

ELTD1 - Introdução à Modelagem e Aprendizado

11: Classificação e agrupamento de imagens

2026-2

PROF. HEITOR SILVÉRIO LOPES

DEPARTAMENTO DE ELETRÔNICA - DAELN

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR

Mineração de imagens

- ▶ A “Mineração de imagens” está incluída dentro de um conceito mais amplo que é a **Mineração de Multimídia** (imagens, vídeo e áudio)
 - ▶ Definição: é o processo de busca e descoberta de conhecimento implícito de alto nível ou de padrões não explicitamente visíveis em imagens
 - ▶ Entenda-se aqui grandes volumes de imagens.
 - ▶ É uma área de *Machine Learning* que está em grande evidência devido à enorme quantidade de imagens armazenadas e disponíveis na internet



Mineração X Processamento de Imagens

► **Processamento de Imagens:**

- Objetiva fazer o tratamento de imagens através de métodos matemáticos/computacionais com o objetivo de melhorar a qualidade de uma imagem ou reconhecer algum padrão

► **Mineração de imagens:**

- É interdisciplinar, pois utiliza conceitos e métodos de: Bancos de dados, Estatística, *Machine Learning*, Reconhecimento de padrões, e Inteligência computacional.
- Também pode utilizar os métodos tradicionais de Processamento de Imagens para o pré-processamento inicial dos dados visuais
- Segue os mesmos passos do processo de mineração de dados usual, exceto pelo fato de transformar as imagens em conjuntos de atributos (*features*) numéricos



Tarefas aplicáveis ao processamento de imagens

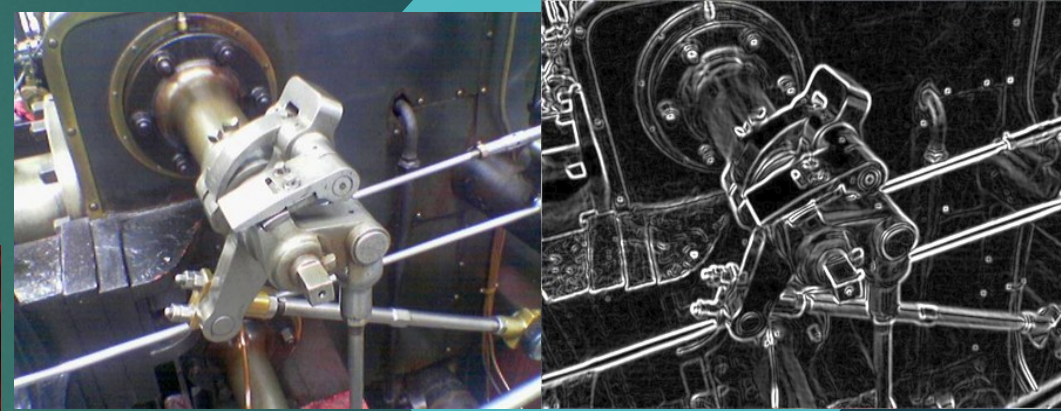
IMPORTANTE

1. Pré-processamento: filtragem/transformação
2. Extração/seleção de características (descritores numéricos ou *features*)
3. Classificação e/ou agrupamento de imagens
4. Busca reversa
5. Recuperação multimodal de imagens (*Multimodal Image Retrieval*)

1- Pré-processamento: filtros de imagens

- ▶ Objetiva **evidenciar/accentuar** alguma característica importante das imagens

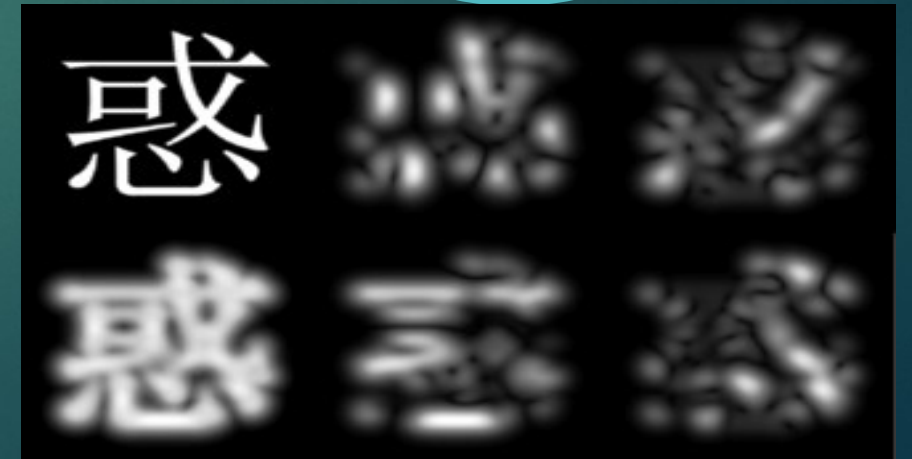
- ▶ Filtro de bordas (Canny) →



- ▶ Filtros de cor:

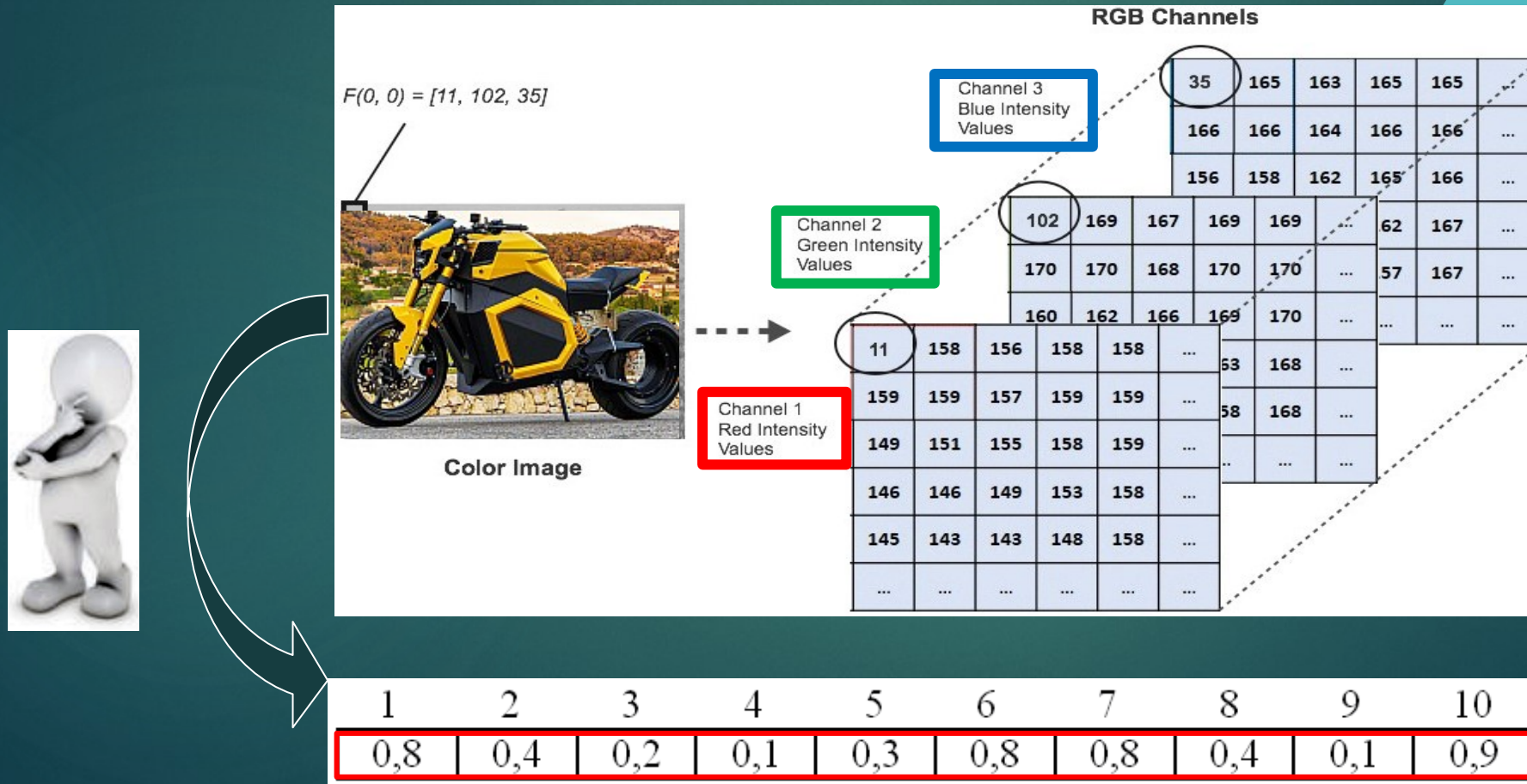


- ▶ Filtros de direção: (Sobel) →



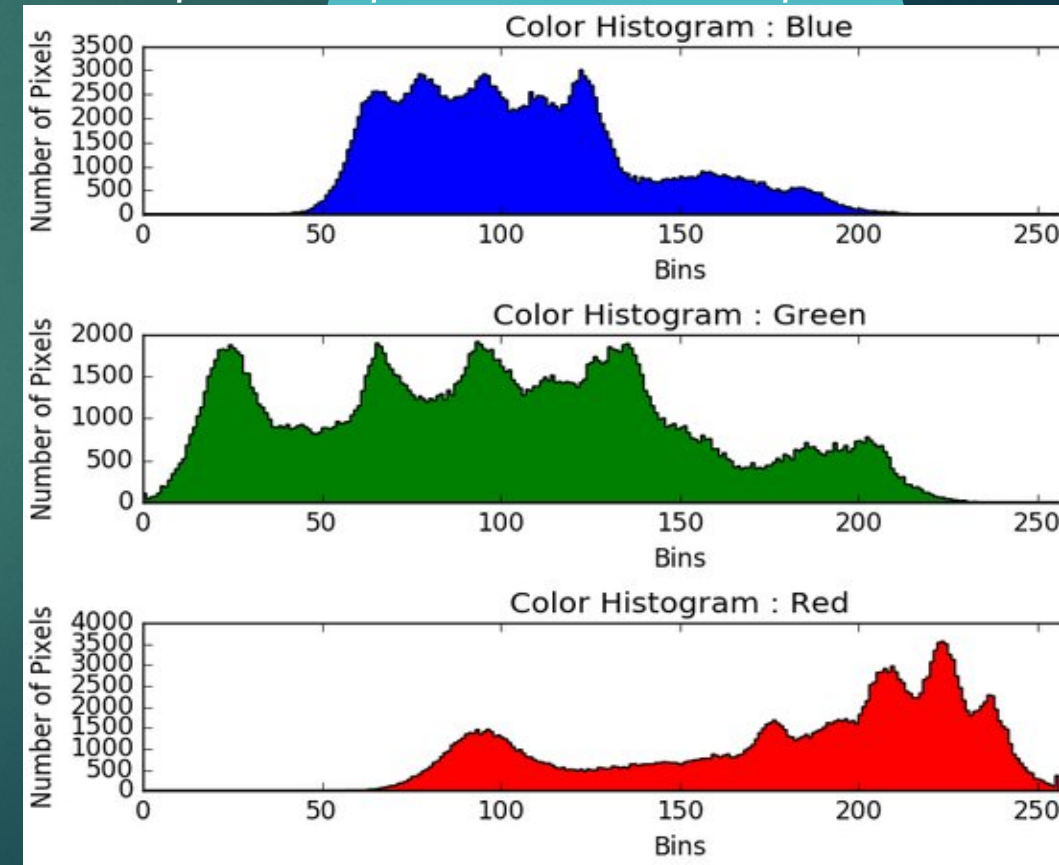
2- Extração de *features*

- ▶ É a simples transformação de imagens em vetores numéricos, realizada por métodos matemáticos



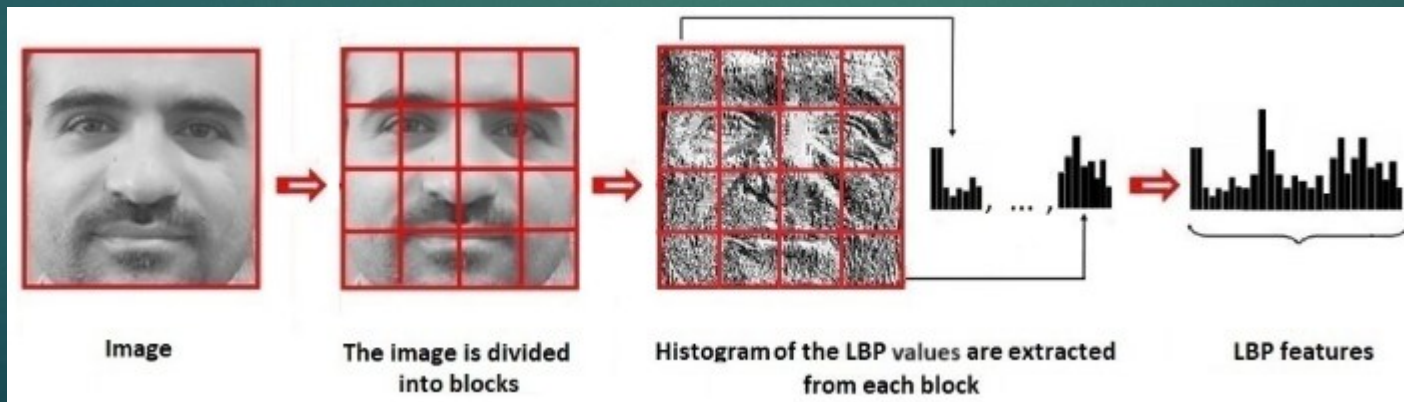
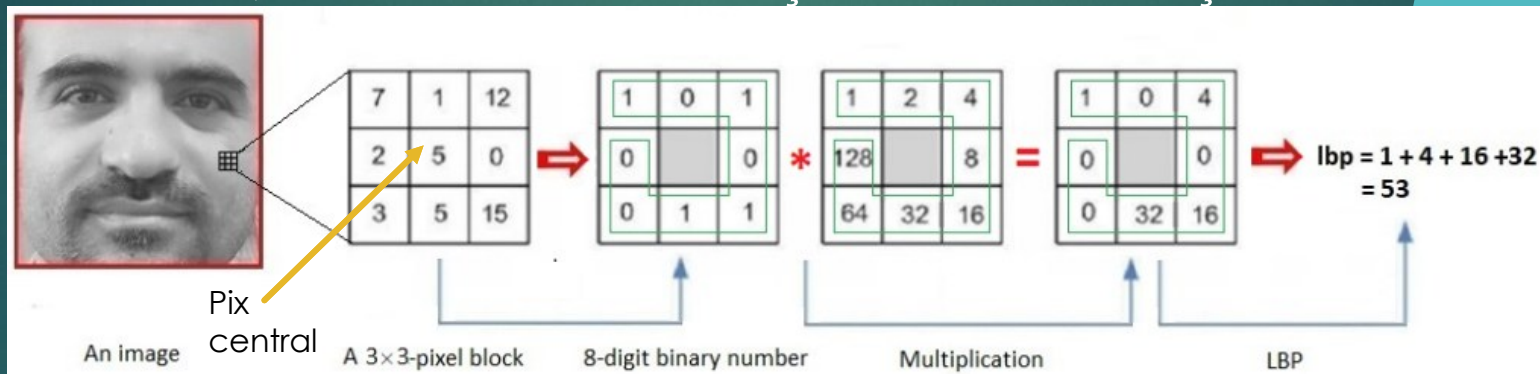
2- Extração de *features*

- ▶ **Descritores de cor:** são invariantes em relação à escala, translação e rotação da imagem
 - ▶ Histograma RGB: é uma combinação dos histogramas de cor dos canais *Red*, *Green* e *Blue*. Cada ponto é o número de pixels que contém aquela intensidade da cor, para cada canal
 - ▶ Outros descritores de cor menos usuais: *Color moments*, *color coherent vector*, *color SIFT descriptors*, etc



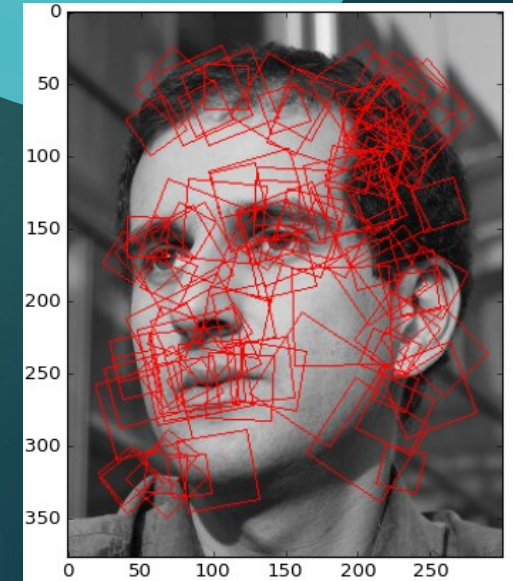
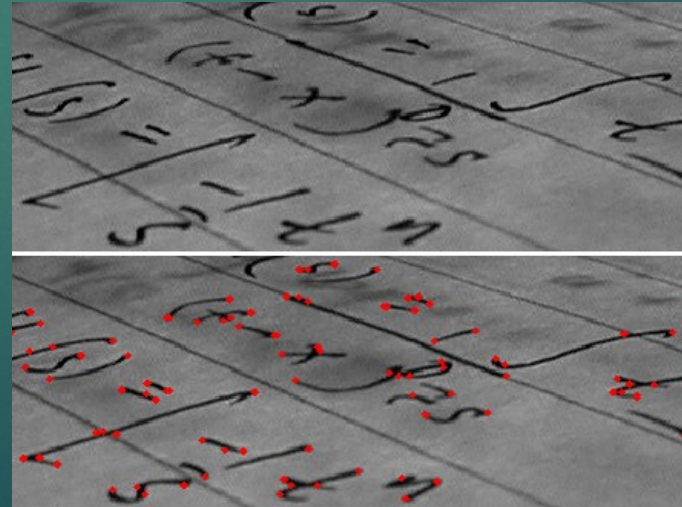
2- Extração de *features*

- ▶ **Descritores de textura:** medem a suavidade e regularidade da imagem
- ▶ LBP (*Local Binary Patterns*): é muito popular por ser computacionalmente eficiente, robusto a mudanças de iluminação



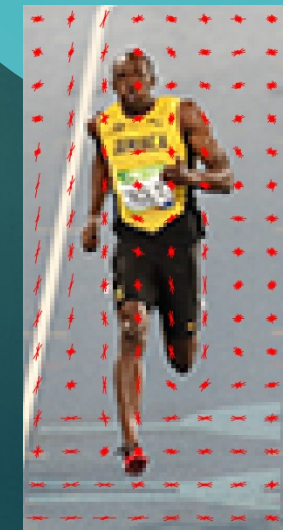
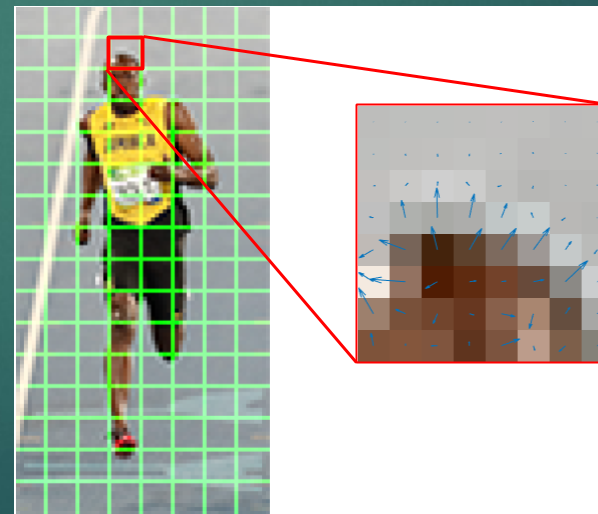
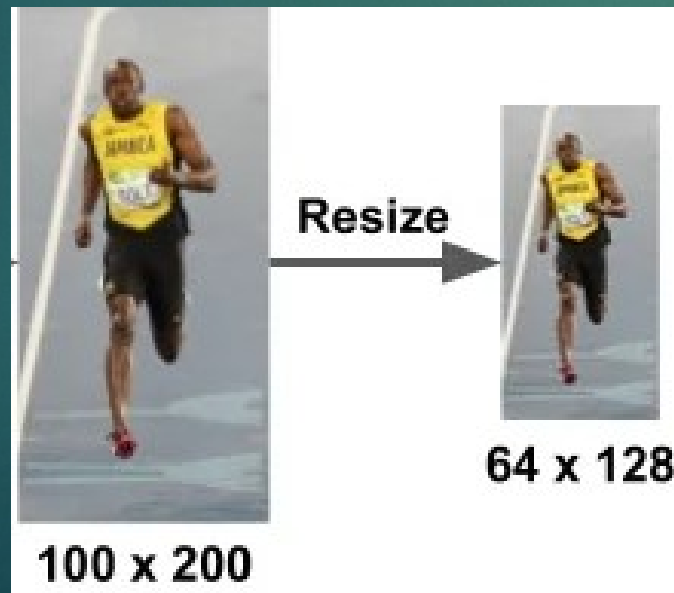
2- Extração de *features*

- ▶ **Descritores no domínio da frequência:** tem baixo custo computacional e são baseados na detecção de pontos de interesse na imagem:
 - ▶ **SIFT** (*Scale Invariant Feature Transform*) é robusto a variações de iluminação e pequenas variações posicionais, e invariante a rotação e escala
 - ▶ **SURF** (*Speed-Up Robust Features*) é um detetor de pontos de interesse invariante a rotação e escala
 - ▶ **ORB** (*Oriented FAST and Rotated BRIEF*): versão mais rápida do SURF



2- Extração de *features*

- ▶ Histograma de Gradientes Orientados (HOG):
 - ▶ É inspirado no SIFT, sendo baseado na distribuição das orientações dos gradientes (derivadas nos eixos x e y da imagem). Método patenteado em 1986
 - ▶ **Muito utilizado para detecção de pedestres, veículos e animais em imagens**



Algumas bibliotecas em Python

- ▶ Color histogram:

<https://pyimagesearch.com/2021/04/28/opencv-image-histograms-cv2-calchist/>

<https://medium.com/@rndayala/image-histograms-in-opencv-40ee5969a3b7>

- ▶ Local Binary Patterns: <https://www.geeksforgeeks.org/create-local-binary-pattern-of-an-image-using-opencv-python/>

<https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>

- ▶ Gray-Level Co-occurrence Matrix:

<https://github.com/alfianhid/Feature-Extraction-Gray-Level-Co-occurrence-Matrix-GLCM-with-Python>

Algumas bibliotecas em Python

- ▶ SIFT: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html
<https://github.com/rmislam/PythonSIFT>
- ▶ SURF: <https://mahotas.readthedocs.io/en/latest/surf.html>
- ▶ Blob detection: <https://learnopencv.com/blob-detection-using-opencv-python-c/>
- ▶ HOG: <https://www.thepythoncode.com/article/hog-feature-extraction-in-python> http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html
<https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>
- ▶ Sobel e Laplaciano: https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_image_Gradient_Sobel_Laplacian_Derivatives_Edge_Detection.php
- ▶ Canny edge detector: https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html
- ▶ ORB: https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html

3- Tarefas: Classificação e agrupamento de imagens

IMPORTANTE

- ▶ Os mesmos conceitos de **Modelagem Descritiva** e **Modelagem Preditiva** valem para a análise/classificação de imagens
- ▶ Um método (simples) baseado em **descritores de imagem** pode ser útil quando:
 - ▶ As características que diferenciam as imagens são bem específicas (p.ex. cor, forma, etc)
 - ▶ Deseja-se um modelo para compreender (descrever) as características relevantes das imagens;
 - ▶ É requerido um **baixo custo computacional** ←
- ▶ Um método **convolucional/transfer-learning** para extração de *features* pode ser útil quando:
 - ▶ A tarefa é de **alta complexidade semântica**; ←
 - ▶ Se deseja, prioritariamente, alta acurácia, às custas de um alto custo computacional

3- Tarefas: Classificação e agrupamento de imagens

- ▶ Utilizam-se os mesmos métodos usados em bases de dados:
 - ▶ Classificação (**aprendizado supervisionado**): OneR, Árvore de decisão, Redes Neurais, etc...



$f1$	$f2$	$f3$	$f4$	$f5$	...	f_n	classe
0,123	0,946	0,856	0,168	0,02	...	0,431	F



$f1$	$f2$	$f3$	$f4$	$f5$	...	f_n	classe
0,002	0,701	0,287	0,949	0,923	...	0,581	M

- ▶ Agrupamento (**aprendizado não-supervisionado**): K-means, Hierárquico, etc...

4- Tarefa: Busca reversa de imagens

- ▶ É uma forma de CBIR (*Content-Based Image Retrieval*) onde se dá uma imagem de amostra (*query*) e o sistema busca em uma base de imagens aquelas que sejam relacionadas
- ▶ A busca reversa pode ser utilizada para:
 - ▶ Localizar a origem ou o autor da imagem na internet
 - ▶ Encontrar outras versões (melhor resolução, imagens falsificadas, etc)
 - ▶ Descobrir páginas onde a imagem aparece
- ▶ Também se pode utilizar: SIFT (*Scale-Invariant Feature Transform*) para extrair 128 características locais das imagens, ou BoVW - *Bag-of-Visual-Words* (vetor esparsa de ocorrência das características) para representar a imagem

4- Tarefa: Busca reversa de imagens

- ▶ Algoritmo do Google para busca reversa de imagens:
 - ▶ Análise da imagem: extrai características de cor, linhas, texturas, pontos de interesse, etc
 - ▶ Geração da query: cria um vetor esparsa das características (BoVW)
 - ▶ Comparação com o BD de imagens: compara a query com bilhões de imagens já cadastradas no BD
 - ▶ Resultados: retorna as imagens de maior similaridade
- ▶ Sistemas que utilizam busca reversa:
 - ▶ *Google search by image, Yahoo image search, Bing images, Picsearch, Pixsy, Pinterest, TinEye, ...*

<https://images.google.com/>

<https://www.bing.com/images>

<https://www.picsearch.com>

<https://br.images.search.yahoo.com>

5- Tarefa: Recuperação multimodal de imagens

- ▶ *Multimodal Image Retrieval*
- ▶ É uma área relativamente nova de pesquisa
- ▶ Objetiva encontrar, em um banco de dados, recursos de uma modalidade (p.ex. imagem) utilizando uma *query* de outra modalidade (texto, *sketch*, audio, etc)



Text



Image



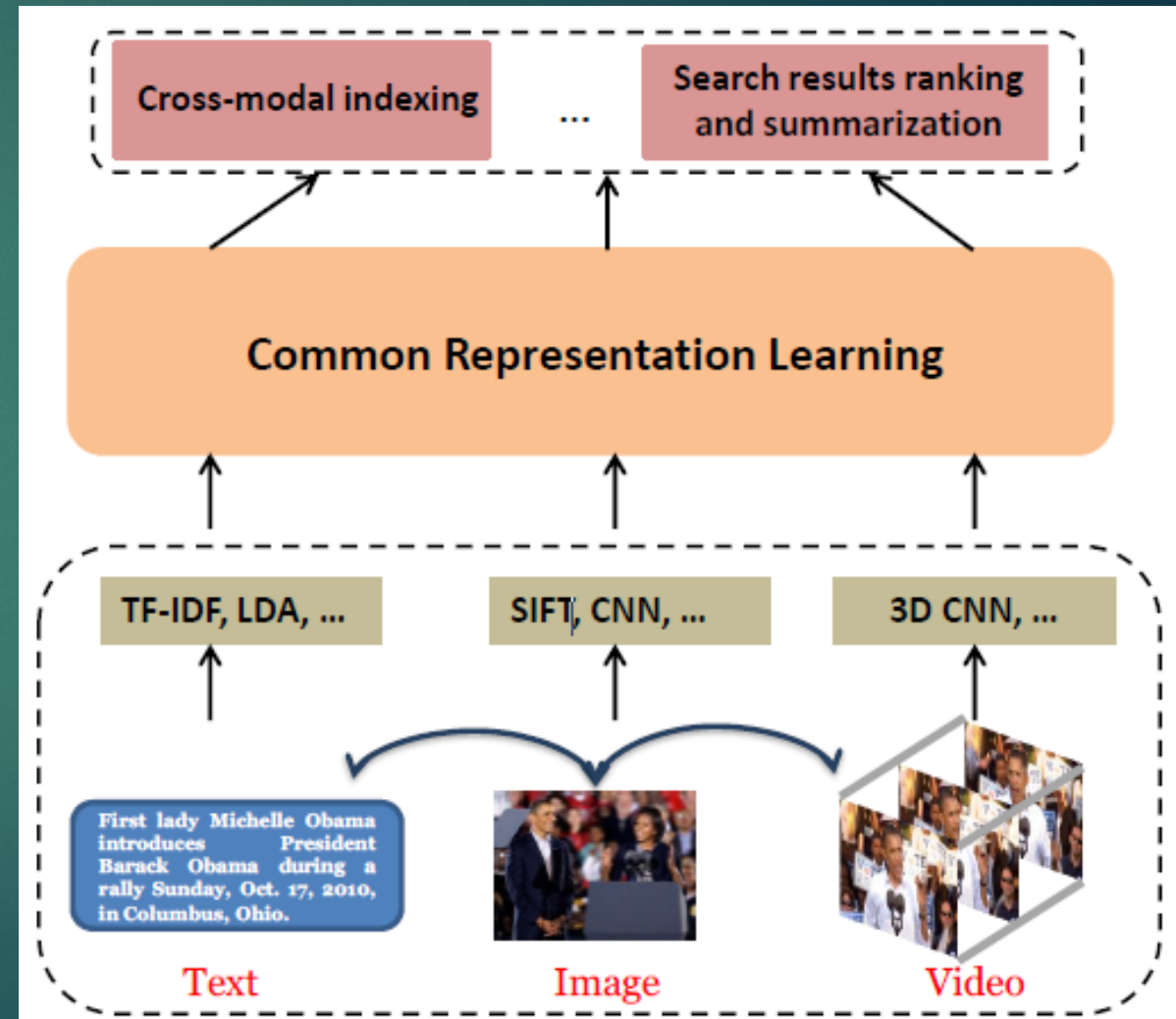
Video



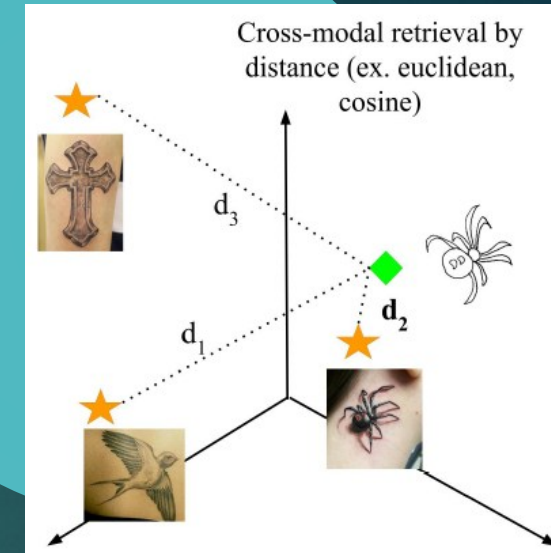
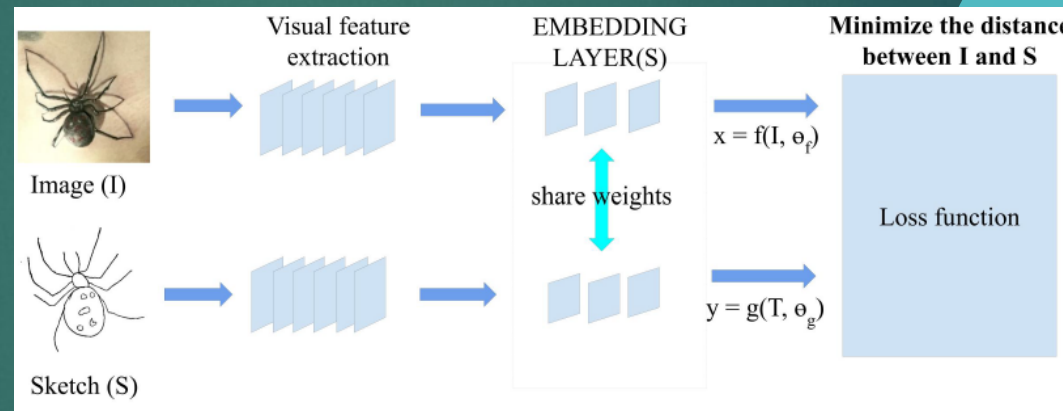
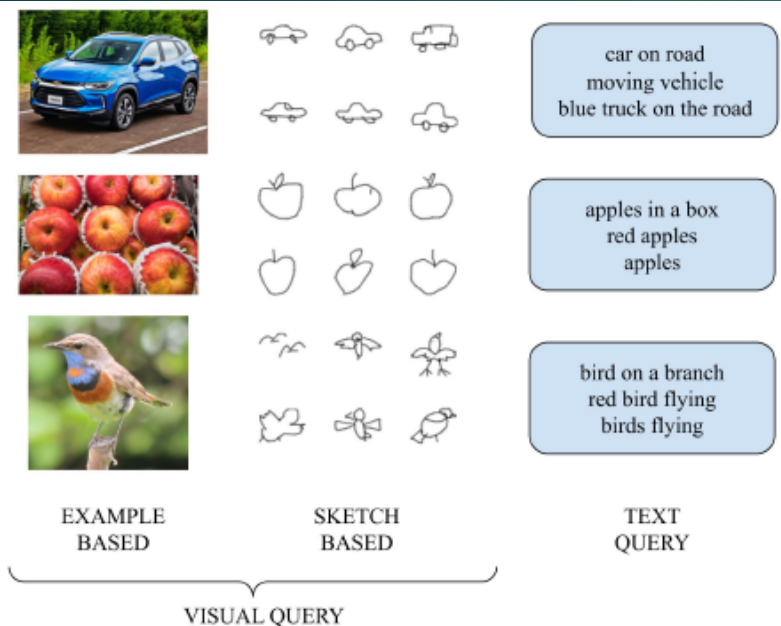
Audio

5- Tarefa: Recuperação multimodal de imagens

- ▶ Diversos descritores (*features*) são extraídos de cada entrada de acordo com a sua modalidade
- ▶ Transforma-se as diferentes *features* para um espaço de representação comum antes de fazer a busca



5- Tarefa: Recuperação multimodal de imagens

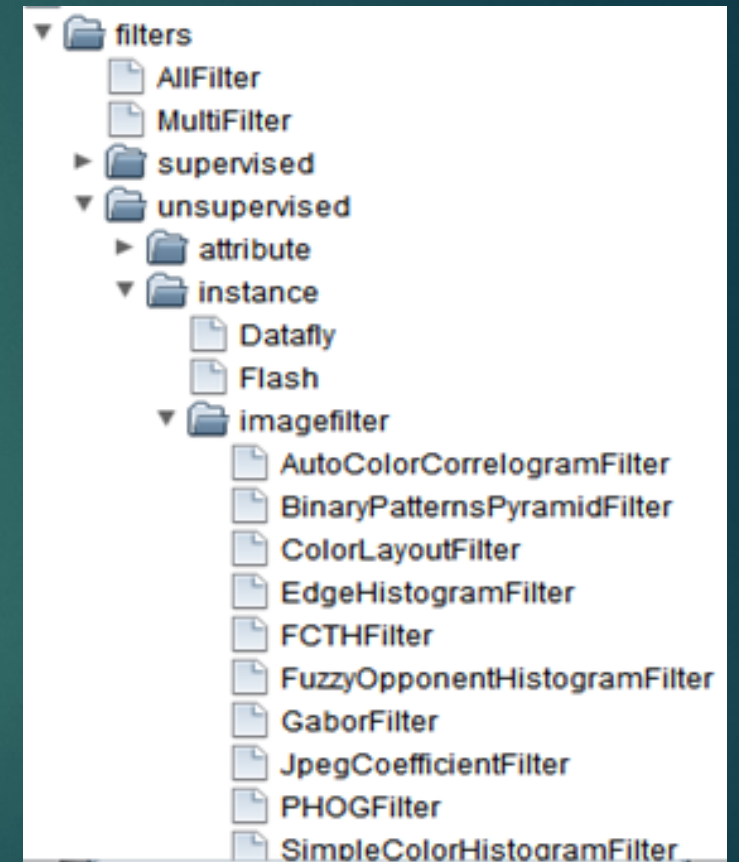


Brenda C. S. Berno, Sketch-based multimodal image retrieval using deep learning, Dissertação de Mestrado, CPGEI, UTFPR, 2021



Extração de *features* com Weka

- ▶ O **Weka** (v. 3.8) tem o pacote *image filters* que gera *features* (cor, textura, frequência) a partir de arquivos de imagens
- ▶ Escolher um ou mais filtros para aplicar:
 - ▶ AutoColorCorrelogram
 - ▶ BinaryPatternsPyramid
 - ▶ ColourLayout
 - ▶ EdgeHistogram
 - ▶ FCTH
 - ▶ FuzzyOpponentHistogram
 - ▶ Gabor
 - ▶ JpegCoefficient
 - ▶ PHOG
 - ▶ SimpleColorHistogram
- ▶ As *features* calculadas podem ser utilizadas em outro programa (Python ou Orange)



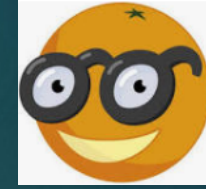
<https://www.cs.waikato.ac.nz/ml/weka/>

Extração de features com Orange

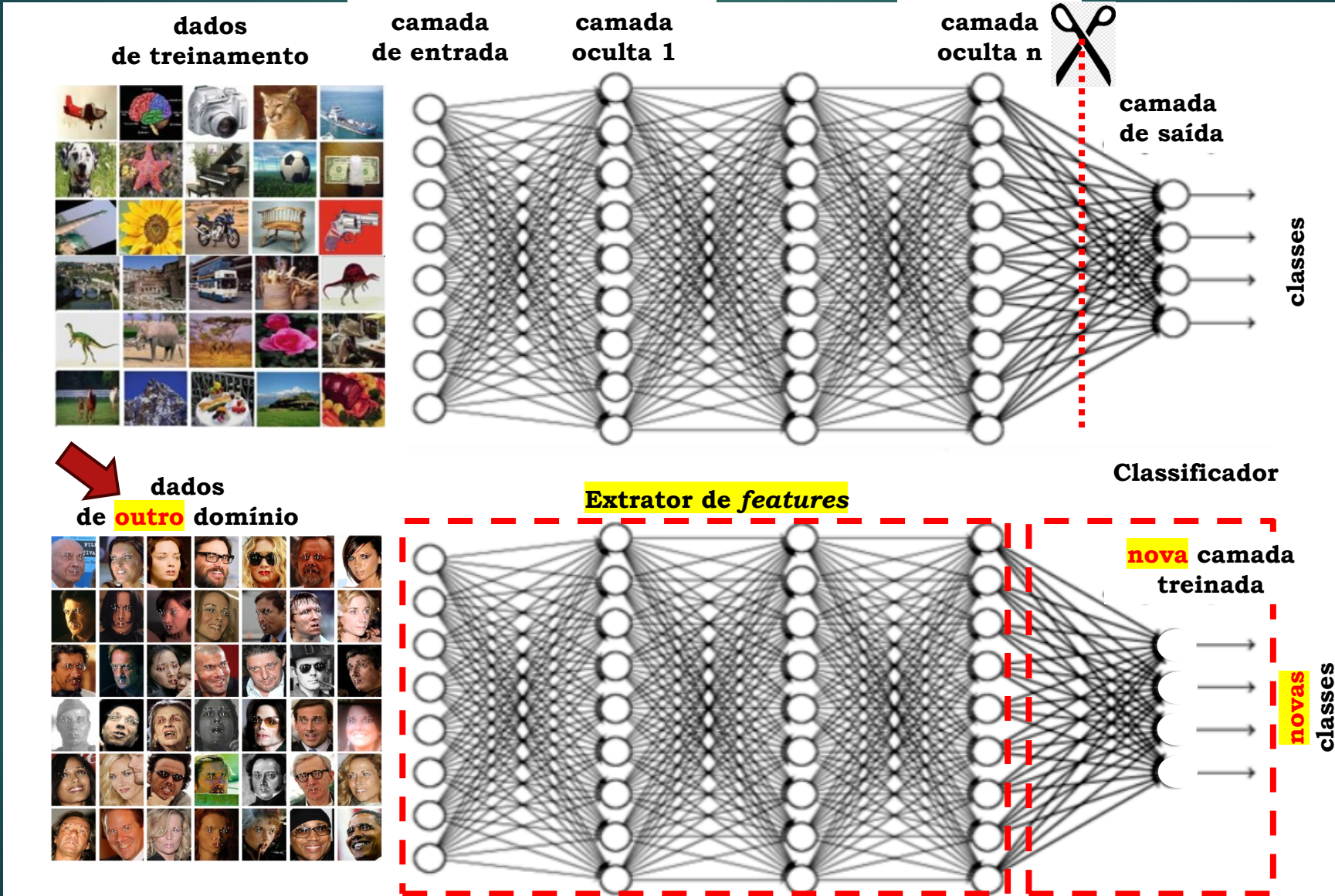
- ▶ O Orange **NÃO** tem extratores de *features* específicos para imagens
- ▶ O Orange tem um módulo de *Image Embedding* onde é possível utilizar uma rede convolucional (CNN) pré-treinada e fazer *transfer learning*
- ▶ As seguintes CNNs estão disponíveis:
 - ▶ Squeeze-net (1000 *features*) ★
 - ▶ Inception-v3 (4096 *features*)
 - ▶ VGG-16, VGG-19 (4096 *features*)
 - ▶ Painters (2048 *features*),
 - ▶ DeepLoc (512 *features*),
 - ▶ Openface (128 *features*)

★ O processamento é feito na nuvem, exceto para a Squeeze-net

Extração de features com Orange

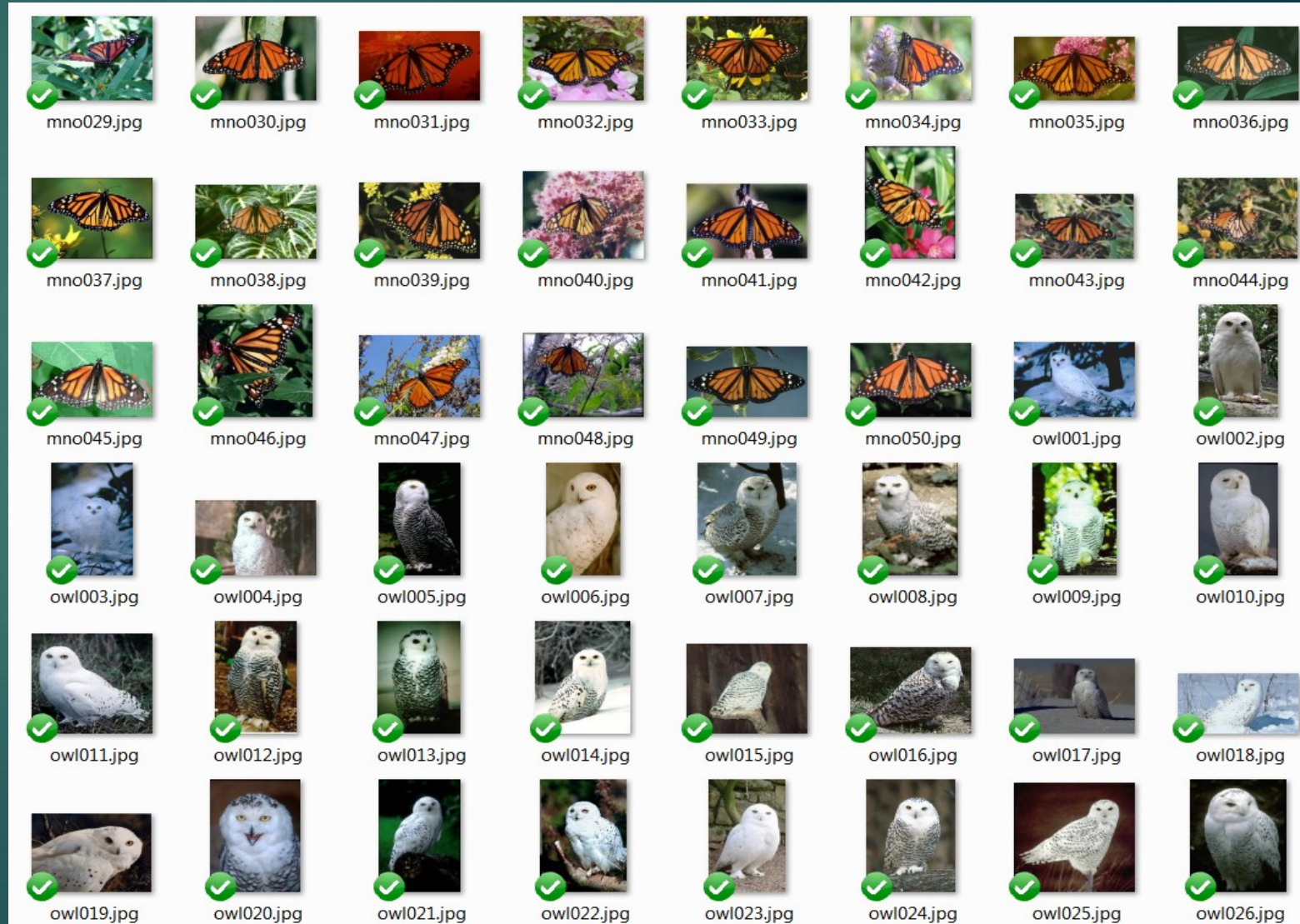


- ▶ O princípio do *Transfer-Learning*



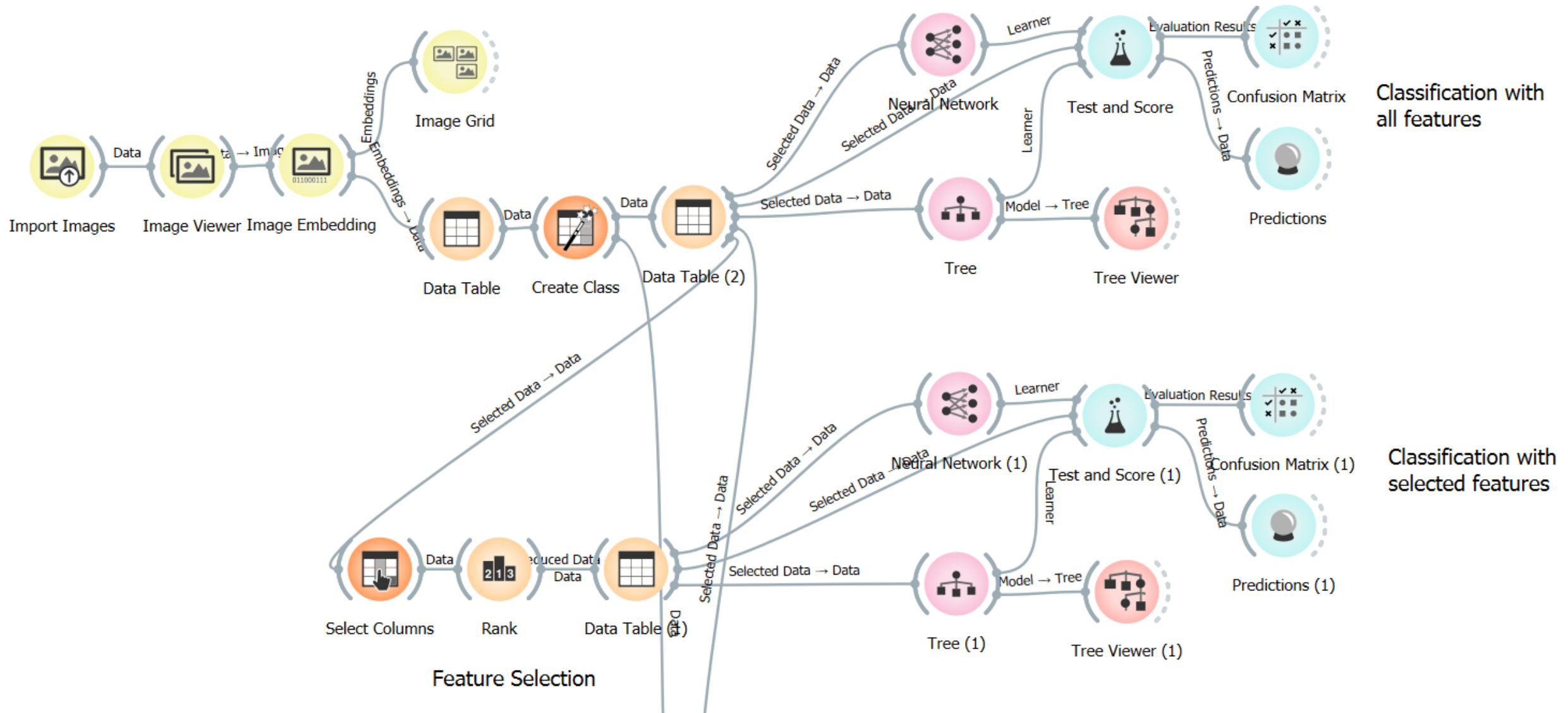
Estudo de caso #1: Butterfly x Owl

- ▶ Objetivo: classificar imagens de **borboletas-monarca** e de **corujas-das-neves**
- ▶ Objetos de outras classes são inseridos como “ruídos” para verificar o desempenho do classificador
- ▶ Utiliza-se a rede **Squeeze-net** para extrair um vetor de **1000** dimensões representando cada imagem



-

Estudo de caso #1: Butterfly x Owl



Estudo de caso #1: Butterfly x Owl

► Com todas as *features*:

Tree

Neural Network

tree

Neural Network

Show: Number of instances

Predicted

		butterfly	owl	other	Σ
Actual	butterfly	45	1	4	50
	owl	0	46	4	50
	other	4	7	1	12
	Σ	49	54	9	112

Predicted

		butterfly	owl	other	Σ
Actual	butterfly	50	0	0	50
	owl	0	50	0	50
	other	2	0	10	12
	Σ	52	50	10	112

► Com top-100 *features*:

Tree (1)

Neural Network

Actual

Predicted

	butterfly	owl	other	Σ
butterfly	45	1	4	50
owl	0	48	2	50
other	4	5	3	12
Σ	49	54	9	112

Tree (1)

Neural Network

Actual

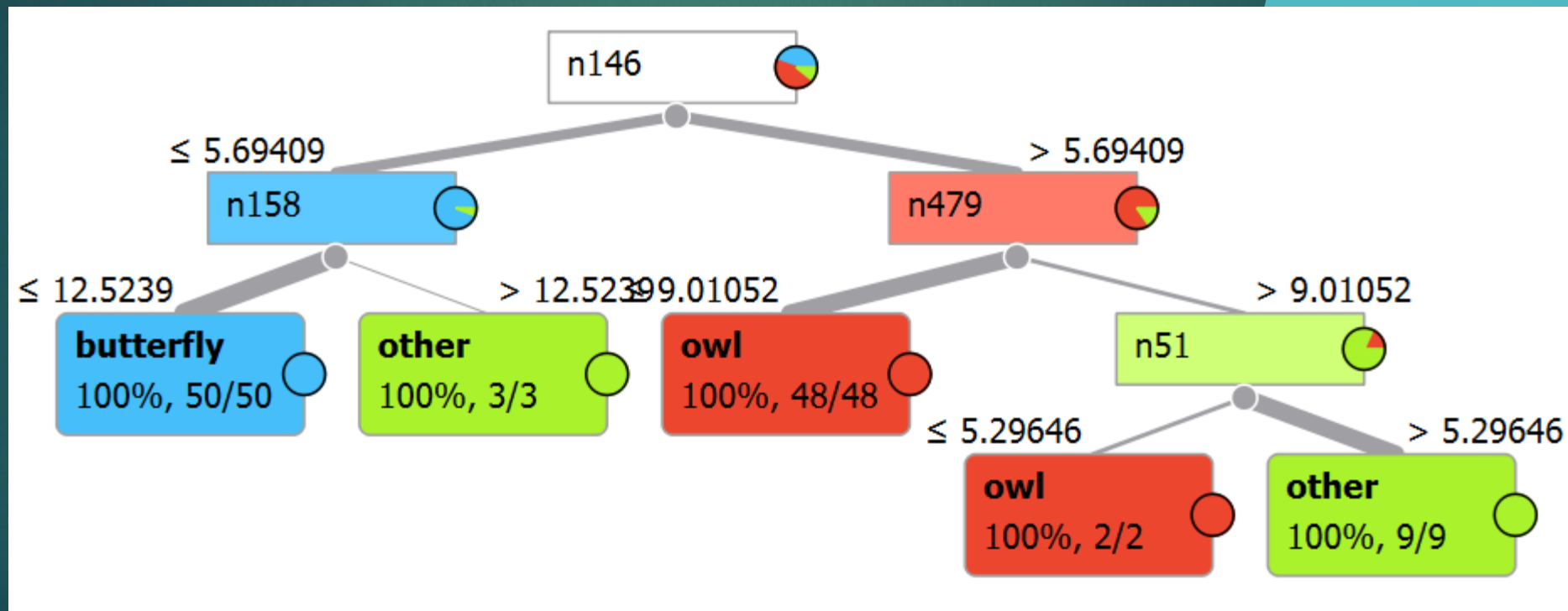
Predicted

	butterfly	owl	other	Σ
butterfly	50	0	0	50
owl	0	49	1	50
other	2	2	8	12
Σ	52	51	9	112

► Observe a importância do processo de *feature selection*

Estudo de caso #1: Butterfly x Owl

- ▶ A Árvore de Decisão gera um classificador muito simples e com 100% de acerto



Estudo de caso #1: Butterfly x Owl

- ▶ Teste em “**mundo aberto**” (*open-world*): Treinando com as duas classes conhecidas (borboleta e coruja) e testando com dados desconhecidos

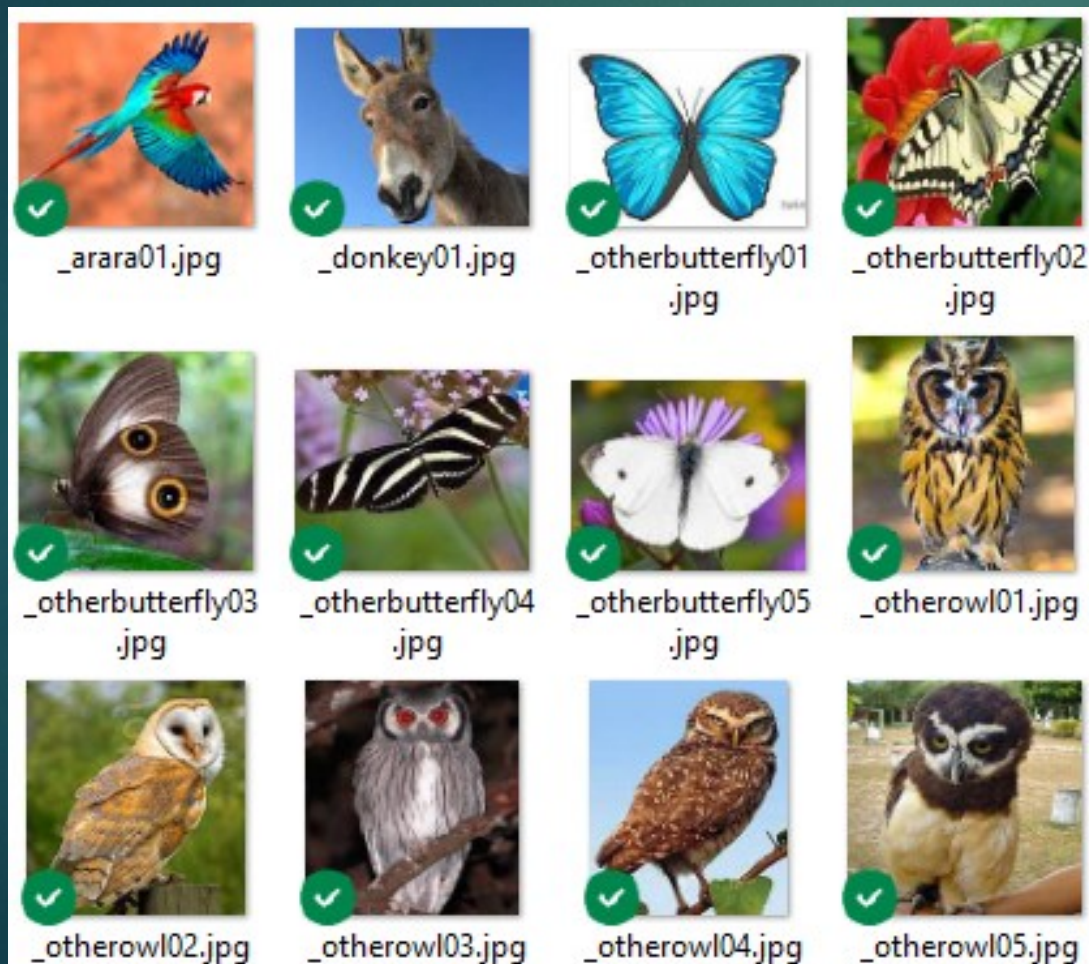


image name	Tree (2)		Neural Network	
_arara01	owl	?	butterfly	?
_donkey01	owl	?	owl	?
_otherbutterfly01	butterfly	⬡	butterfly	⬡
_otherbutterfly02	butterfly	⬡	butterfly	⬡
_otherbutterfly03	owl	⊘	owl	⊘
_otherbutterfly04	owl	⊘	butterfly	⬡
_otherbutterfly05	butterfly	⬡	owl	⊘
_otherowl01	owl	⬡	butterfly	⊘
_otherowl02	owl	⬡	owl	⬡
_otherowl03	owl	⬡	owl	⬡
_otherowl04	owl	⬡	owl	⬡
_otherowl05	owl	⬡	owl	⬡

Estudo de caso #2: Agrupamento e classificação de faces

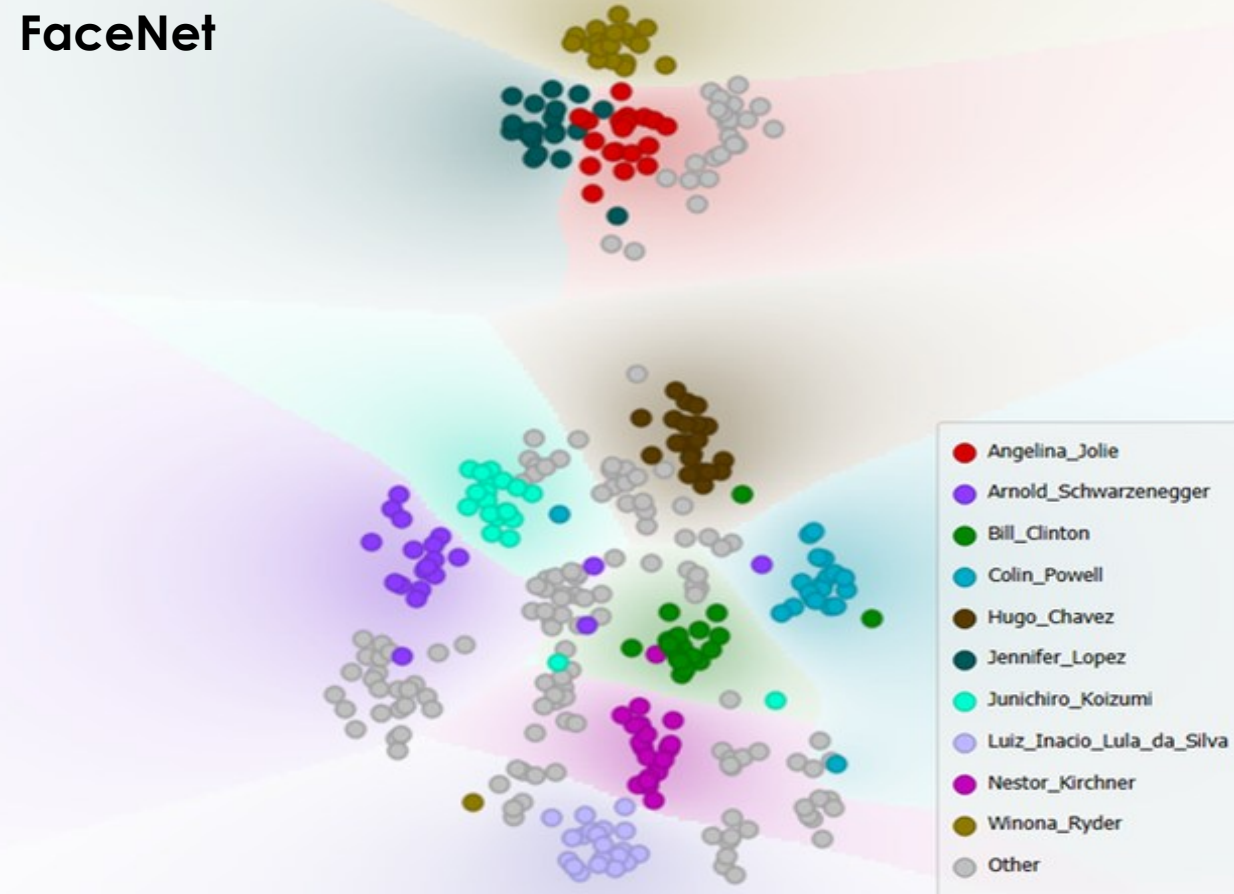
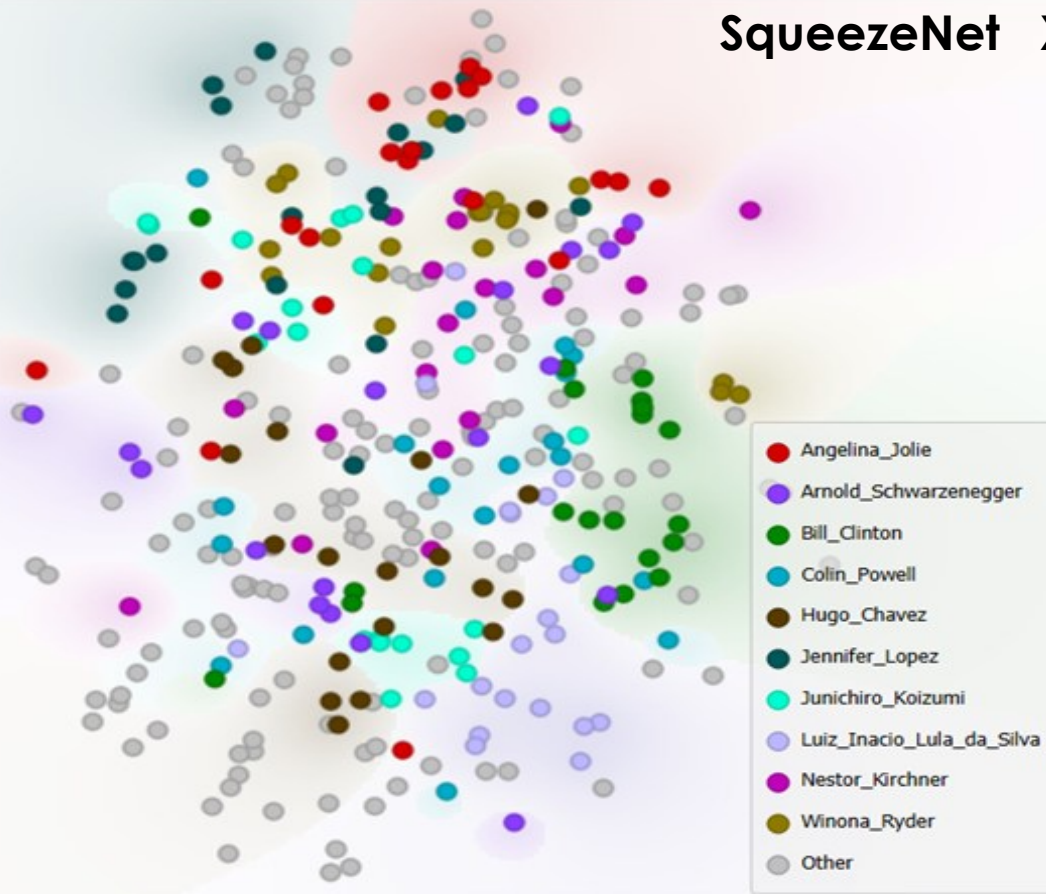
- ▶ **Dataset:** 350 imagens de 23 pessoas diferentes, COM/SEM a informação de classe
- ▶ **Tarefa1:** Comparar dois modelos de CNN para extração de *features* (OpenFace X SqueezeNet), para a tarefa de agrupamento (SEM usar a informação de classe)
- ▶ **Tarefa2:** Comparar os dois conjuntos de *features* extraídas para a tarefa de classificação, com e sem seleção de atributos



Estudo de caso #2: Agrupamento de faces

- Tarefa1: O estudo preliminar com T-SNEE mostra a extraordinária capacidade de agrupamento da rede especializada, comparada com a rede de propósito geral

SqueezeNet X FaceNet



Estudo de caso #2: Agrupamento de faces

► Tarefa 1: Agrupamento

- O Coeficiente de Silhouette indicou, para ambos os casos, 2 grupos
- As *features* extraídas pela rede *FaceNet* permitiram a separação em dois grupos majoritariamente baseados no gênero (à exceção de uma classe: Q.E.II)
- O agrupamento hierárquico corroborou o achado anterior e mostra claramente que o melhor agrupamento ocorre com a rede especializada
- Alguns poucos erros ocorrem quando a imagem tem mais de uma pessoa identificável

Estudo de caso #2: Classificação de faces

► Tarefa 2: Classificação

- A rede especializada, apesar de ter apenas 128 *features*, consegue levar ambos os classificadores a ótimos resultados

Model ▼	AUC	CA	F1	Precision	Recall
SVM (1)	0.990	0.937	0.930	0.929	0.937
Neural Network	0.993	0.931	0.931	0.934	0.931

- A rede de propósito geral tem 1000 *features*, mas não consegue levar os classificadores a bons resultados

Model ▼	AUC	CA	F1	Precision	Recall
SVM (2)	0.921	0.534	0.499	0.501	0.534
Neural Network	0.948	0.626	0.618	0.622	0.626

- Adicionalmente, utilizando somente 30 componentes principais (PCA) obtém-se resultados marginalmente melhores com a rede especializada

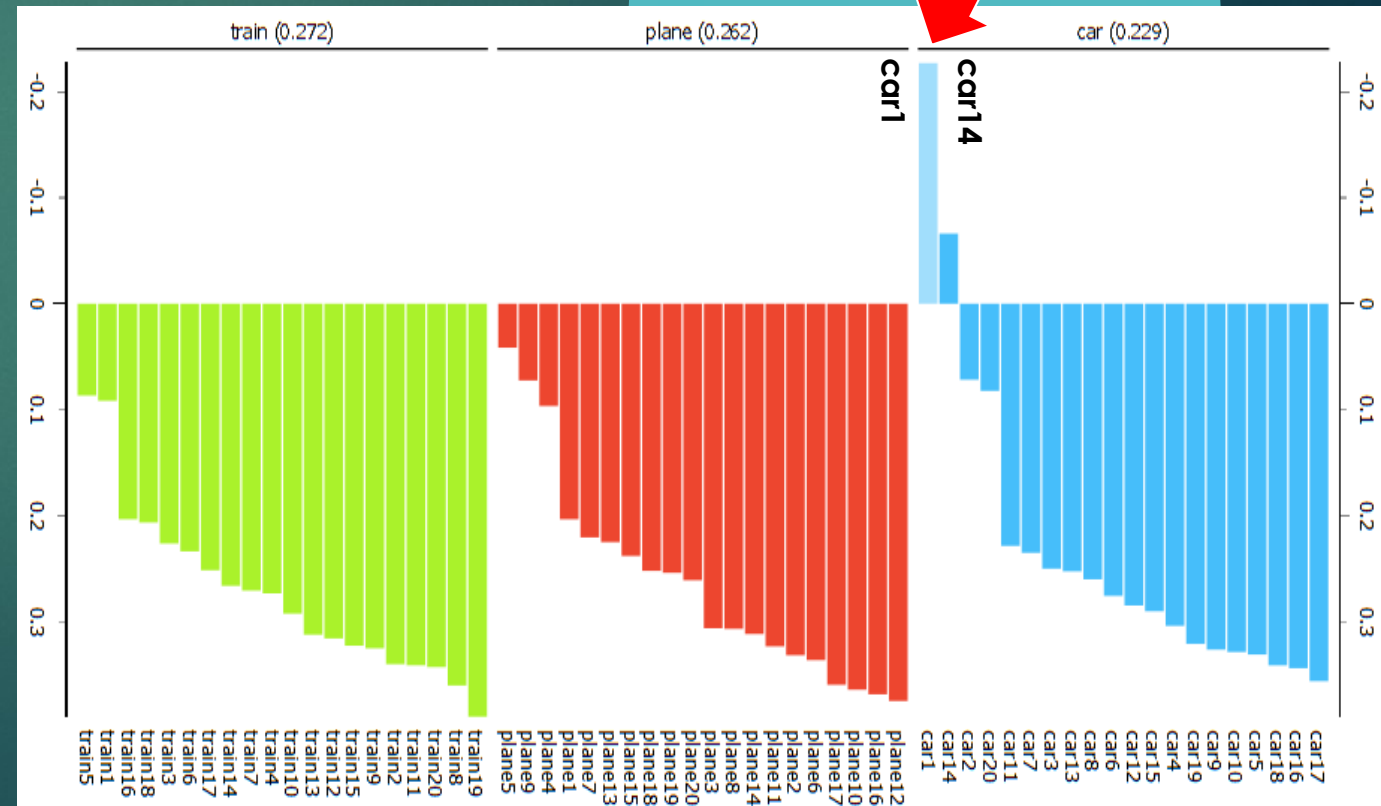
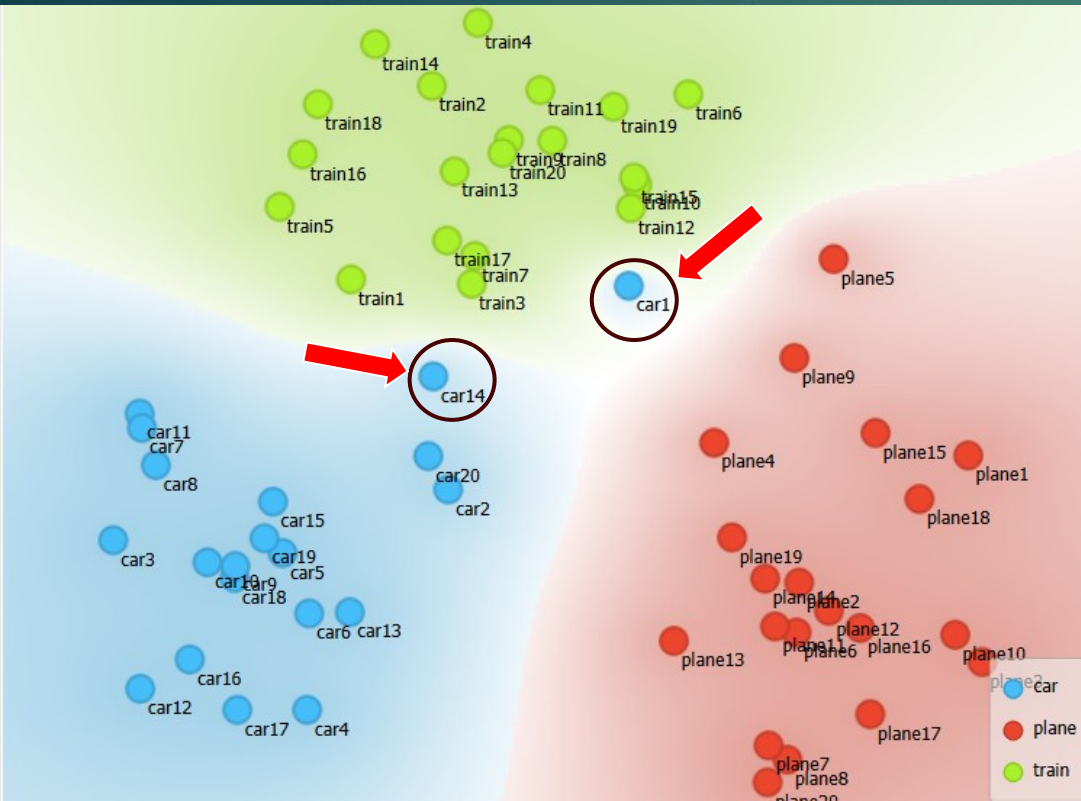
Estudo de caso #3: Classificação de meios de transporte

- ▶ *Dataset*: 3 classes de imagens: aviões, trens e carros. Partição: para cada classe: 20 imagens para treino e 15 para teste
- ▶ Objetivo: comparar o desempenho da classificação utilizando *features* extraídas por uma CNN versus *features* extraídas pelos descritores de imagens (c/ Weka).



Estudo de caso #3: classificação de meios de transporte

- ▶ O estudo preliminar com T-SNEE mostra clara separação entre classes. O coeficiente de Silhouette corrobora esta observação



Estudo de caso #3: classificação de meios de transporte

- ▶ Devido à boa separabilidade entre classes, todos os classificadores obtém ótimos resultados e a árvore de decisão resultante é excepcionalmente simples

Model	AUC	CA	F1	Precision	Recall
Tree (1)	0.983	0.978	0.978	0.979	0.978
SVM (1)	1.000	1.000	1.000	1.000	1.000
Neural Network	1.000	1.000	1.000	1.000	1.000

