  
GATO

## Segmentação de instância

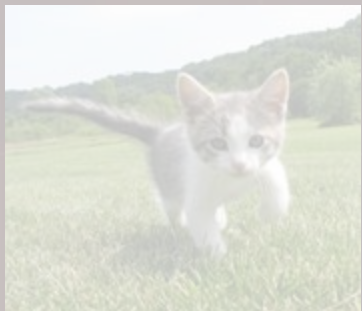


CÃO CÃO  
GATO

Múltiplos objetos

# SEGMENTAÇÃO SEMÂNTICA

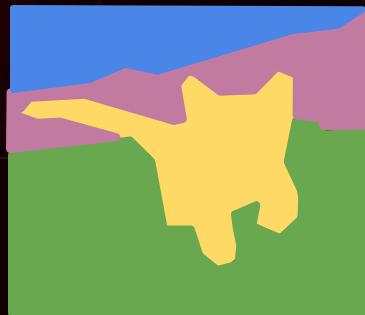
## Classificação



GATO

Sem  
extensão  
espacial

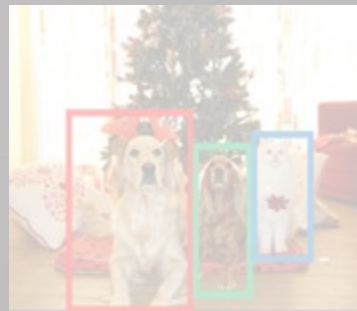
## Segmentação semântica



GRAMA GATO  
ÁRVORE CÉU

Nenhum  
objeto,  
apenas pixels

## Deteccção de objetos



CÃO CÃO  
GATO

## Segmentação de instância



CÃO CÃO  
GATO

Múltiplos objetos



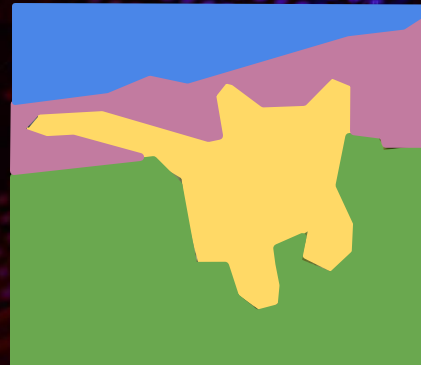
# SEGMENTAÇÃO SEMÂNTICA

A segmentação subdivide uma imagem em seus objetos e regiões constituintes, sendo uma das tarefas mais desafiadoras em processamento de imagens.

Rotular cada pixel da imagem com uma categoria. Não diferencia instâncias, se preocupa apenas com pixels.

Descontinuidade: **detecção de pontos, linhas e bordas**

Similaridade: **limiarização**





# EX:DETECÇÃO DE BORDAS

- Gradiente
- O detector de Borda Marr-Hildreth
- O detector de Bordas de Canny

Imagem original



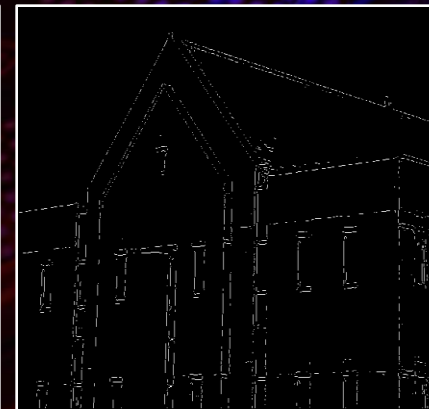
Detector de Canny  
( $T_L = 0,04$ ,  $T_H = 0,10$ ,  
 $\alpha = 4$ , filtro  $25 \times 25$ )



Gradiente limiarizado  
da imagem suavizada

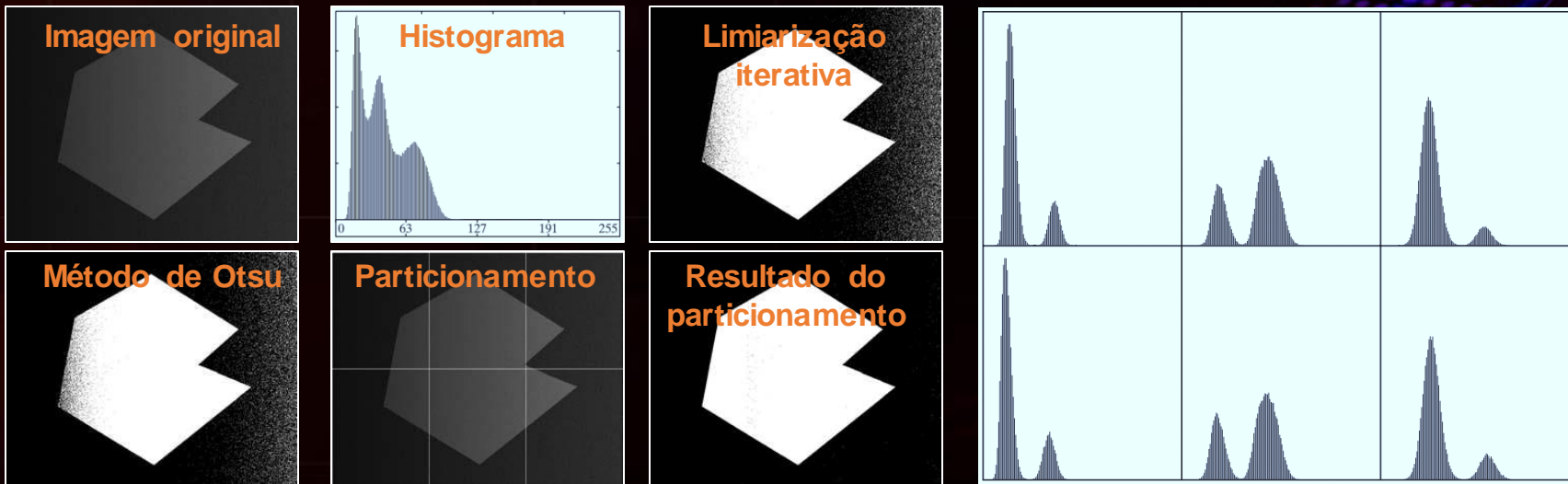


Algoritmo de Marr-  
Hildreth



# EX: LIMIAÇÃO VARIÁVEL

- Aplicar técnicas de correção de ruídos e iluminação nem sempre é suficiente, o que pode levar ao uso da **limiarização variável**, como o **particionamento**
- Porém, a subdivisão pode não funcionar quando os objetos e o fundo não ocupam regiões de tamanho razoavelmente comparável



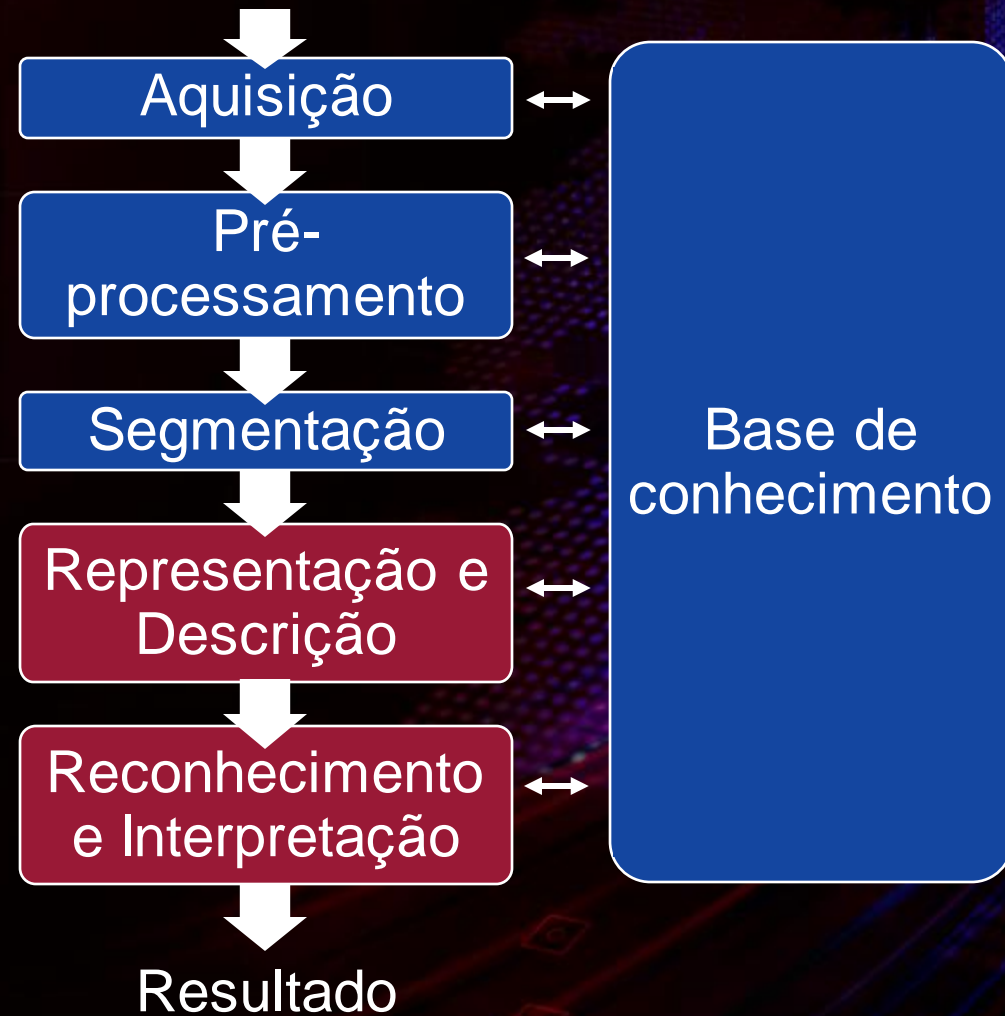


# ETAPAS

**Representação e  
Descrição**

**Reconhecimento e  
Interpretação**

domínio do problema





# CLASSIFICAÇÃO DE IMAGENS

## Classificação

Dados:  $(x, y)$

$x$  é a entrada / característica

$y$  é a classe / alvo

Tarefa: atribuir um rótulo/classe à imagem inteira

Classificação de imagens:  
Prever a categoria

$x$

$y$



Gato



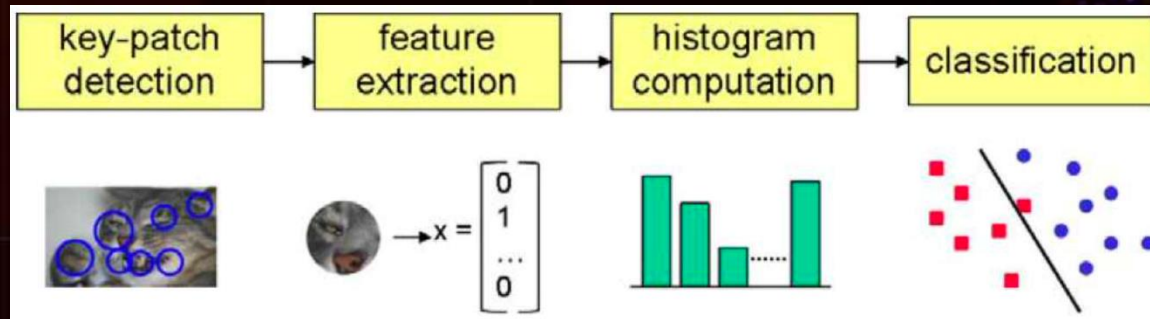
Cão



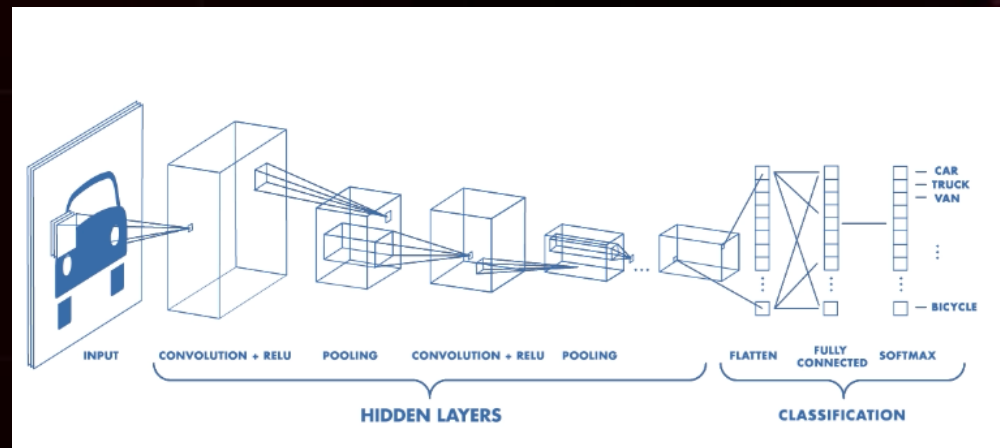
Macaco

# CLASSIFICAÇÃO DE IMAGENS

- **Abordagem clássica**



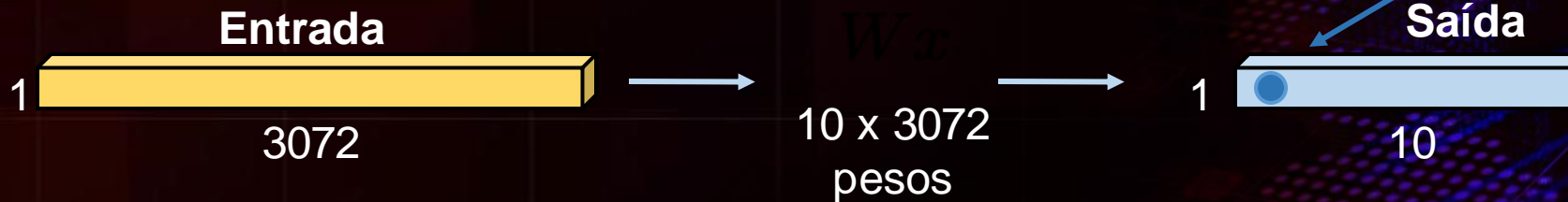
- **Abordagem mais atual**



# REDES NEURAIS

*Camadas totalmente conectadas (Fully-Connected Layers)*

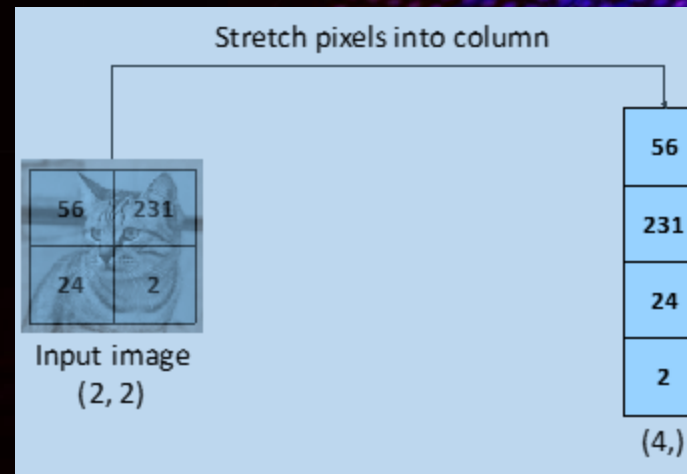
32x32x3 imagem -> estica para 3072 x 1



1 número: o resultado do produto interno entre uma linha de  $W$  e a entrada (um produto interno de 3072 dimensões)

**Problema:** os classificadores não respeitam a estrutura espacial de imagens.

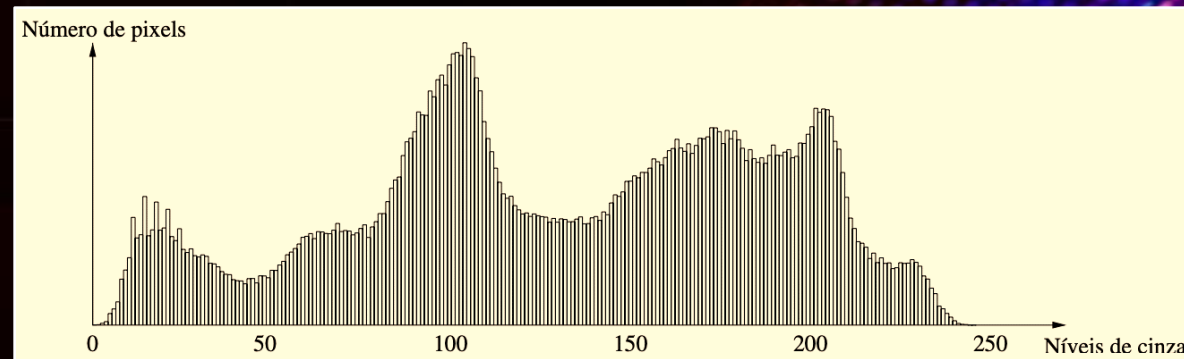
**Solução:** definir novos nós computacionais que operem em imagens.





# EXEMPLO: GCH

- **Global Color Histogram**
- Descritor de cores mais popular na literatura
- Abordagem global
- Contagem de ocorrências de cores (histogramas)

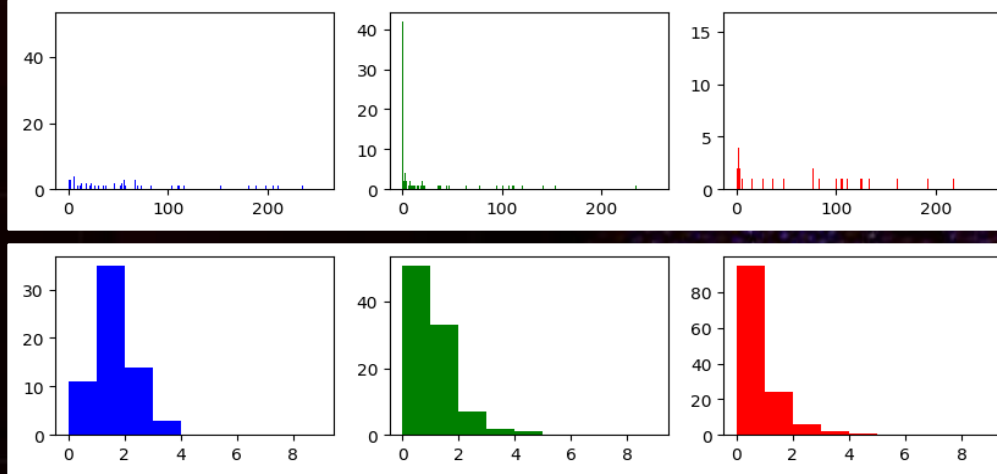


# EXEMPLO: GCH

- Quantização uniforme dos canais de cores (e.g., RGB e HSV)
- Histograma de imagem colorida com 64 posições ou BINS:
  - $color(x, y) = R(x, y) + fg * G(x, y) + fb * B(x, y)$
  - $color(x, y) = (S + fh * H)$
- Invariante à **escala e rotação**
- $\{R, G, B, S, H\}$  são normalizados
- $fg = 4, fb = 16$  e  $fh = 16$



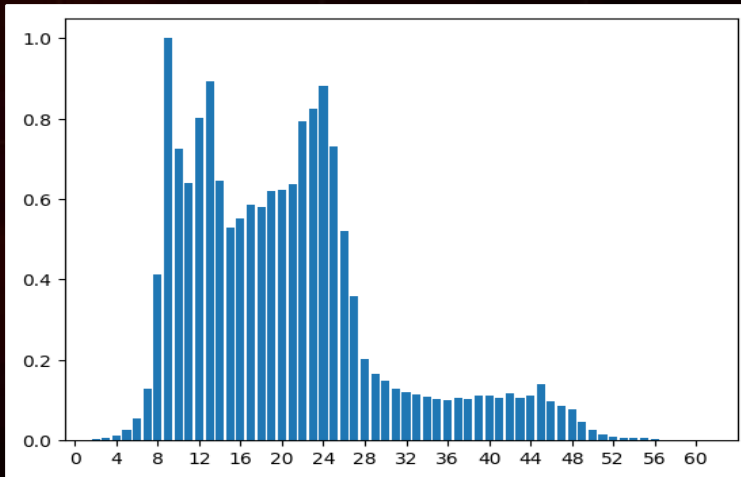
# EXEMPLO: GCH



$$color(x, y) = R(x, y) + fg * G(x, y) + fb * B(x, y)$$

$fg = 4$  e  $fb = 16$

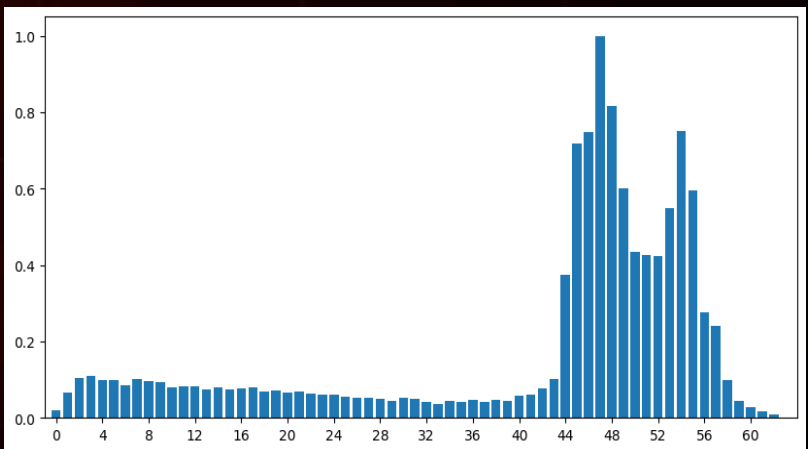
Normalizado



```
[2.0623e-04 1.5469e-03 2.4052e-03 5.4321e-03 1.1513e-02 2.5085e-02
5.3619e-02 1.2769e-01 4.1162e-01 1.0000e+00 7.2363e-01 6.3867e-01
8.0322e-01 8.9209e-01 6.4453e-01 5.2832e-01 5.5176e-01 5.8545e-01
5.8105e-01 6.2012e-01 6.2354e-01 6.3721e-01 7.9346e-01 8.2422e-01
8.8184e-01 7.3193e-01 5.2148e-01 3.5815e-01 2.0276e-01 1.6638e-01
[ 6 45 70 158 335 730 1560 3714 11974 29094 21059 18584
23367 25958 18755 15364 16053 17027 16912 18045 18146 18532 23091 23974
25663 21289 15176 10422 5898 4841 4294 3708 3516 3369 3179 3001
2940 3057 2985 3270 3211 3097 3406 3041 3282 4036 2847 2492
2227 1353 753 455 265 222 203 169 107 47 24 15
4 2 0 1]
```



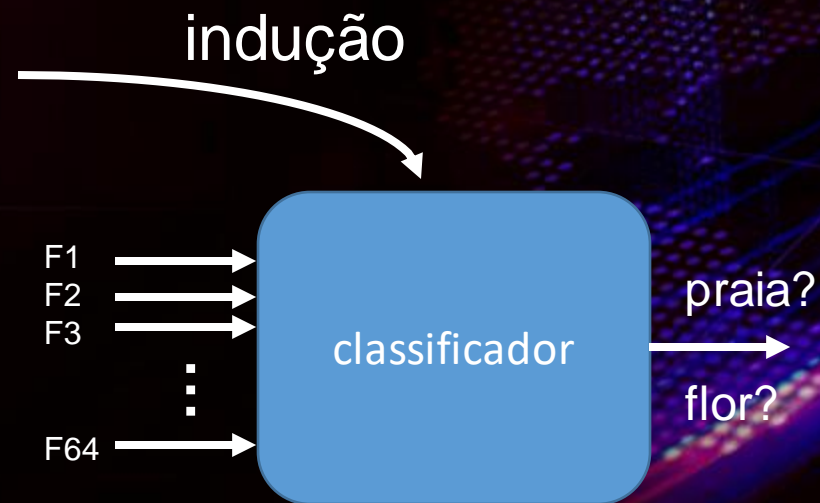
# EXEMPLO: GCH



F1	F2	F3	...	F64	C
.006	.033	.054	...	0.00	praia
.026	.259	.439	...	0.00	flor
.264	.816	1.00	...	0.00	flor
⋮					
.021	.066	.104	...	0.00	praia

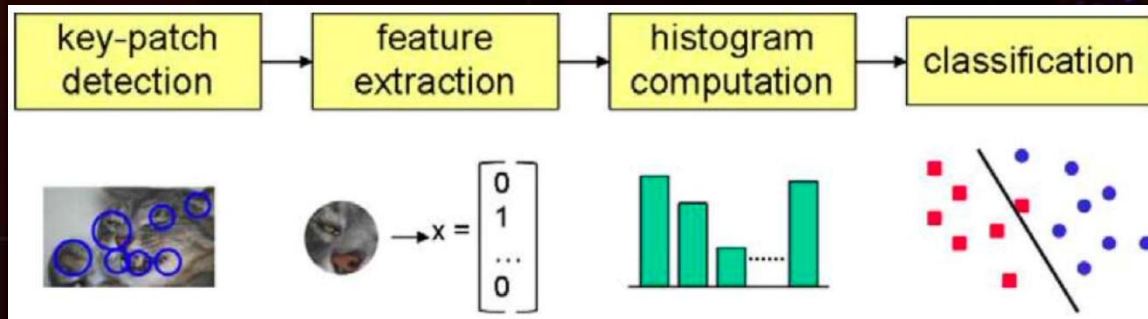
# EXEMPLO: GCH

F1	F2	F3	...	F64	C
.006	.033	.054	...	0.00	praia
.026	.259	.439	...	0.00	flor
.264	.816	1.00	...	0.00	flor
⋮					
.021	.066	.104	...	0.00	praia

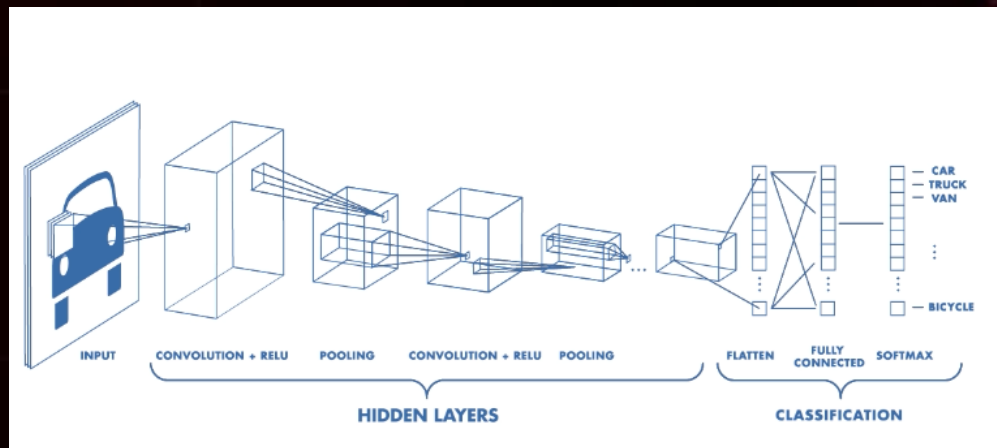


# CLASSIFICAÇÃO DE IMAGENS

- Abordagem clássica



- Abordagem mais atual

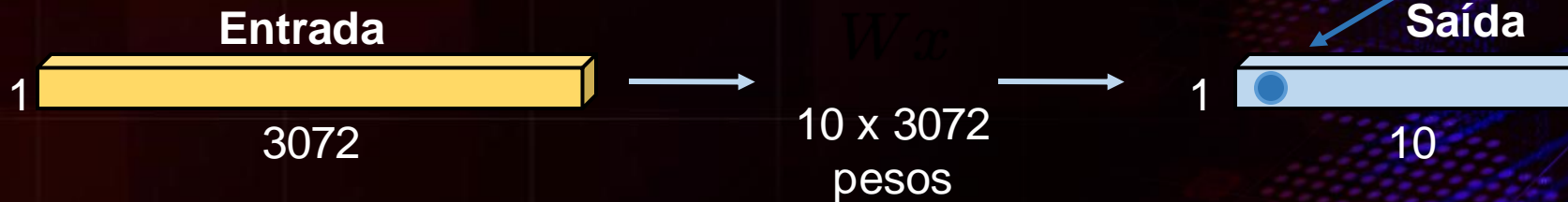




# REDES NEURAIS

*Camadas totalmente conectadas (Fully-Connected Layers)*

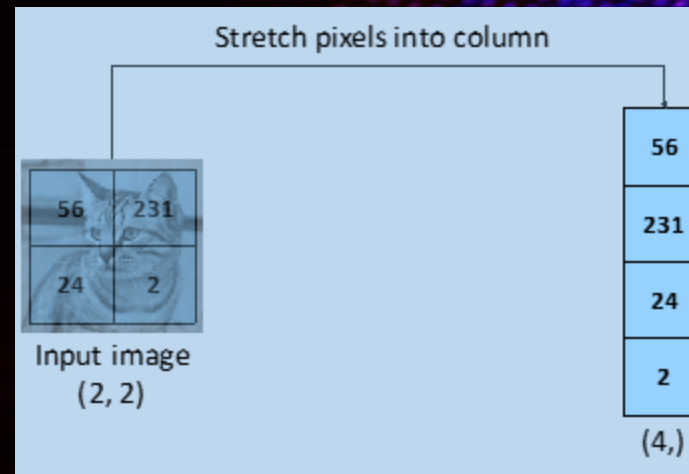
32x32x3 imagem -> estica para 3072 x 1



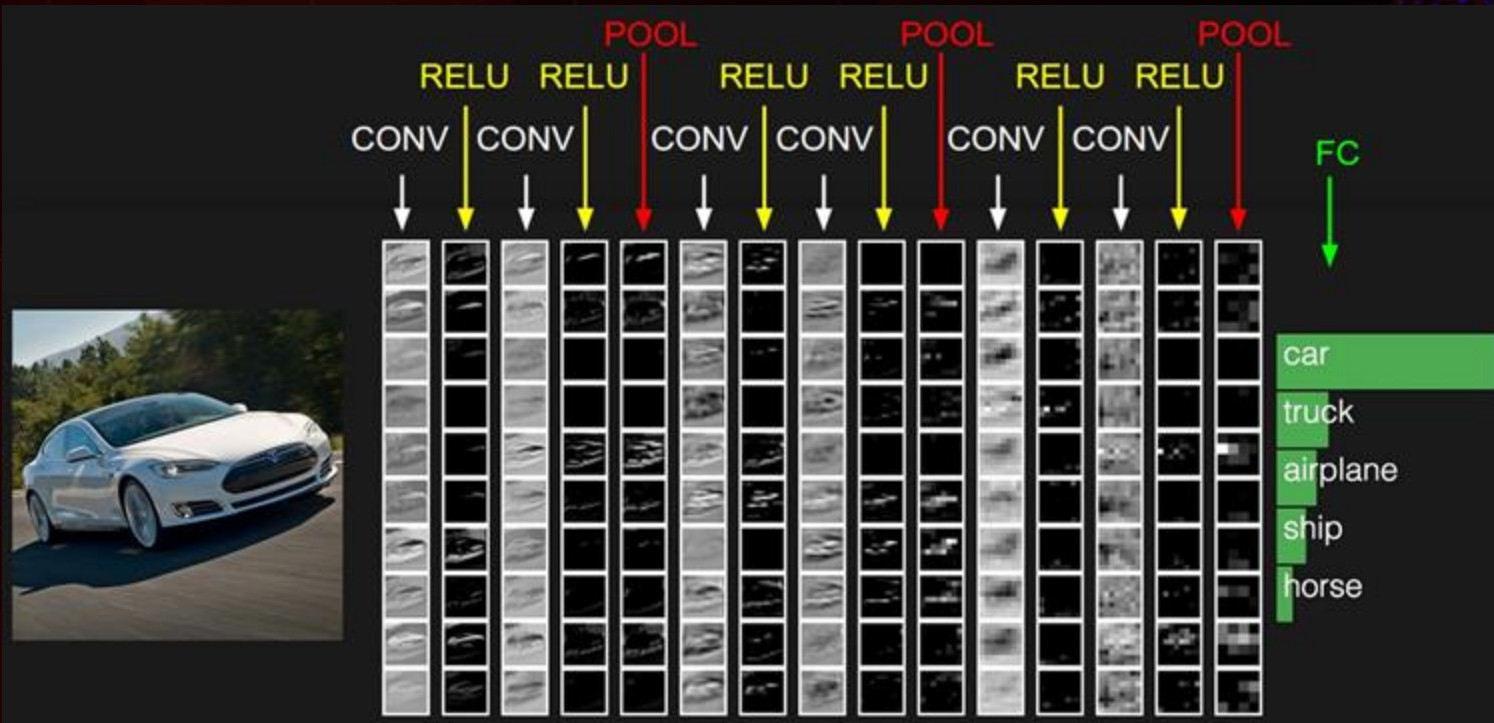
1 número: o resultado do produto interno entre uma linha de  $W$  e a entrada (um produto interno de 3072 dimensões)

**Problema:** os classificadores não respeitam a estrutura espacial de imagens.

**Solução:** definir novos nós computacionais que operem em imagens.



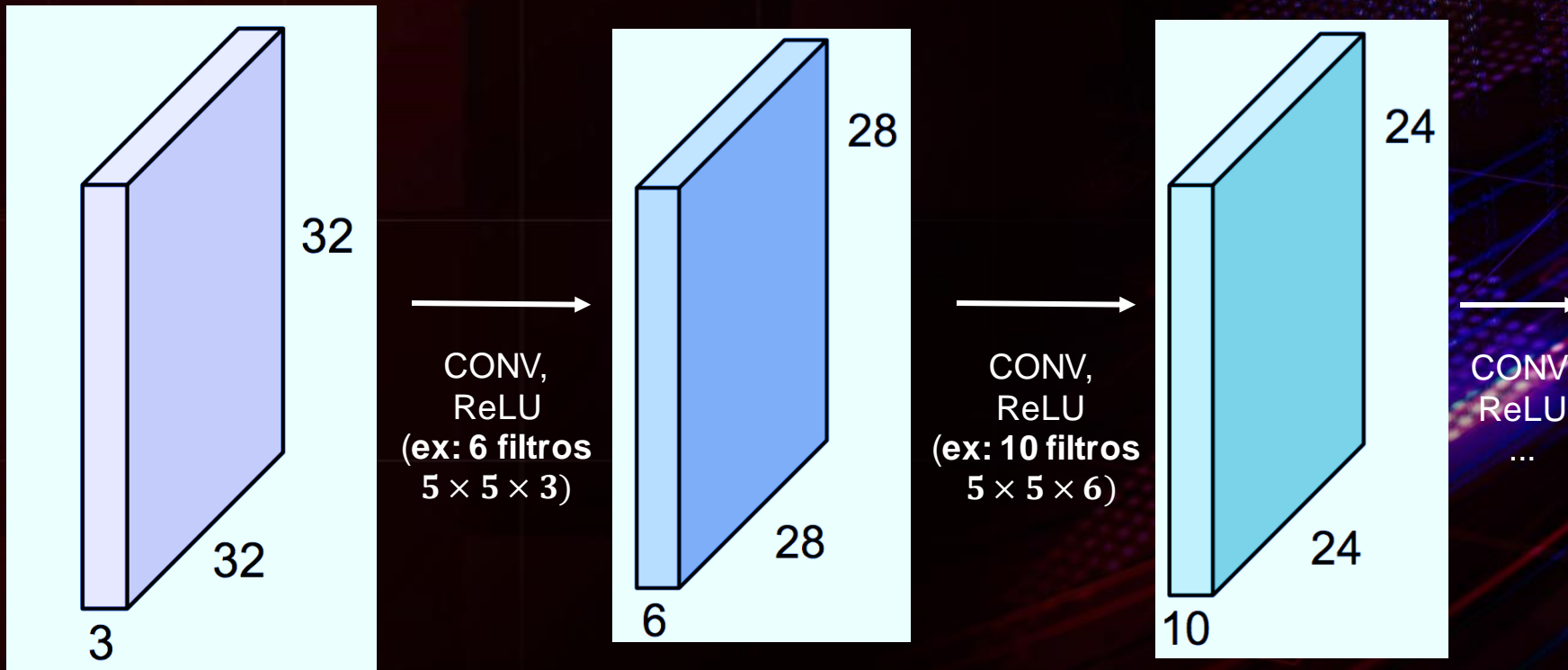
# REDE NEURAL CONVOLUCIONAL





# REDE CONVOLUCIONAL

“Sequência de camadas de convolução intercaladas por **funções de ativação**”





# CAMADA DE POOLING

- **Reduz os volumes** para que se tornem gerenciáveis
- Atua sobre cada mapa de ativação de **maneira independente**

2	3	2	5
5	6	9	8
3	2	1	2
1	2	0	1

1 único mapa de ativação

MAX *pooling* com  
filtro  $2 \times 2$  e *stride* 2



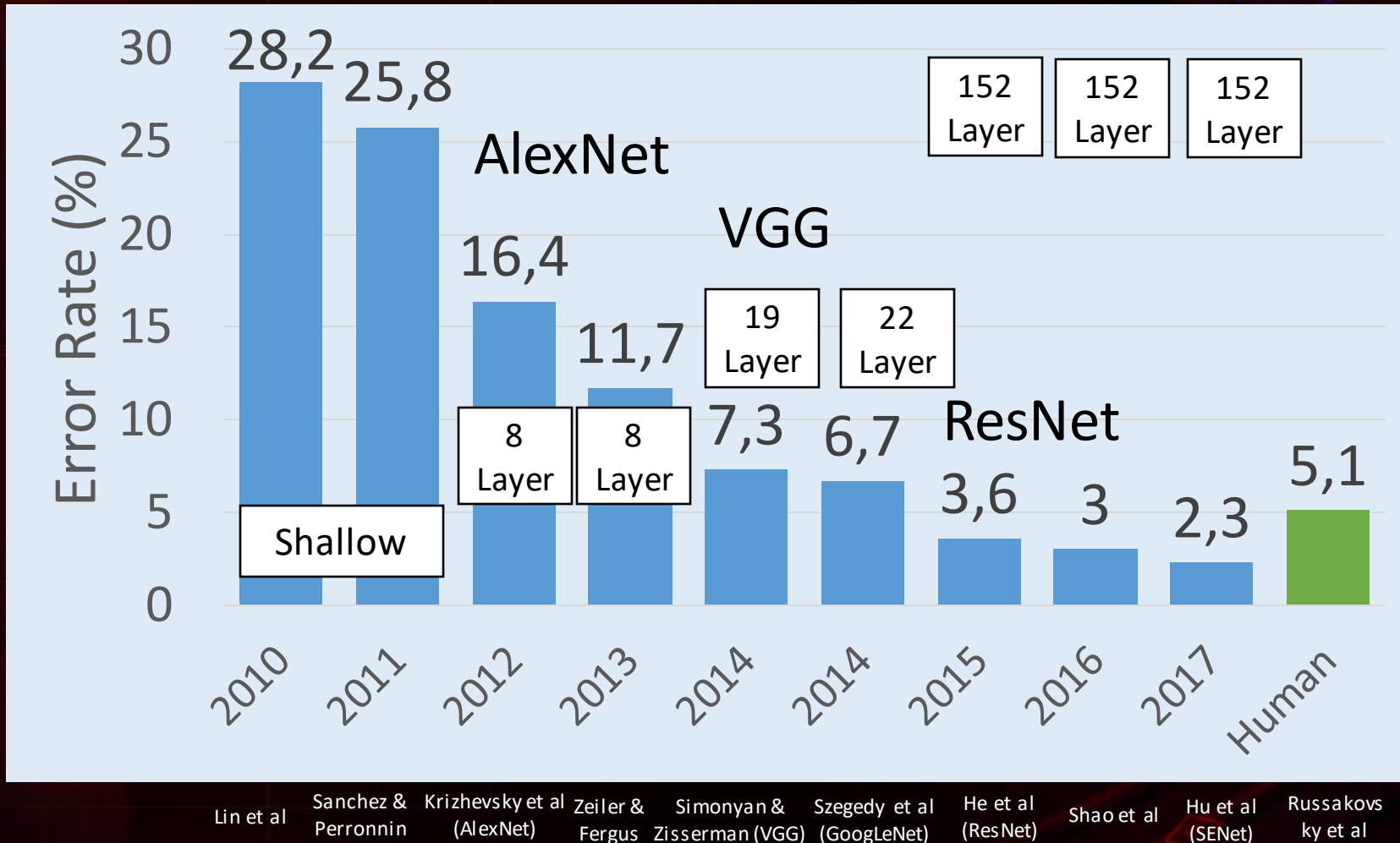
6	9
3	2

AVG *pooling* com  
filtro  $2 \times 2$  e *stride* 2



4	6
2	1

# ARQUITETURAS CNN





# AVALIAÇÃO DE CLASSIFICADORES

Existem diferentes protocolos para realizar a separação dos dados disponíveis em conjuntos de treinamento e teste:

- *Holdout*
- Reamostragem aleatória (*Random Subsampling*)
- Validação Cruzada (*Cross-Validation*)
- *Leave-one-out*
- *Bootstrap*



# MATRIZ DE CONFUSÃO

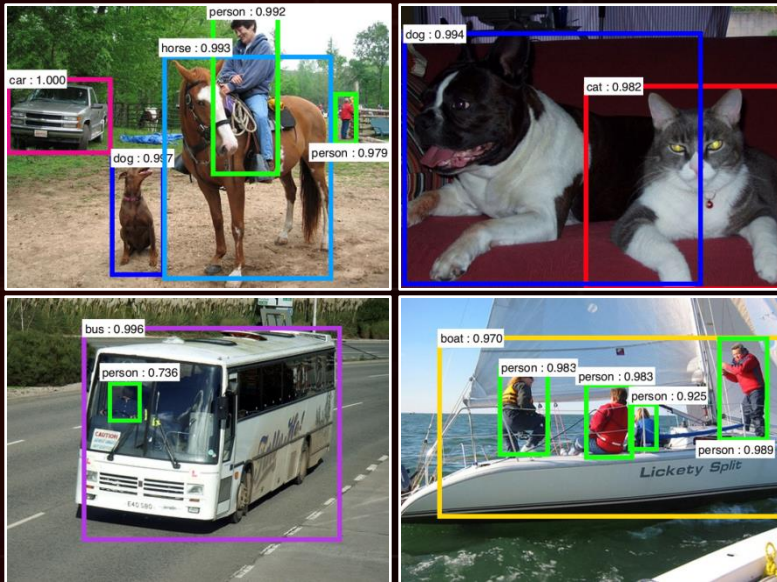
- Também chamada de Tabela de Contingência
  - Permite a extração de **diversas medidas** de desempenho preditivo
  - Usada para distinguir os tipos de erros
  - Usada para problemas binários ou multiclasse

Classe Prevista	Classe Verdadeira		
	A	B	C
A	25	10	0
B	0	40	0
C	5	0	20

Valores fora da diagonal principal: erros!

# APLICAÇÕES

# Detecção



[Faster R-CNN: Ren, He, Girshick e Sun, 2015]

# Segmentação

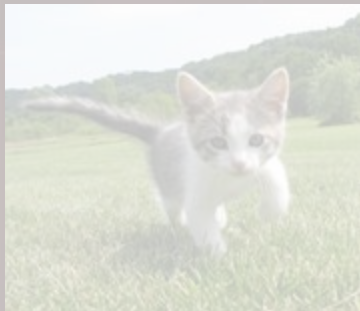


**[Farabet et al., 2012]**



# DETECÇÃO DE OBJETOS

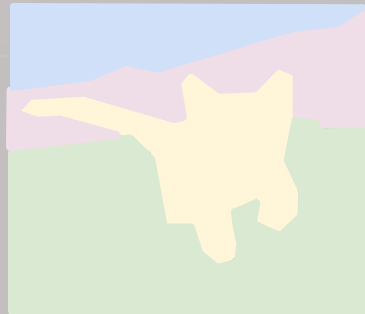
## Classificação



GATO

Sem  
extensão  
espacial

## Segmentação semântica



GRAMA GATO  
ÁRVORE CÉU

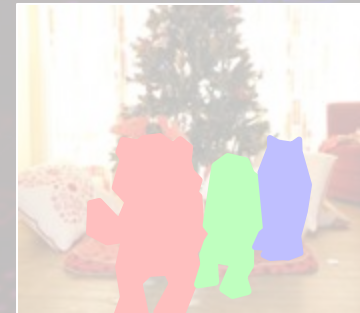
Nenhum objeto,  
apenas pixels

## Detecção de objetos



CÃO CÃO  
GATO

## Segmentação de instância



CÃO CÃO  
GATO

Múltiplos objetos



# EXEMPLO: DETECÇÃO DE PESSOA

Dalal & Triggs '05:

Treinar **SVM** usando descritores **HOG**

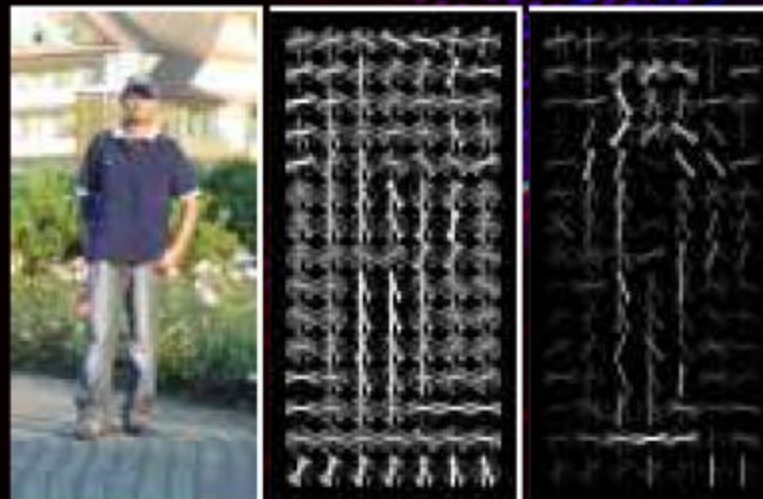
2 classes, pessoa / não pessoa

No teste:

Extrair HOGs em várias escalas

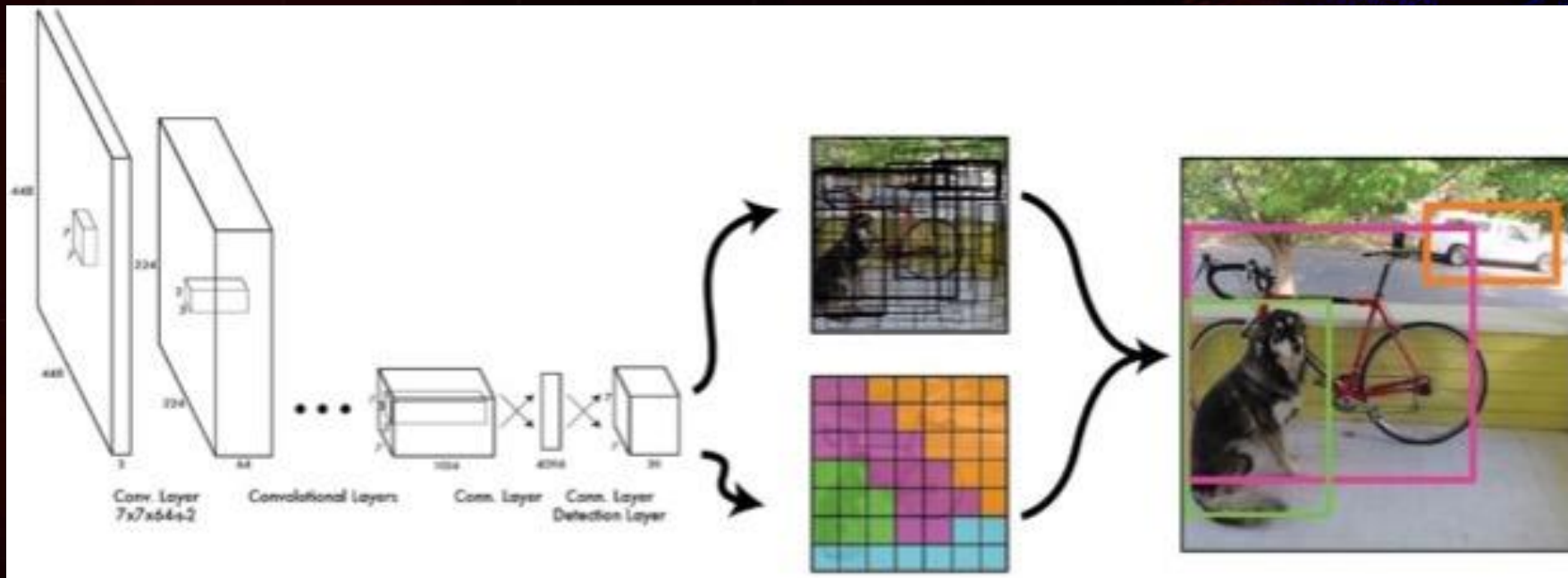
Rodar SVM em todas localizações

Respostas = Pessoa?



# SUCESSO DAS REDES NEURAIS

- É possível treinar uma rede neural (profunda) para realizar o *pipeline* completo

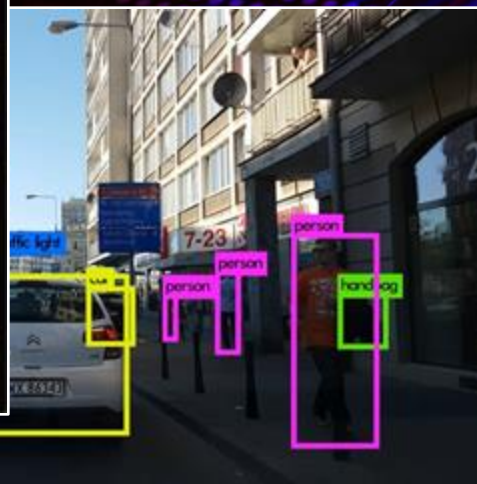




# SEGMENTAÇÃO X DETECÇÃO

- Rótulos por pixels
- Apenas categoria

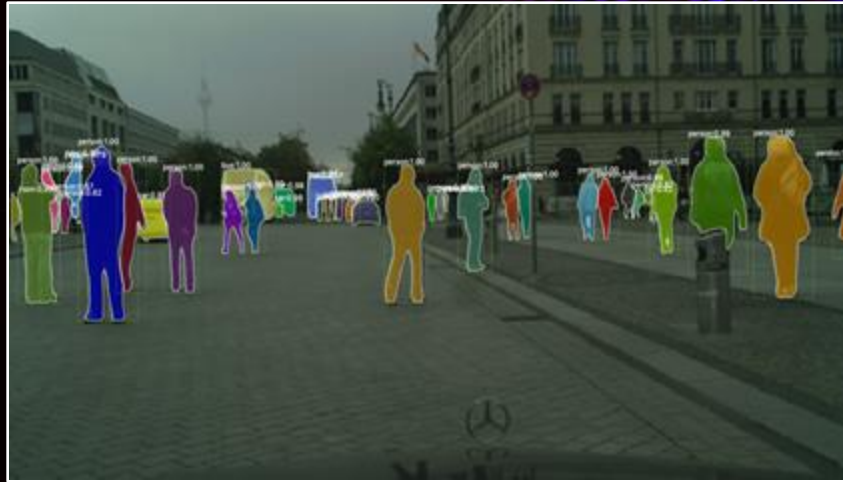
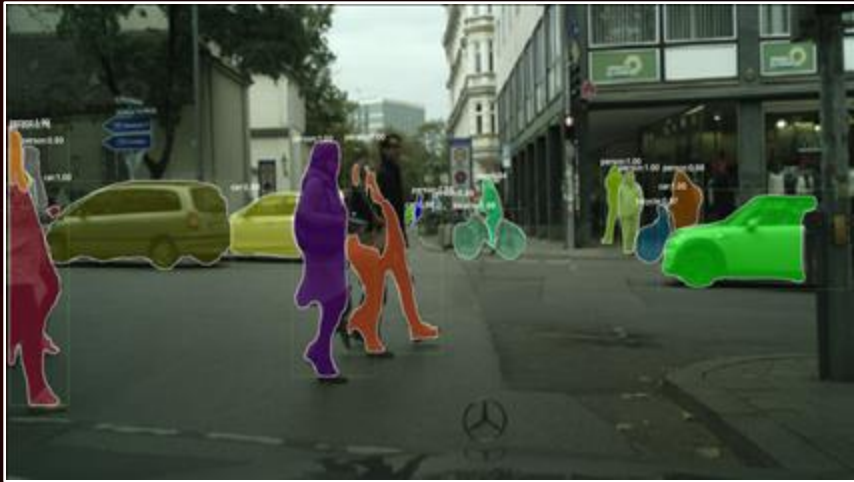
Rótulos por 'caixas'  
Categoria + instância





# SEGMENTAÇÃO DE INSTÂNCIA

- Segmentação em nível de instância  
A qual classe cada pixel pertence  
Também a qual instância



# RESUMO

1. Introdução à Visão Computacional
2. Pré-processamento de imagens
3. Imagens coloridas e câmeras
4. Aprendizado de Máquina
5. Segmentação de imagens
6. Classificação de imagens
7. Reconhecimento de objetos



# **VISÃO COMPUTACIONAL**

**Webconferência**

**Revisão**

**Prof. Márcio P. Basgalupp**