

Grupo de Estudos de Visão Computacional

Davi de Jesus
Hugo Pessoni

Apoio



Introdução à Visão Computacional e Estrutura de Imagens

Goiânia
2024



INF

INSTITUTO DE
INFORMÁTICA

Sumário

1. Boas Vindas
2. História
3. Introdução a Visão Computacional
4. Estrutura de Imagens
5. Espaços de Cor
6. Formatos de Imagem
7. Aplicações de Visão Computacional
8. Projeto Prático

Boas Vindas

- Objetivos do grupo:
 - Aprendizado
 - **Incentivar** e ensinar os mais novos
 - Material de **apoio** para projetos
 - **Portifólio** pessoal
 - Gerar **oportunidades**
- Metodologia:
 - Cronograma inicial de **9 semanas**
 - Apresentações semanais (**quintas das 18 as 20**)
 - 2 apresentadores (ideal)
 - Provocar **discussões saudáveis** em aula
 - Aplicações **práticas**

Boas Vindas

The screenshot shows the GitHub repository 'R2CV'. The README page contains sections for 'Propósito', 'Conteúdo do Repositório', and 'Semana 1: Boas-vindas / Introdução à Visão Computacional e Estrutura de Imagens'. The repository has 15 commits from 'dvzk1' and 1 branch.

Nosso repositório no GitHub:

- **Propósito:** Documentar o nosso trabalho e servir como um norte para quem almeja ingressar na área ou necessita buscar conhecimento prático e mais acessível possível para projetos/faculdade.
- Slides, Notebooks, Sugestões de Materiais e Resumos são bem-vindos!
- Todos podem e **devem** contribuir, mas para isso, tem de passar pelas **Pulls-Requests**!
- **Melhoria contínua** do conteúdo, garantindo longevidade ao projeto.

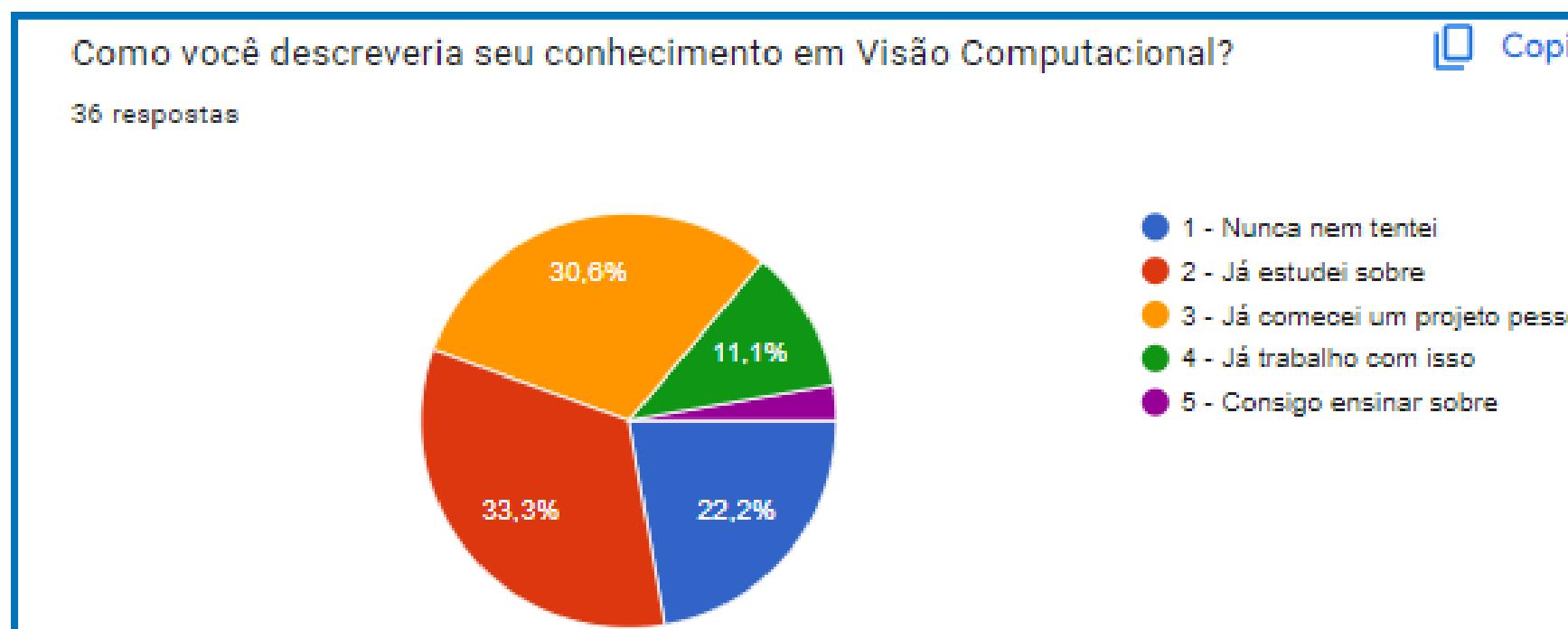
Link de acesso: <https://github.com/dvzk1/R2CV>

Boas Vindas

- **Expectativas:**
 - Proatividade
 - Compromisso
 - Propósito
 - União
- **Calendário:**
 1. Introdução à Visão Computacional e Estrutura de Imagens
 2. Estrutura das Imagens Digitais e Espaços de Cores
 3. Detecção de Bordas e Extração de Características
 4. Algoritmos Clássicos vs Modernos para Detecção e Reconhecimento
 5. Introdução ao Deep Learning – Redes Neurais Perceptron e MLP
 6. Regularização e Otimização em Redes Neurais
 7. Redes Neurais Convolucionais (CNNs)
 8. Arquiteturas Avançadas de CNNs
 9. Redes Neurais Recorrentes (RNNs) e Aplicações em Vídeo

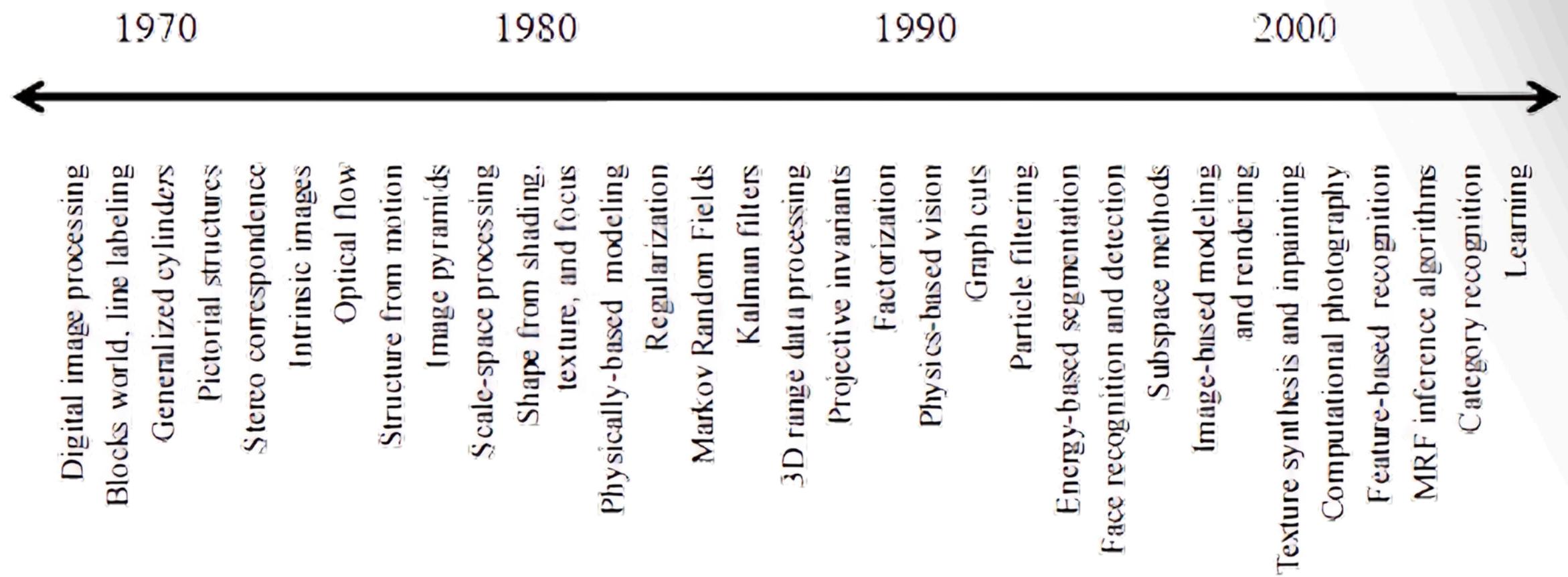
Boas Vindas

- **Ferramentas “necessárias”:**
 - Google Colab / Ambiente de desenvolvimento
 - Python e Bibliotecas
 - Noções matemáticas (Álgebra Linear, Cálculo, Probabilidade e Estatística)
- **Métricas do grupo:**



História

- Timeline:



- Década de 70: início de uma "agenda" para mimetizar a inteligência humana nas máquinas.
 - Buscavam formas de dar uma percepção visual para as máquinas

Surgiram alguns questionamentos iniciais:

- Como são representadas as informações para o computador?
- Como calcular o resultado desejado a partir dessa imagens?
- Como as representações e algoritmos são mapeados em hardware real?

História

- **Década de 80:** foco na criação de métodos matemáticos mais sofisticados para melhorar a extração de valores da imagem.
 - Canny-Edge (algoritmo de detecção de bordas)
 - snake (primeira tentativa em tracking)
- **Década de 90:**
- Temas novos como computação gráfica e renderização.
- Primeiros modelos de visão computacional com um viés mais estatístico. Estatística e física começaram a andar de mãos dadas
 - Algoritmos de reconhecimento facial

História

- **E por fim, os anos 2000:** Os avanços em modelagem estatística e aprendizado de máquina foram um ponto de inflexão para a área de visão computacional
- Abordagens data-driven para modelos de visão computacional se tornaram o core principal de pesquisa na área.

E hoje, assim como todas as outras áreas de IA, a visão computacional caminha a passos largos em várias áreas como classificação e detecção de objetos, segmentação de imagens, rastreamento de objetos (tracking), reconstrução 3D a partir de múltiplas imagens 2D ou vídeos, OCR (Reconhecimento Óptico de Caracteres) etc.

Mas enfim, o que é Visão Computacional?

Visão computacional pode ser definida essencialmente pelo processo de descrever as imagens para o computador, deixar que ele reconstrua as propriedades dessas imagens, e a partir dessa reconstrução, conseguir gerar insights valiosos de forma computadorizada sobre o mundo que vemos.

Um problema de contextualização

Parafraseando Richard Szeliski:

"Hoje em dia, apesar dos avanços significativos em visão computacional (Vision Transformers, Modelos Multi-modais etc.), que melhoraram bastante o desempenho em tarefas de visão computacional, como classificação de imagens, segmentação, e geração de descrições, as máquinas ainda estão longe de alcançar a compreensão profunda e causal que uma criança de dois anos tem ao observar uma imagem."

A visão computacional é uma área complexa justamente porque é um **problema inverso**. Isso significa que, a partir de informações incompletas ou ambíguas, tentamos inferir aspectos complexos do mundo real, como a forma, a cor, a textura, e a posição tridimensional dos objetos.

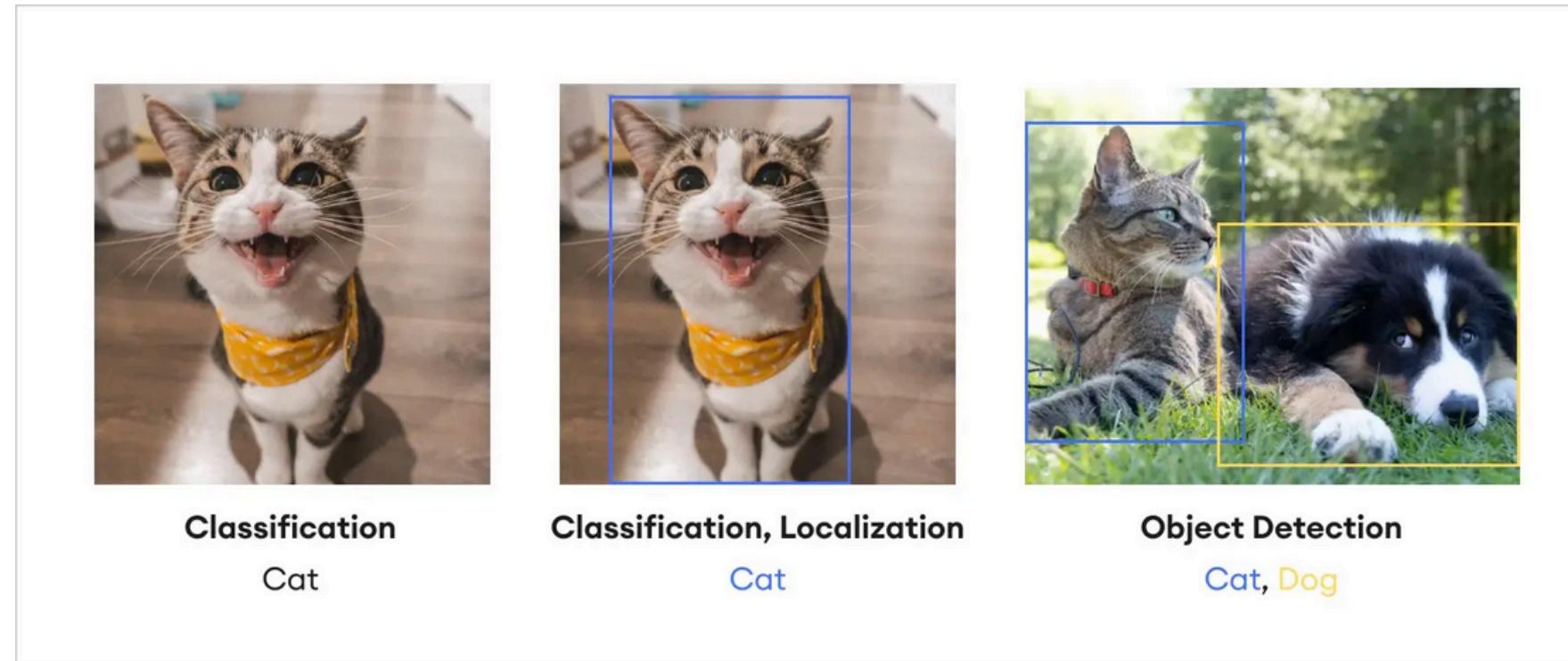
Física, matemática e computação

- E graças a física, a matemática e a computação, que vem melhorando a cada dia, estamos começando a contextualizar uma máquina sobre o mundo que nós humanos vemos.

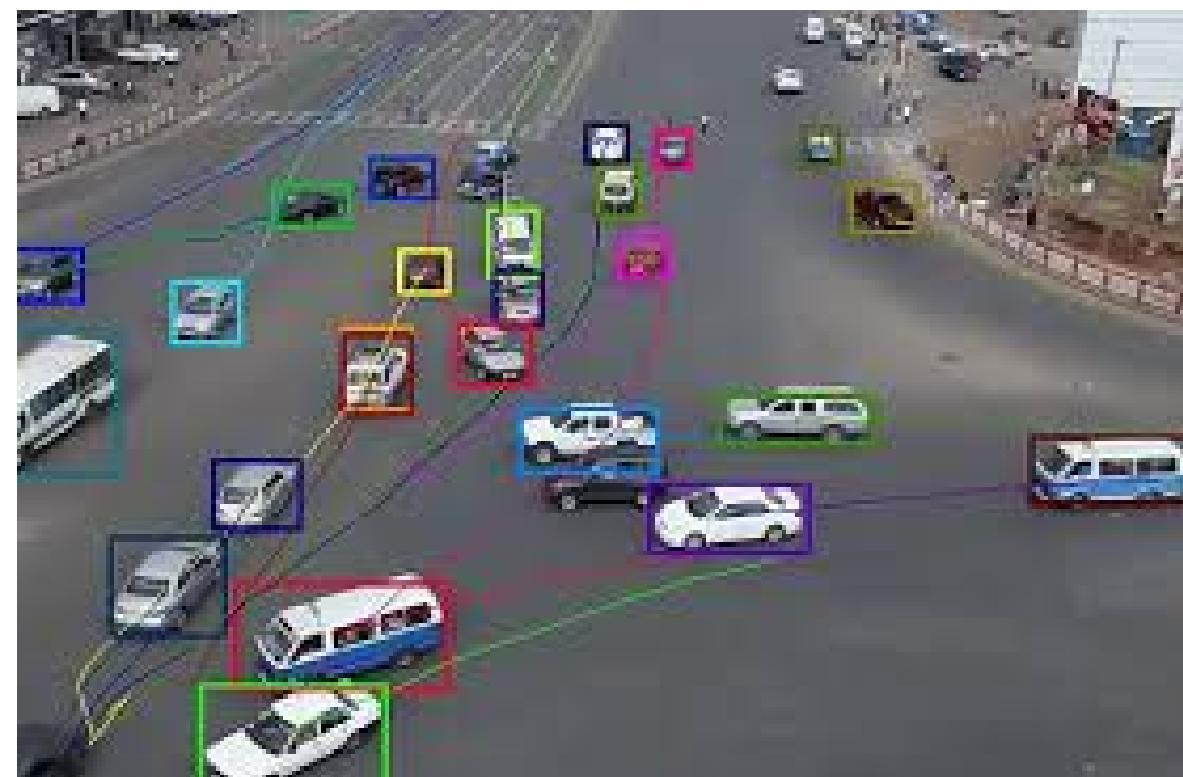
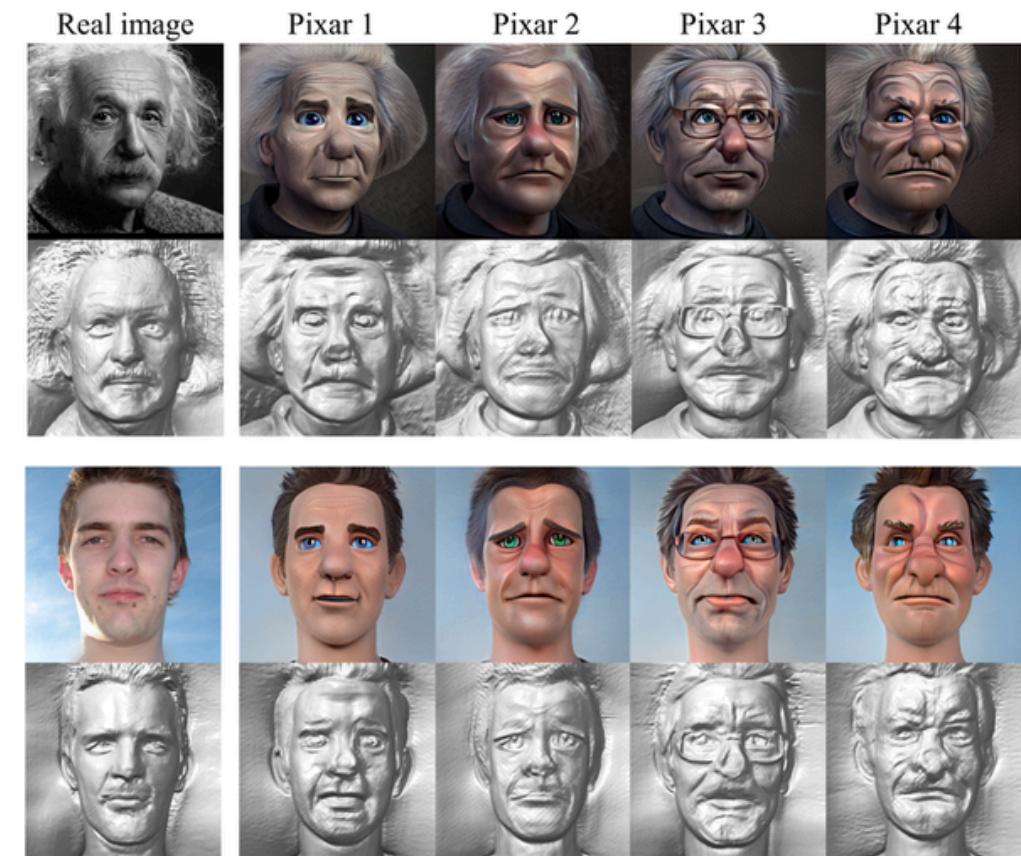
O ponto central é: a visão computacional envolve a conversão de dados visuais (imagens) em representações numéricas, que os algoritmos conseguem processar.

- As imagens que um computador processa são compostas por milhões de pequenos elementos chamados **pixels**.
- Cada pixel é uma unidade individual que contém informações sobre cor ou intensidade de luz da imagem.
- Ou seja, uma imagem digital é basicamente uma matriz de pixels! E ao processar essas representações numéricas (os pixels), o computador pode aplicar transformações e extrações de características que permitem identificar e classificar objetos nas imagens por exemplo.

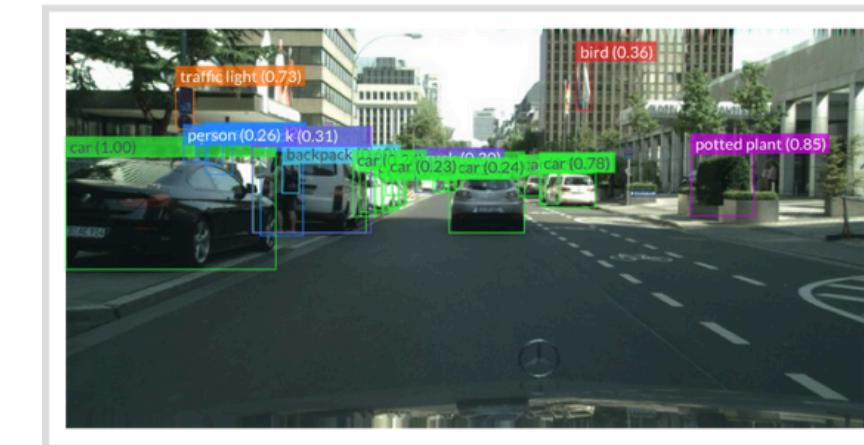
Exemplos de tarefas comuns em visão computacional



Manipulated images with 3D reconstruction



Object Detection



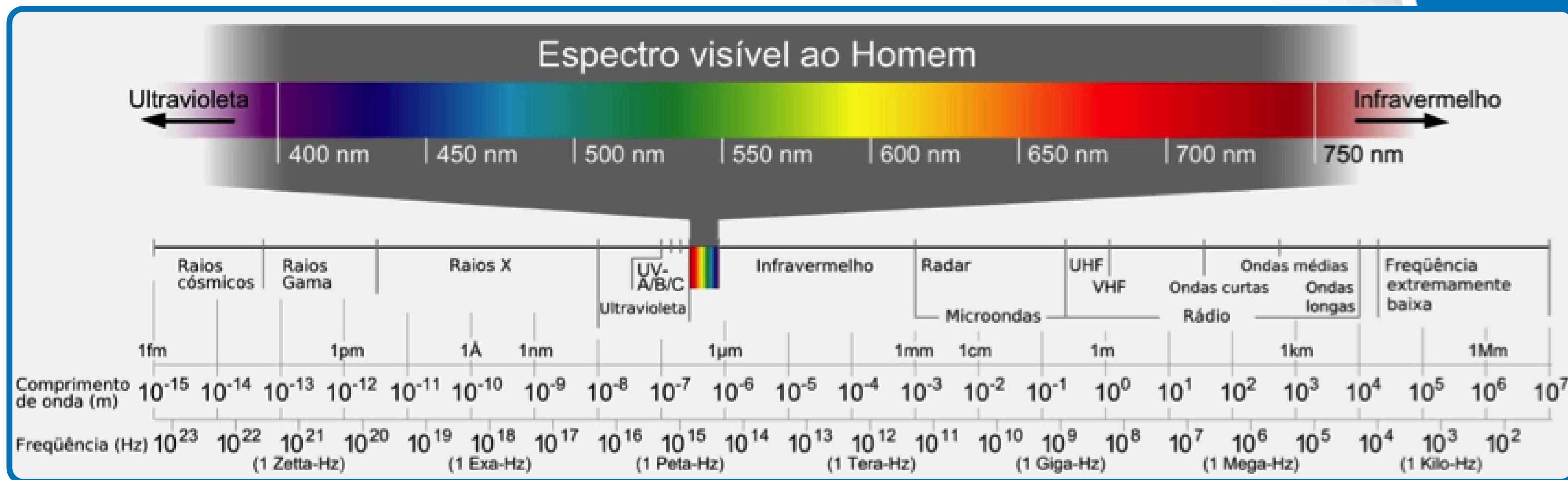
FasterRCNN

Image Segmentation



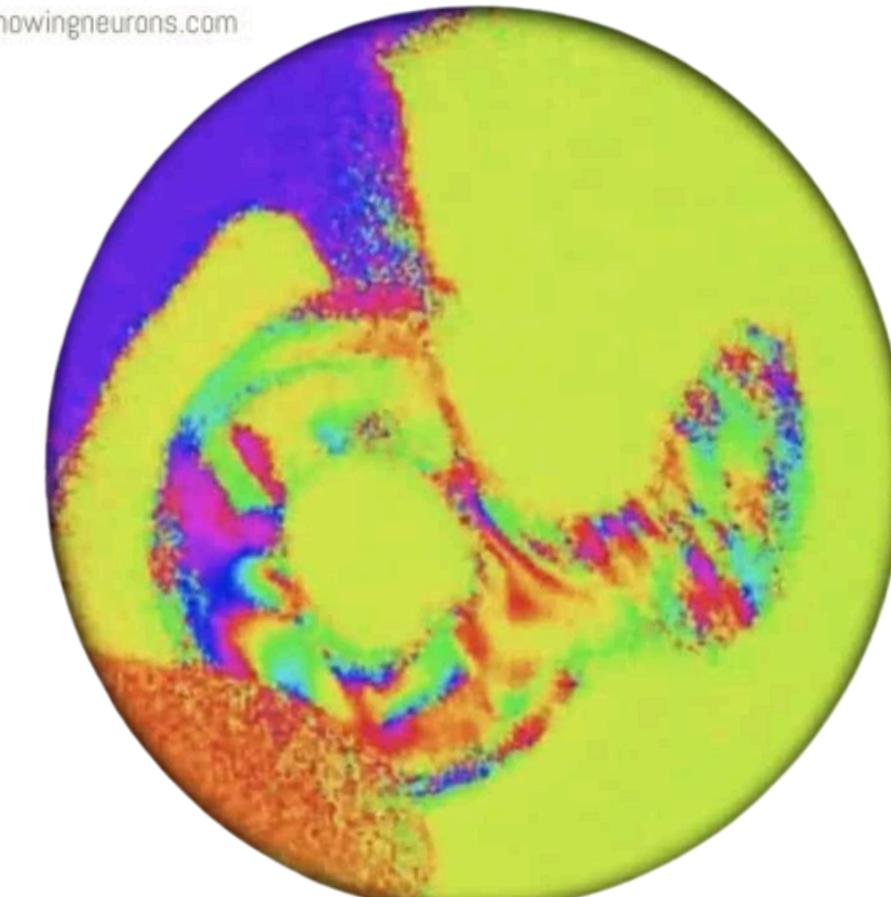
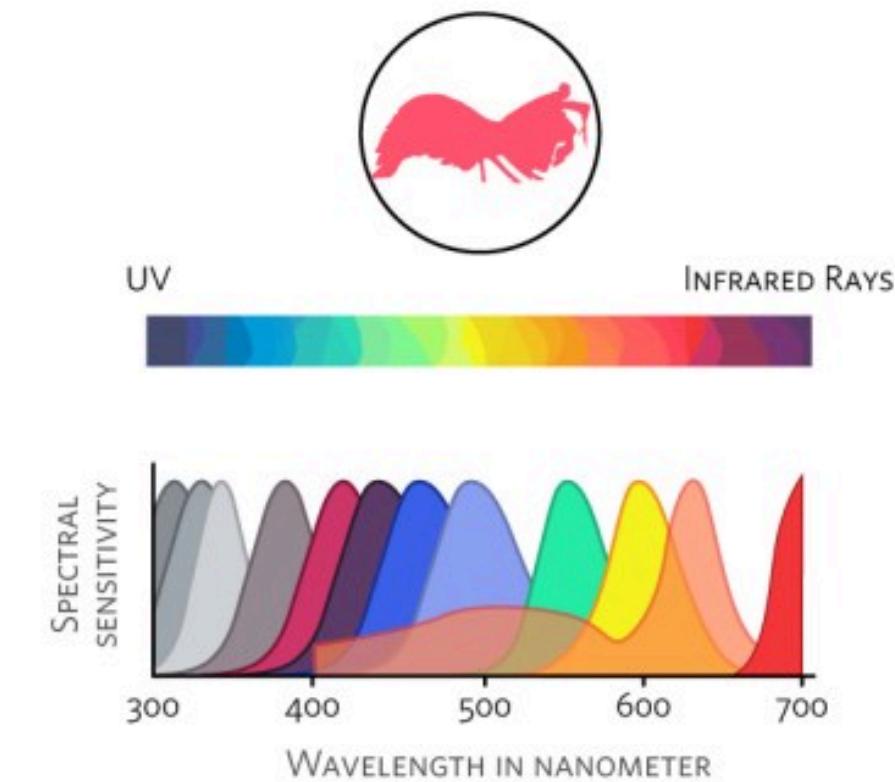
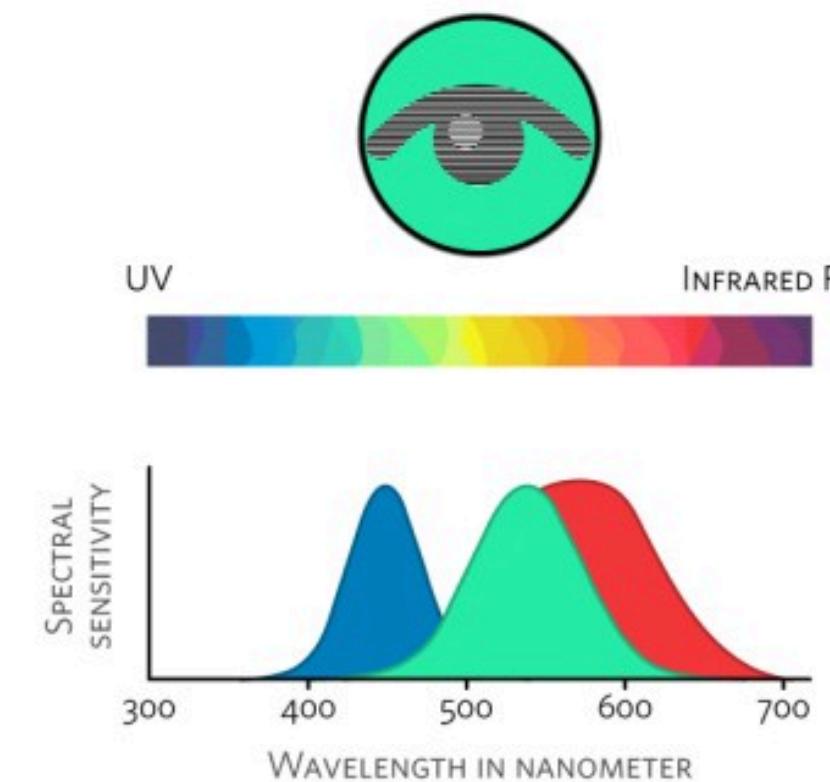
DeepLabv3

Estrutura de Imagens



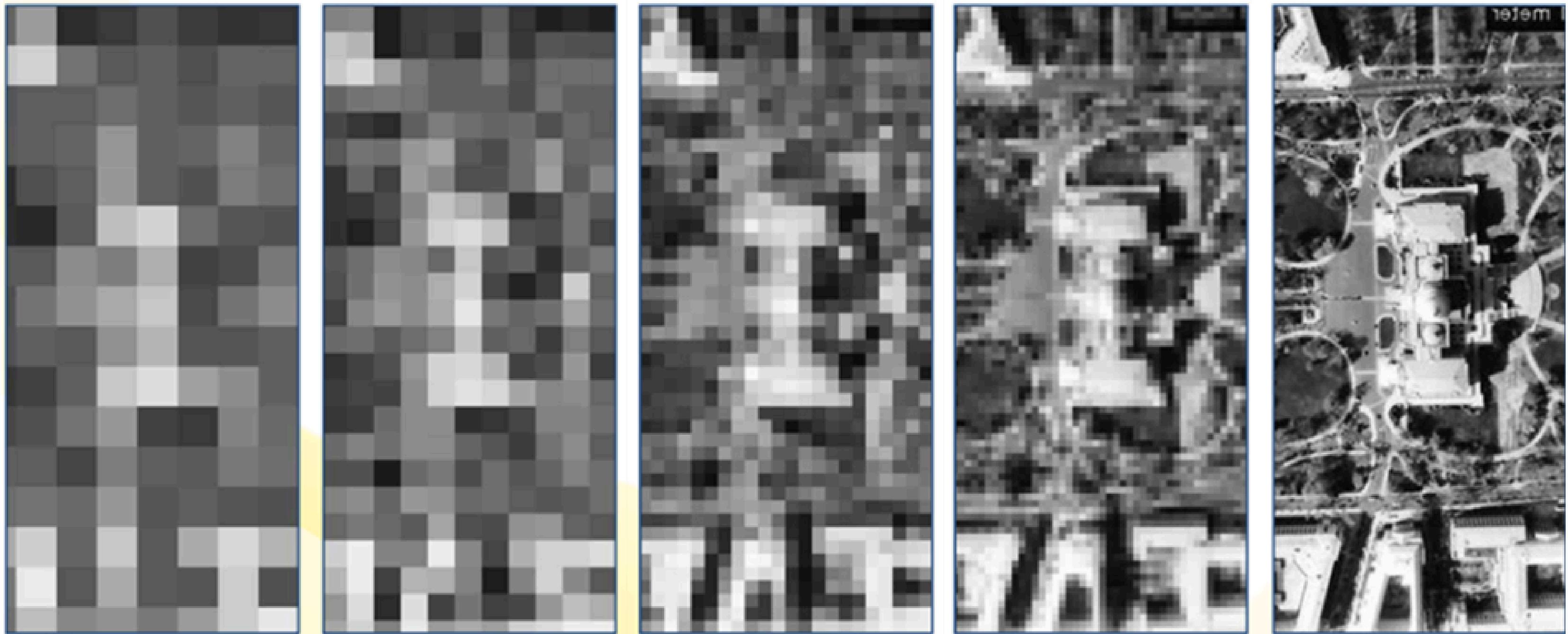
Estrutura de Imagens

Espectro visível de um humano



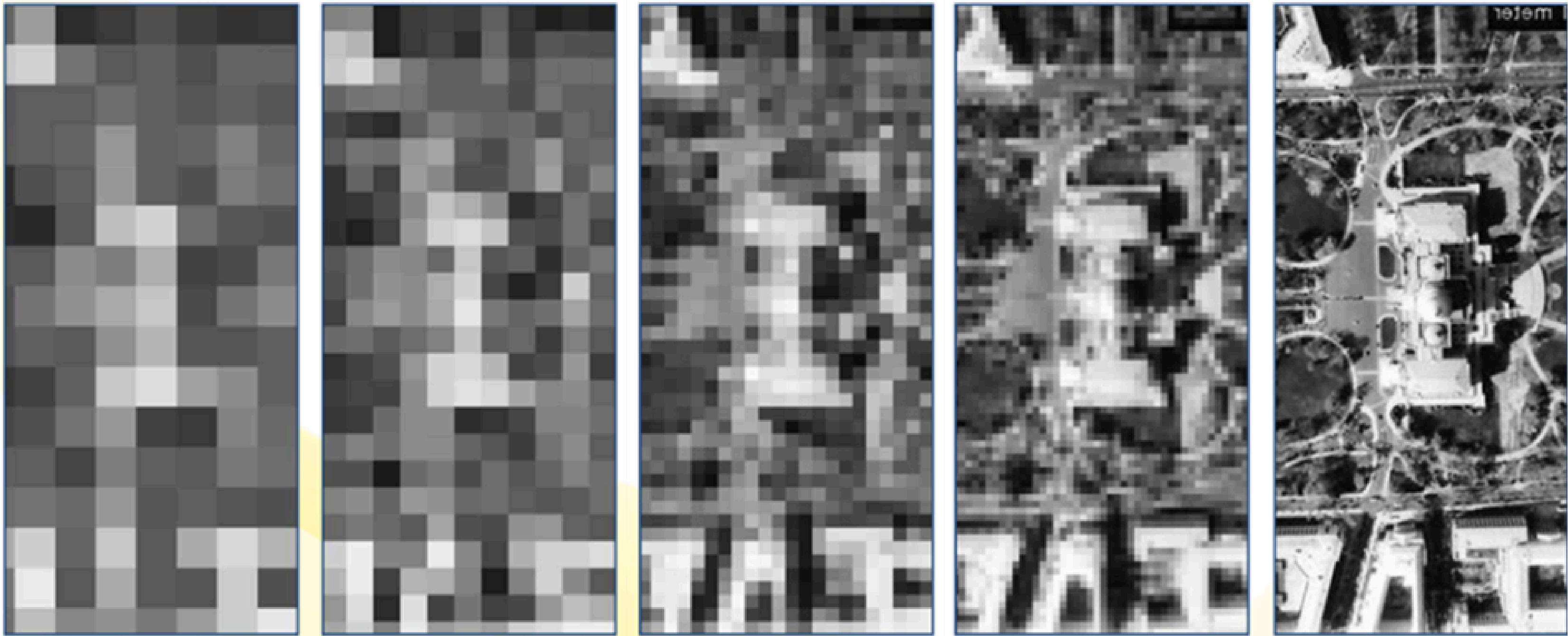
Espectro visível de um camarão

Estrutura de Imagens



Estrutura de Imagens

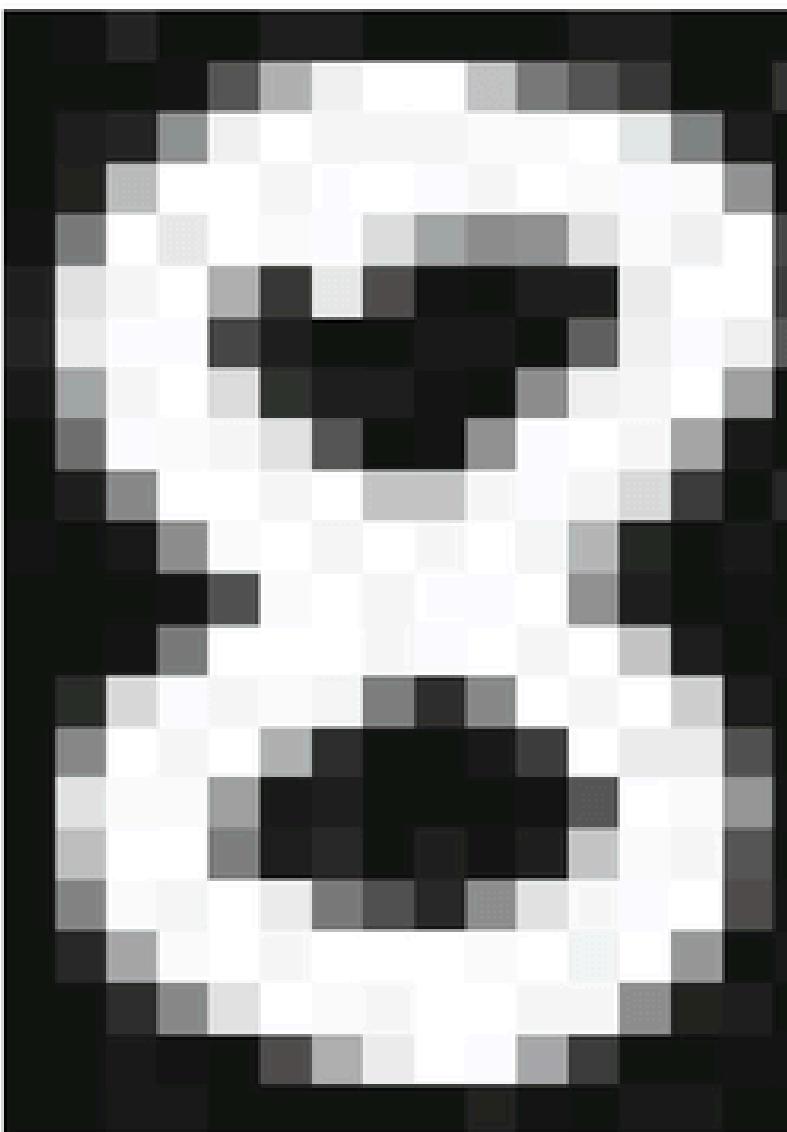
- Imagem Digital - “Toda imagem digital pode ser decomposta em elementos fundamentais como pixels”



Resolução/Quantidade de pixels

Estrutura de Imagens

- Imagem Digital - “Toda imagem digital pode ser decomposta em elementos fundamentais como pixels”
- Uma imagem digital é uma **matriz “bidimensional” de pixels**, onde cada pixel contém informações de cor e intensidade



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0	
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	248	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	14	1	0	6	6	0	0	0	

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0	
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	248	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	14	1	0	6	6	0	0	0	

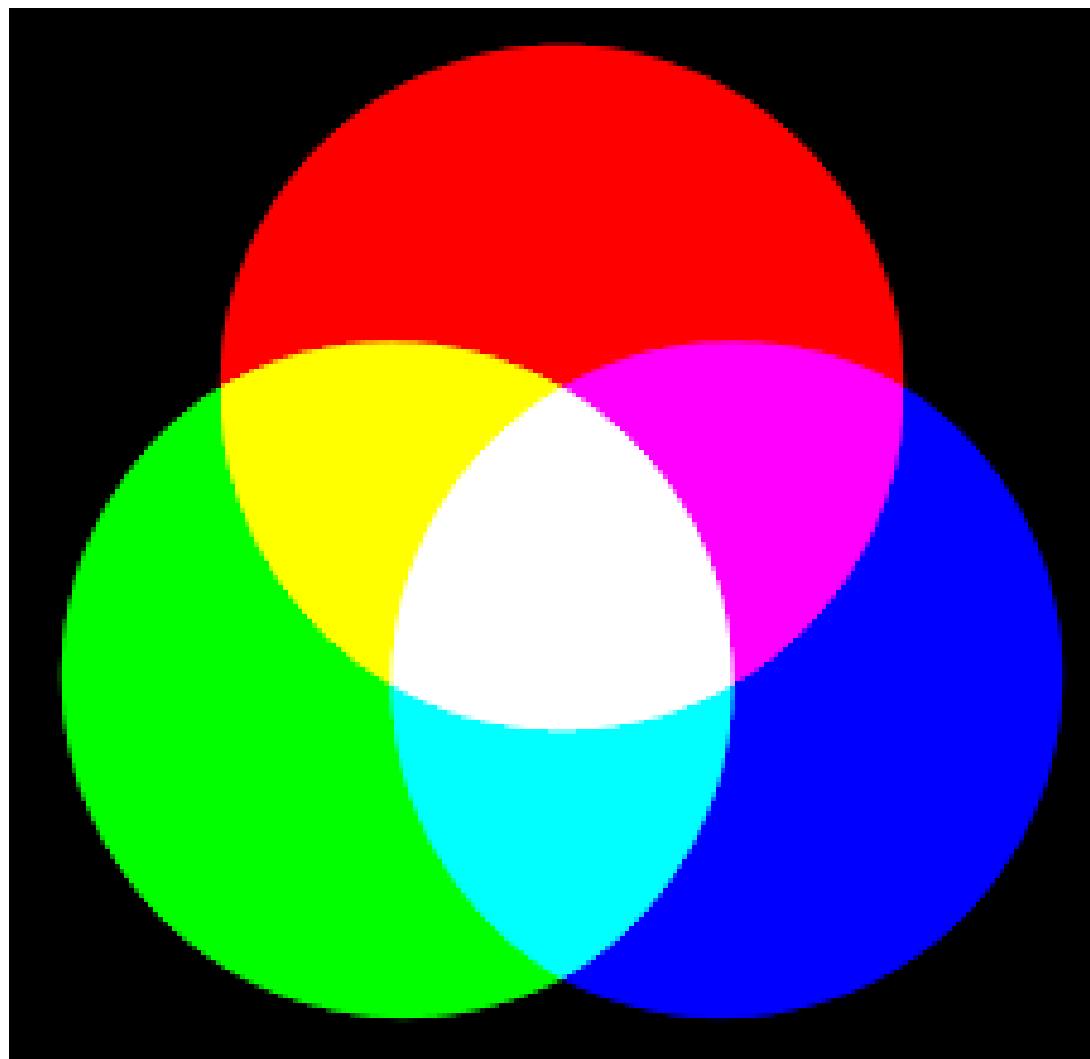
Estrutura de Imagens

- Pixel - é a **menor unidade** de uma imagem digital, representando um ponto de cor.
- Cada pixel carrega valores do **espaço de cor** utilizado na imagem, ao lado temos o RGB.

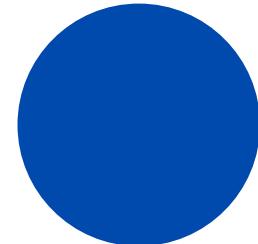


Estrutura de Imagens

- Pixel - é a **menor unidade** de uma imagem digital, representando um ponto de cor.
- Cada pixel carrega valores do **espaço de cor** utilizado na imagem.



- ((0-255), (0-255), (0-255))
- R G B
- Código hexadecimal - representação
 - (0,74,173)
 - #004AAD



Estrutura de Imagens

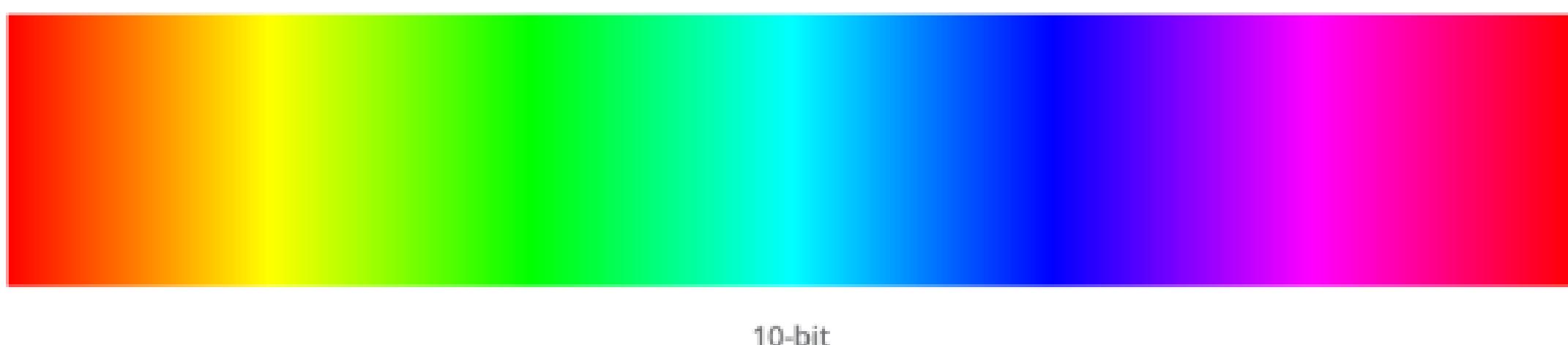
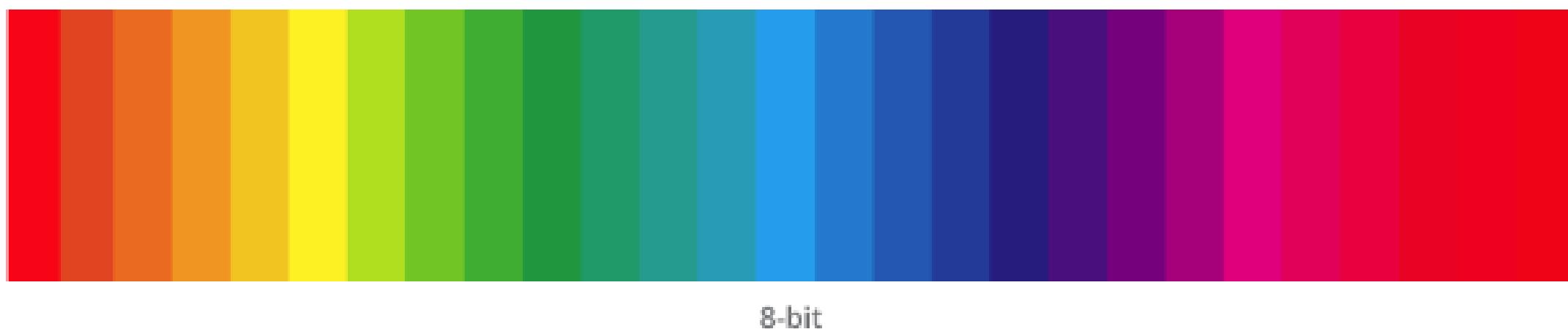
- Profundidade do pixel - a **quantidade de código binário** gerado pelo sensor é a que vai determinar a variação de tonalidades que a imagem tem.

		Binary		
code		R	G	B
0	0	0	0	0
1	0	0	1	blue
2	0	1	0	green
3	0	1	1	cyan
4	1	0	0	red
5	1	0	1	magenta
6	1	1	0	yellow
7	1	1	1	white

- Imagina que posso representar a cor com apenas com código binário, ou seja, tem ou não tem a cor.
- Isso me limita em apenas 8 cores disponíveis 2^3 .
- Se eu aumento essa representação como por exemplo 3 números para cada canal de cor (R, G ou B) agora eu tenho um total de 8 tons de cores para cada canal. Se são 4 números agora serão 16 tons de cor por canal.
- Entendeu?

Estrutura de Imagens

- Profundidade do pixel - a **quantidade de código binário** gerado pelo sensor é a que vai determinar a variação de tonalidades que a imagem tem.
- Quantos mais bits eu tenho de representação de um canal mais tons de cores eu tenho para esse canal.



Bits Por Pixel	Número de Cores Disponíveis
1	2
2	4
4	16
8	256
16	65536
24	16777216
32	16777216 + Transparência
48	281 Trilhões

R G B
256 x 256 x 256 = 16MM

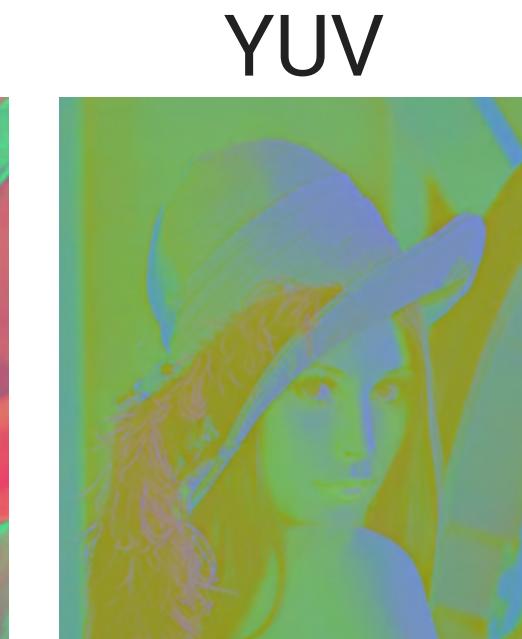
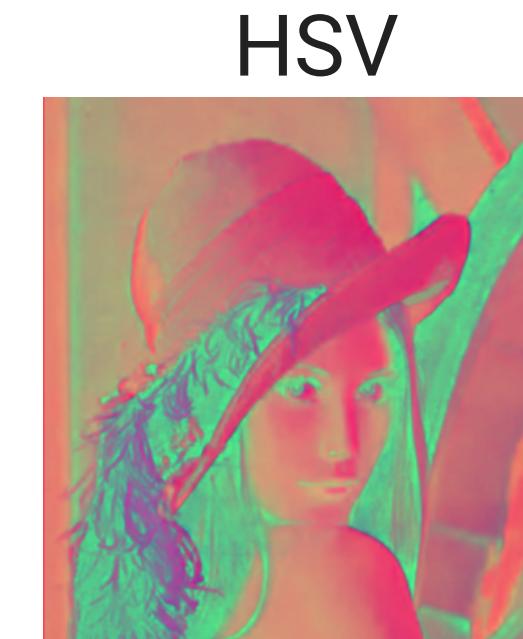
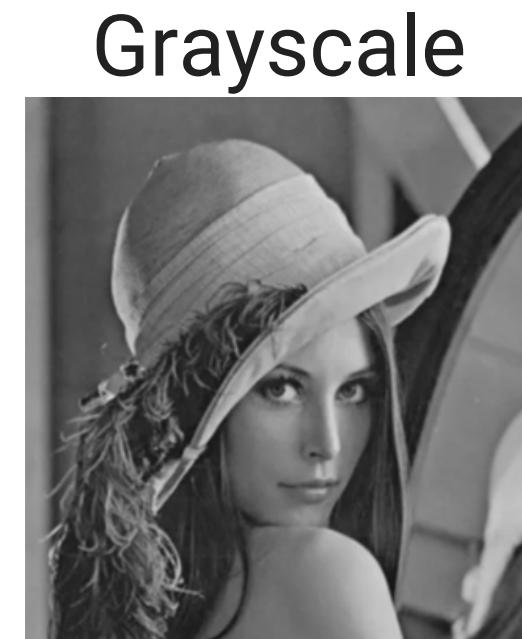
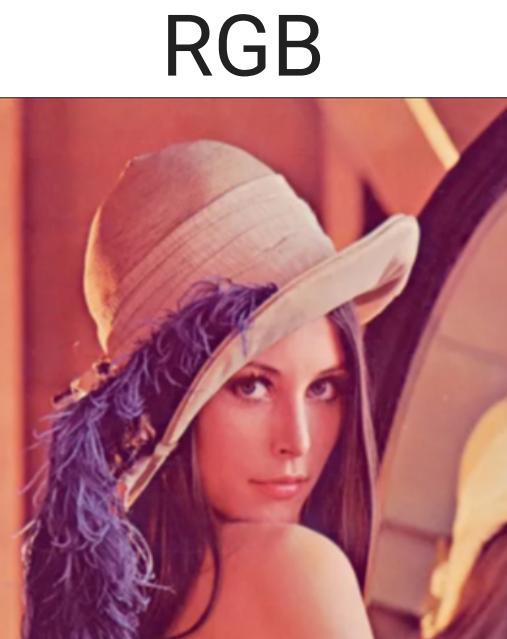
Estrutura de Imagens

- Resolução - é número de pixels em uma imagem ($N \times M$). Quanto **maior a resolução, maior a quantidade** de detalhes que podem ser capturados



Espaços de Cor

- O que é? - uma **maneira matemática de representar** as cores em uma imagem digital.
- Descrever como as cores são vistas ou capturadas por dispositivos (como câmeras) e reproduzidas em telas.
- Para que servem? Manipulação (qualidade), detecção, transformação (filtros). Cada tipo de espaço de cor possui propriedades que o tornam mais adequado para determinadas tarefas.



Formatos de Imagem

Existem diversos formatos de imagens digitais usadas para armazenar e processar arquivos visuais, cada um com suas próprias características, vantagens e desvantagens:

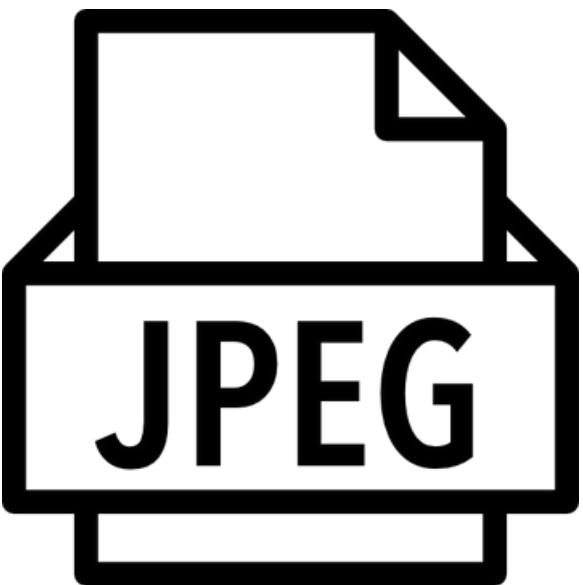


Vantagens do PNG:

- Compressão sem perdas, mantendo a qualidade da imagem.
- Suporte a transparências (canal alpha).

Desvantagens:

- Arquivos maiores e menos eficientes em comparação a formatos como JPEG.



Vantagens do JPEG:

- Alta taxa de compressão (arquivos menores)
- Amplamente suportado por plataformas e dispositivos
- Ideal para armazenar fotografias, onde a perda de alguns detalhes pode não ser perceptível.

Desvantagens:

- Compressão com perdas
- Não suporta transparências.



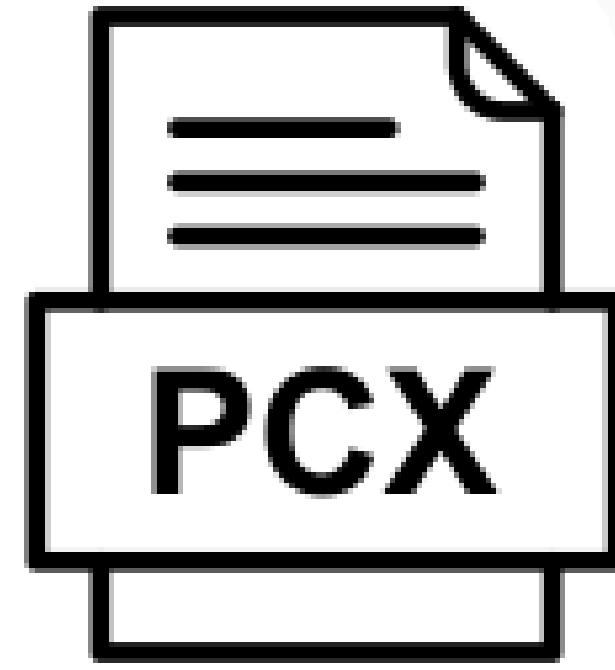
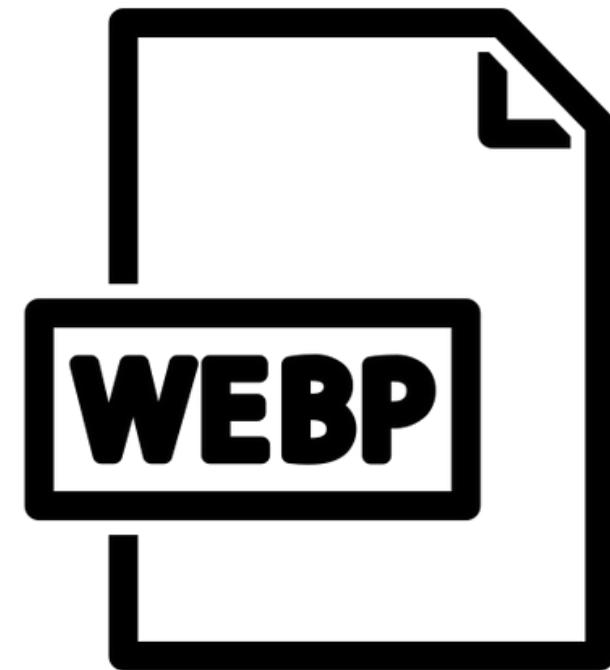
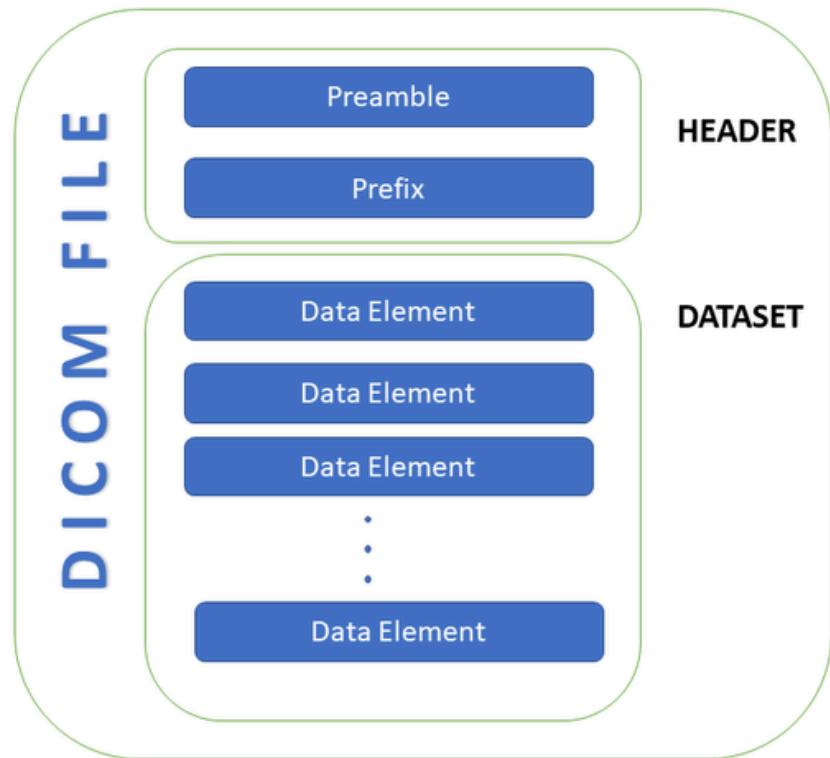
Vantagens do BMP:

- Qualidade máxima da imagem, sem perda de dados.
- Simplicidade de formato, facilitando o processamento em alguns sistemas.

Desvantagens:

- Arquivos muito grandes devido à falta de compressão. Bastante ruim para Big-Data

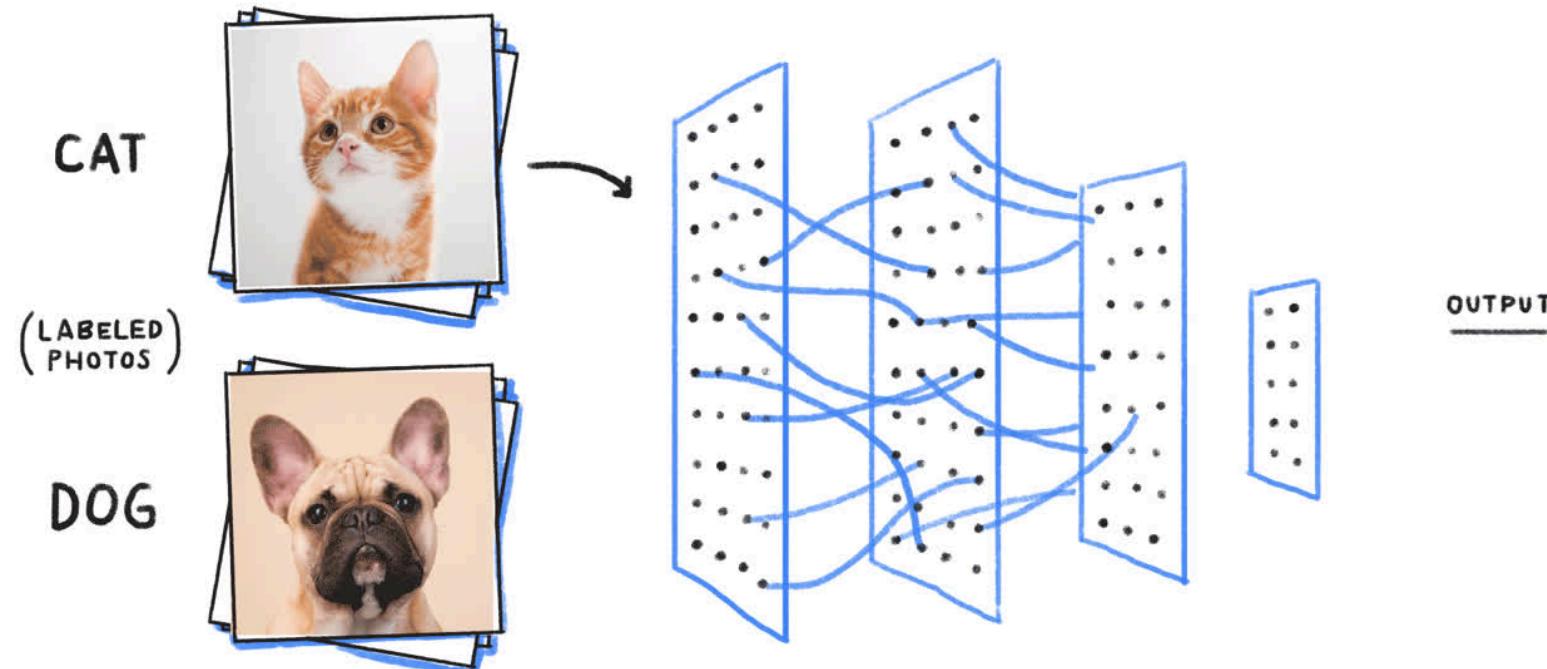
Formatos de Imagem - Uma infinidade



Aplicações de Visão Computacional

Detecção de Objetos

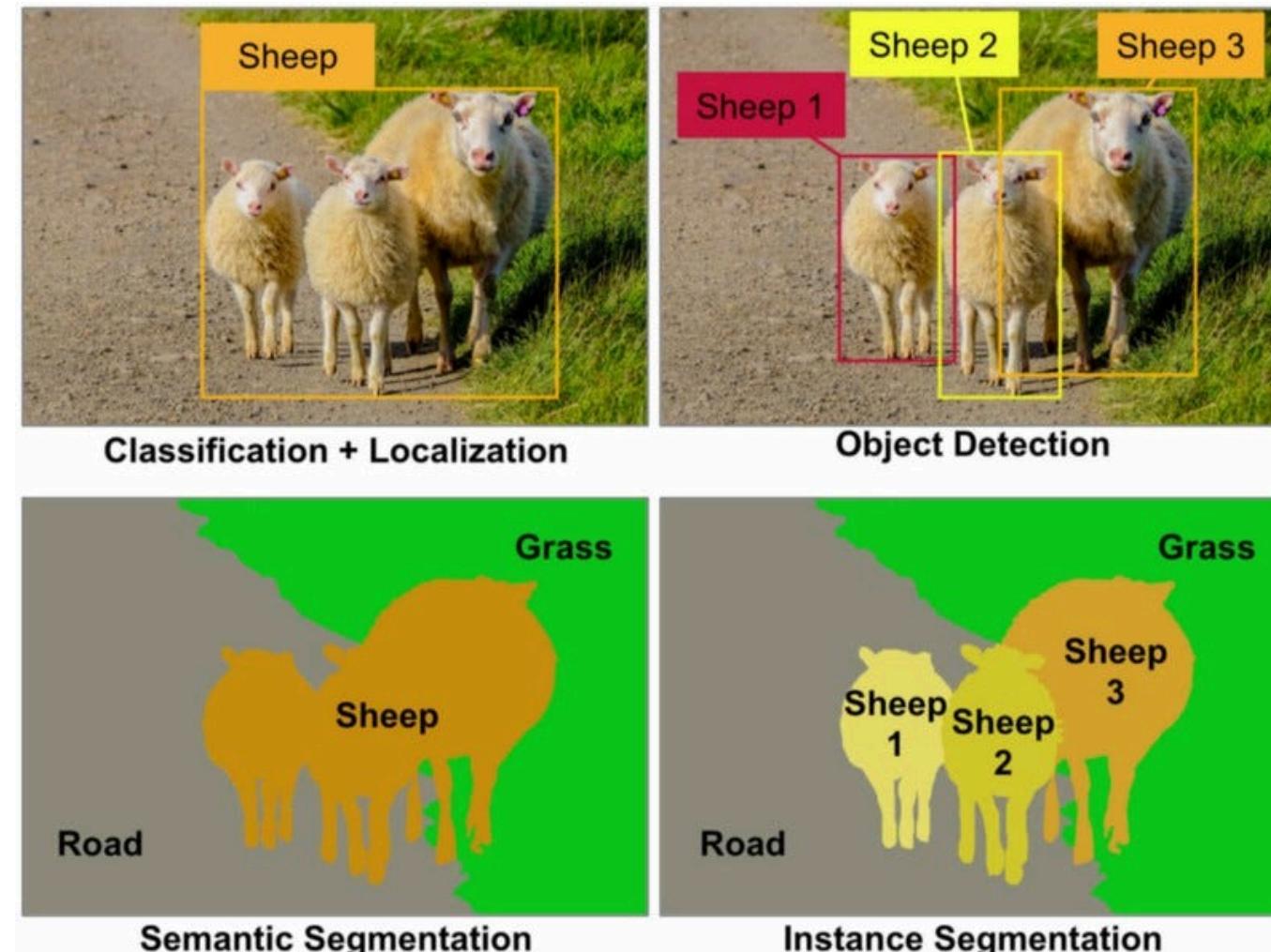
- O que é? Identificação e localização de instâncias de objetos pertencentes a categorias específicas em imagens.
 - Mescla classificação com localização



Aplicações de Visão Computacional

Segmentação de Imagem

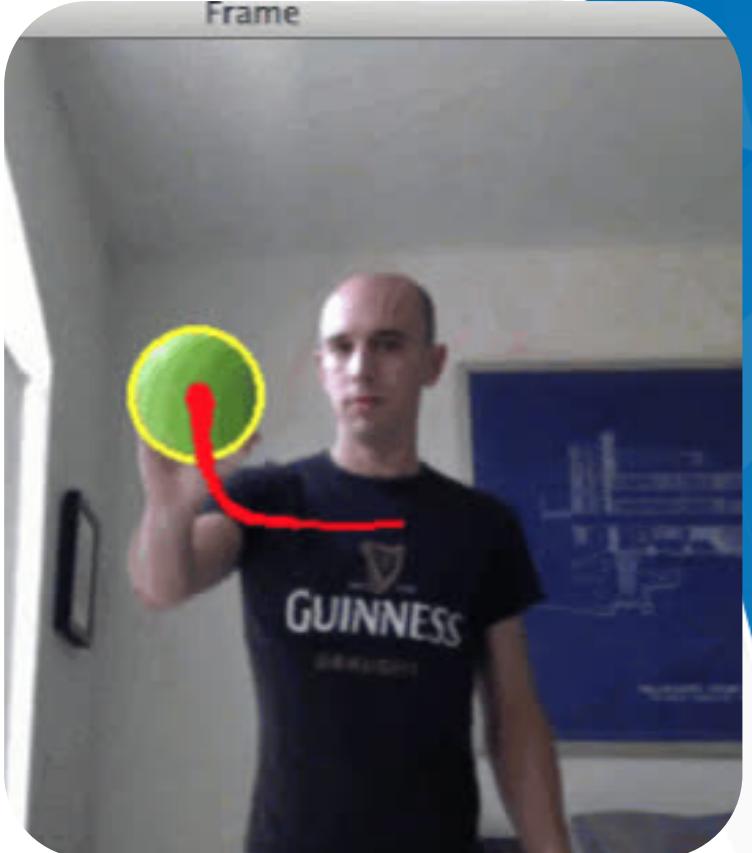
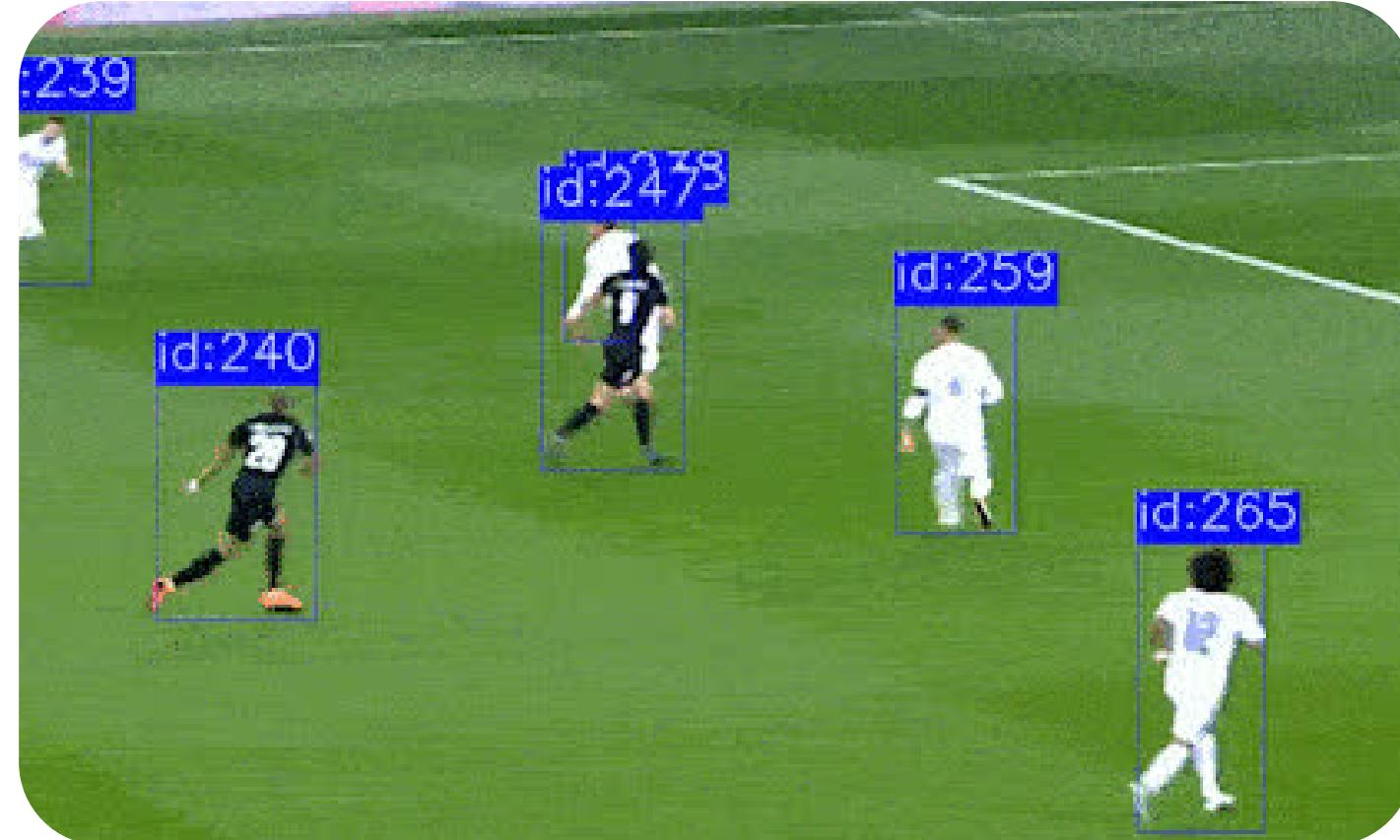
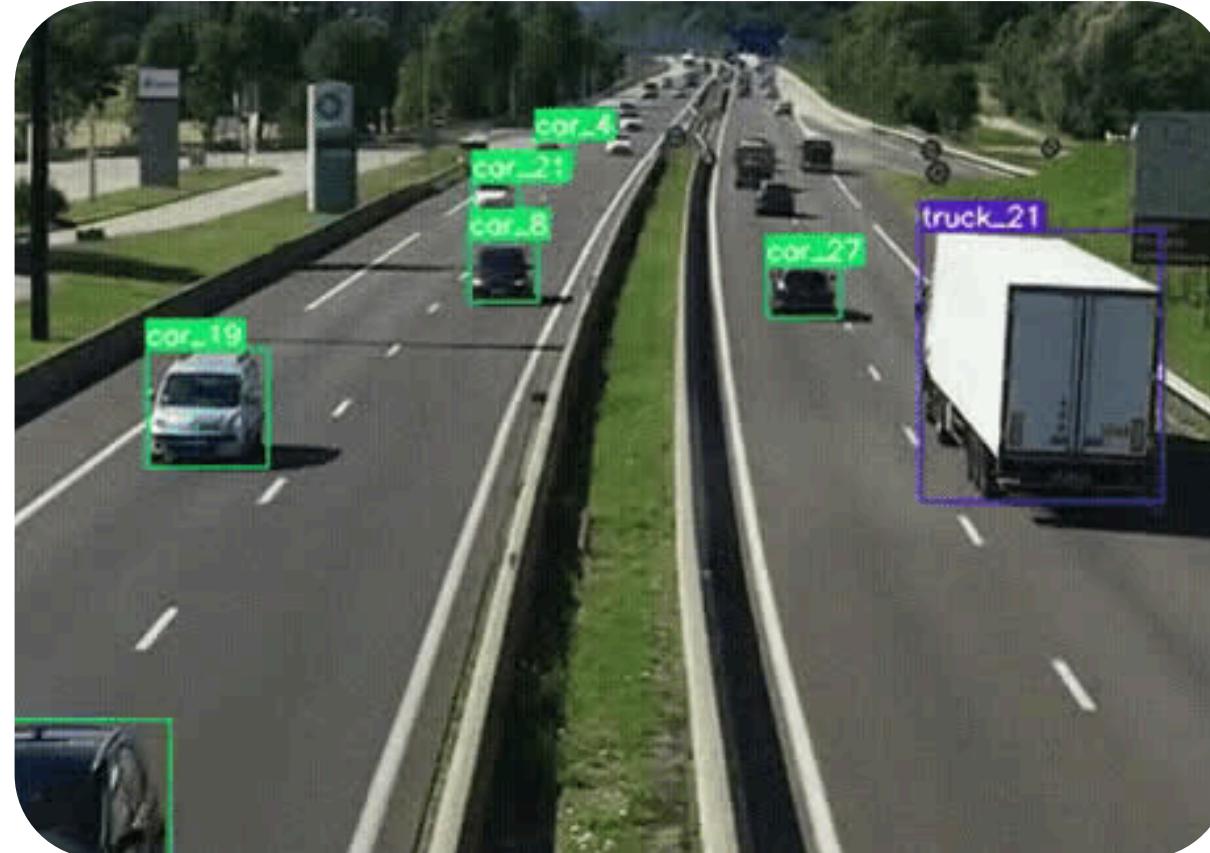
- **O que é?** Aplicação avançada de visão computacional que vai além da detecção de objetos. Seu objetivo é dividir uma imagem em várias partes ou regiões, associando cada pixel a uma classe específica.



Aplicações de Visão Computacional

Rastreamento/Tracking

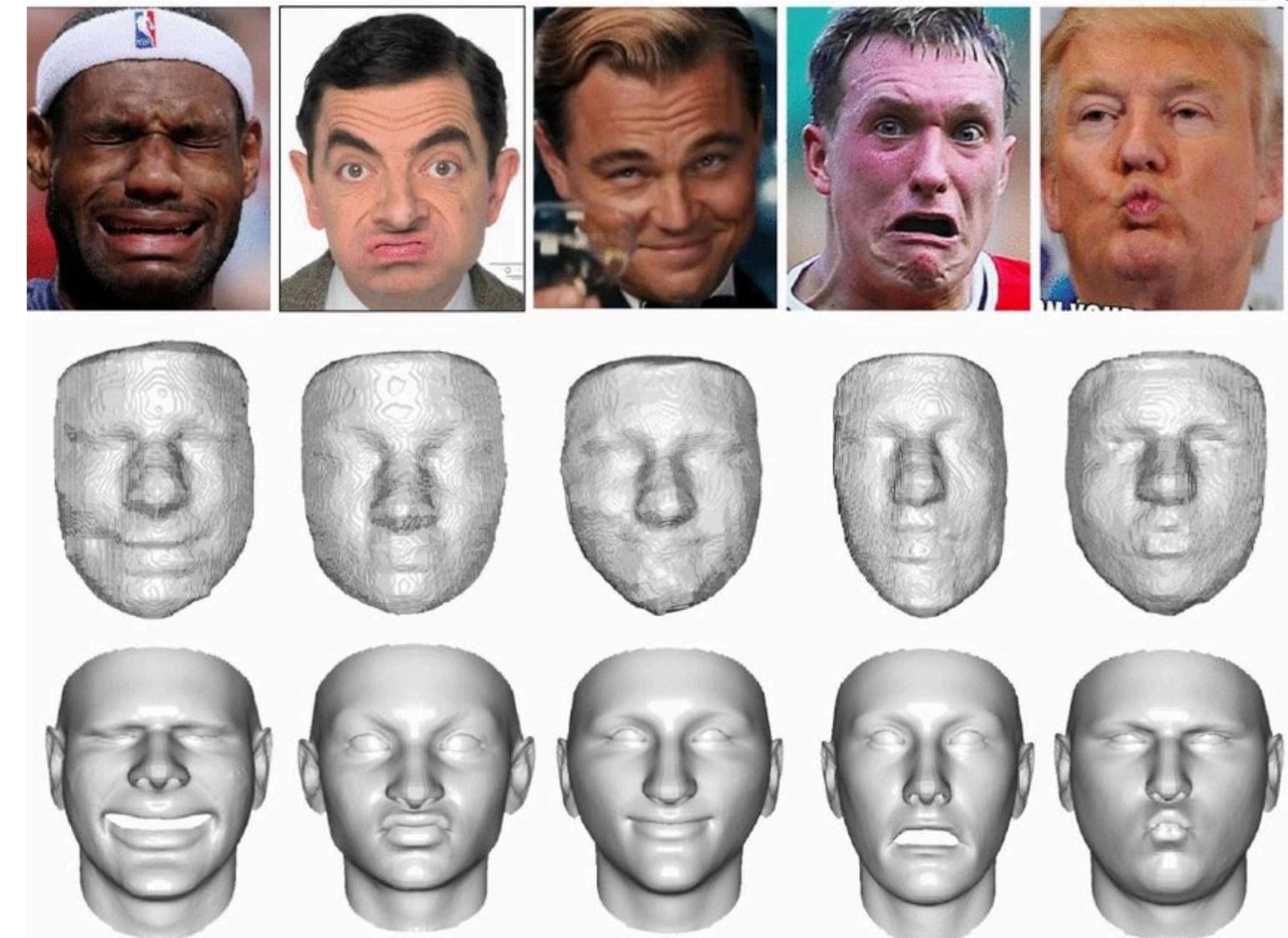
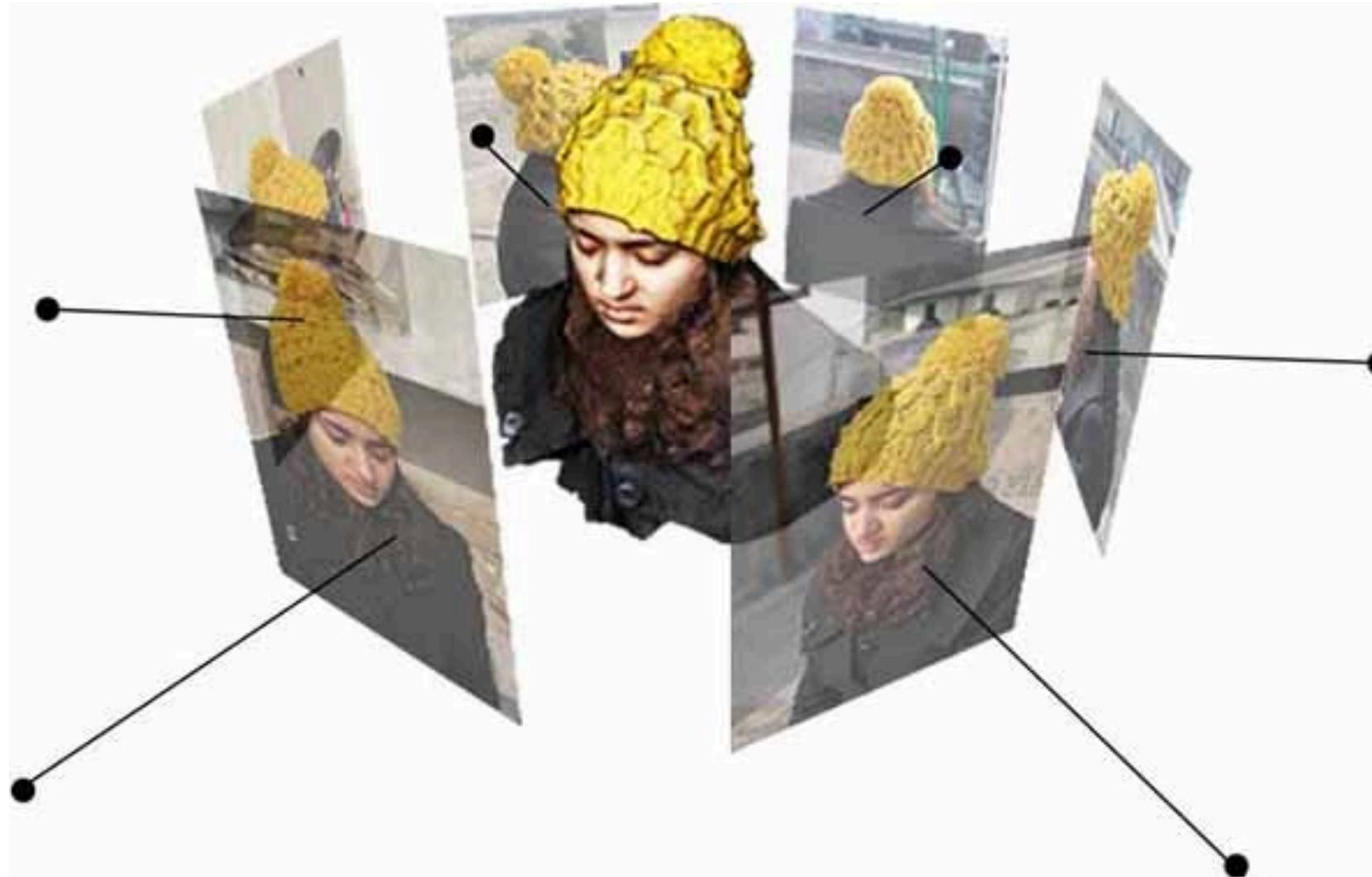
- O que é? Seguir o movimento de objetos em uma sequência de vídeo.



Aplicações de Visão Computacional

Reconstrução 3D

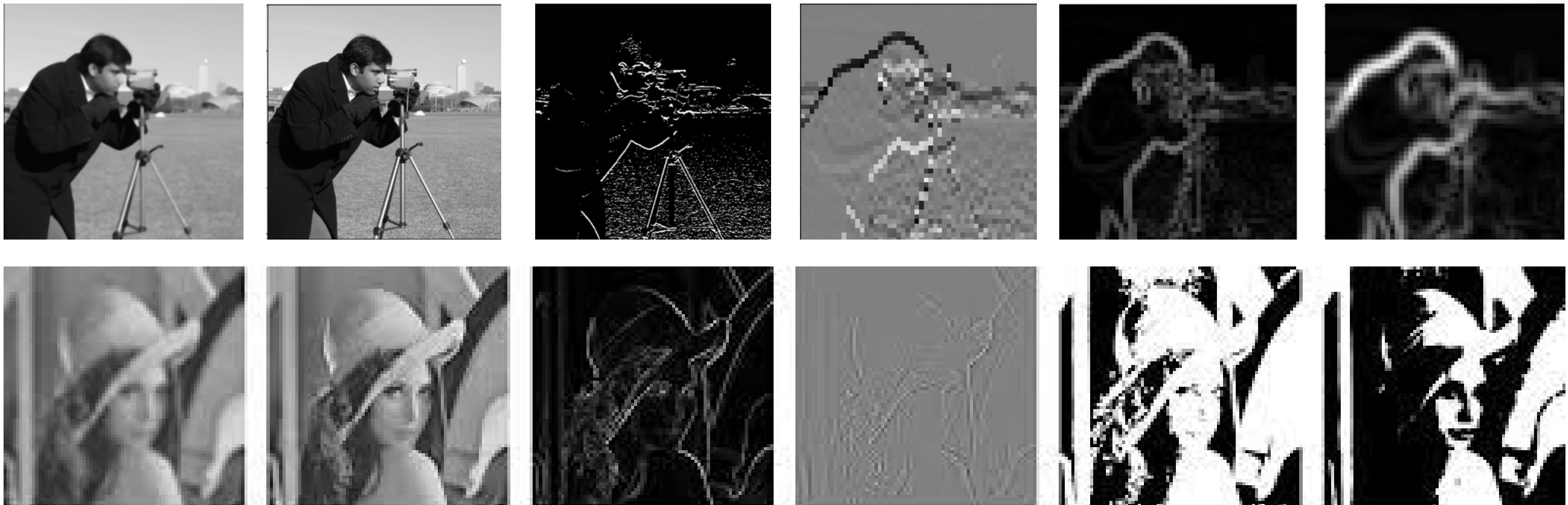
- O que é? Criar modelos 3D a partir de imagens 2D ou sequências de vídeo



Aplicações de Visão Computacional

Filtragem e Melhoramento de Imagem

- O que é? Aplicação de filtros para realçar ou modificar imagens.



Blur

Median

Edge-Detect

High-Pass

Dilate

Erode

Obrigado!

Dúvidas ou sugestões:

davijesus@discente.ufg.br

hugorodriguespessoni@discente.ufg.br

INF
INSTITUTO DE
INFORMÁTICA



Referências.

- https://scikit-image.org skimage-tutorials/lectures/1_image_filters.html
- <https://archive.nptel.ac.in/courses/106/106/106106224/>
- <https://blog.frame.io/2020/02/03/color-spaces-101/>
- <https://medium.com/@livajorge7/the-science-of-color-understanding-color-spaces-in-image-processing-d0e238872a0c>
- <https://www.cambridgeincolour.com/pt-br/tutoriais/bit-depth.htm>
- <https://www.avmakers.com.br/blog/o-que-e-profundidade-de-bits>
- <https://www.engesat.com.br/resolucao-o-que-e/>
- <https://archive.nptel.ac.in/courses/106/106/106106224/>
- Livro: Computer Vision: Algorithms and Applications, 2nd ed. © 2022 Richard Szeliski, The University of Washington <https://szeliski.org/Book>