Prof. Heitor Silvério Lopes
Prof. Thiago H. Silva

# Data Mining & Knowledge Discovery

Class 9a -Images Mining
2025

# Image Mining

- Images mining is included inside a broader concept known as “Multimidia Mining”, which includes, besides images, also video and audio
    - Definition: it is the process of searching and discovering high-level patterns or implicit knowledge (**not explicitly visible**) in images
    - Meaning a large amount of images
    - Image mining is an area of high evidence due to the huge amount of stored images available in the internet

# Image Processing X Image Mining

- Image Processing:
  - The objective is to treat images by means of mathematical/computational methods aiming at obtaining an improved image (related to the original one) or extracting specific patterns
- Image Mining:
  - It is inderdisciplinar, since it uses concepts from: Databases, Statistics, Machine Learning, Pattern Recognition, and Computational Intelligence
  - It also can use the traditional Image Processing methods for **pre-processing** the images
  - Image mining follows the same data mining procedures, as usual, except by the fact that images are transformed into a set of numerical atributes (features)

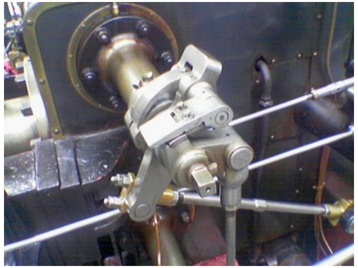
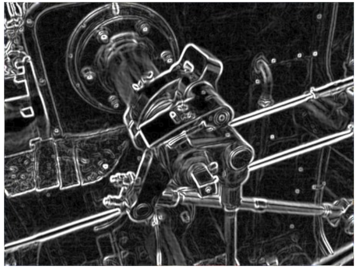
# Steps for processing images

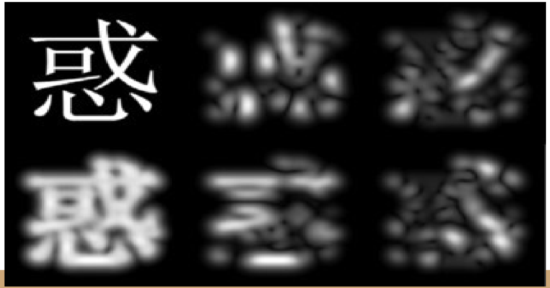
IMPORTANT

- **Pre-processing**:  to eliminate all types of noise and format transformation
- **Feature extraction**: to apply mathematical methods to transform an image into a low-dimensional numeric vector (features)
- **Feature selection**: to reduce the dimensionality of the data
- **Analysis**: using the reduced vector, to apply the traditional classification/clustering methods
- Special procedures: Reversal Search, Multimodal Image Retrieval

# 1- Pre-processing: image filters

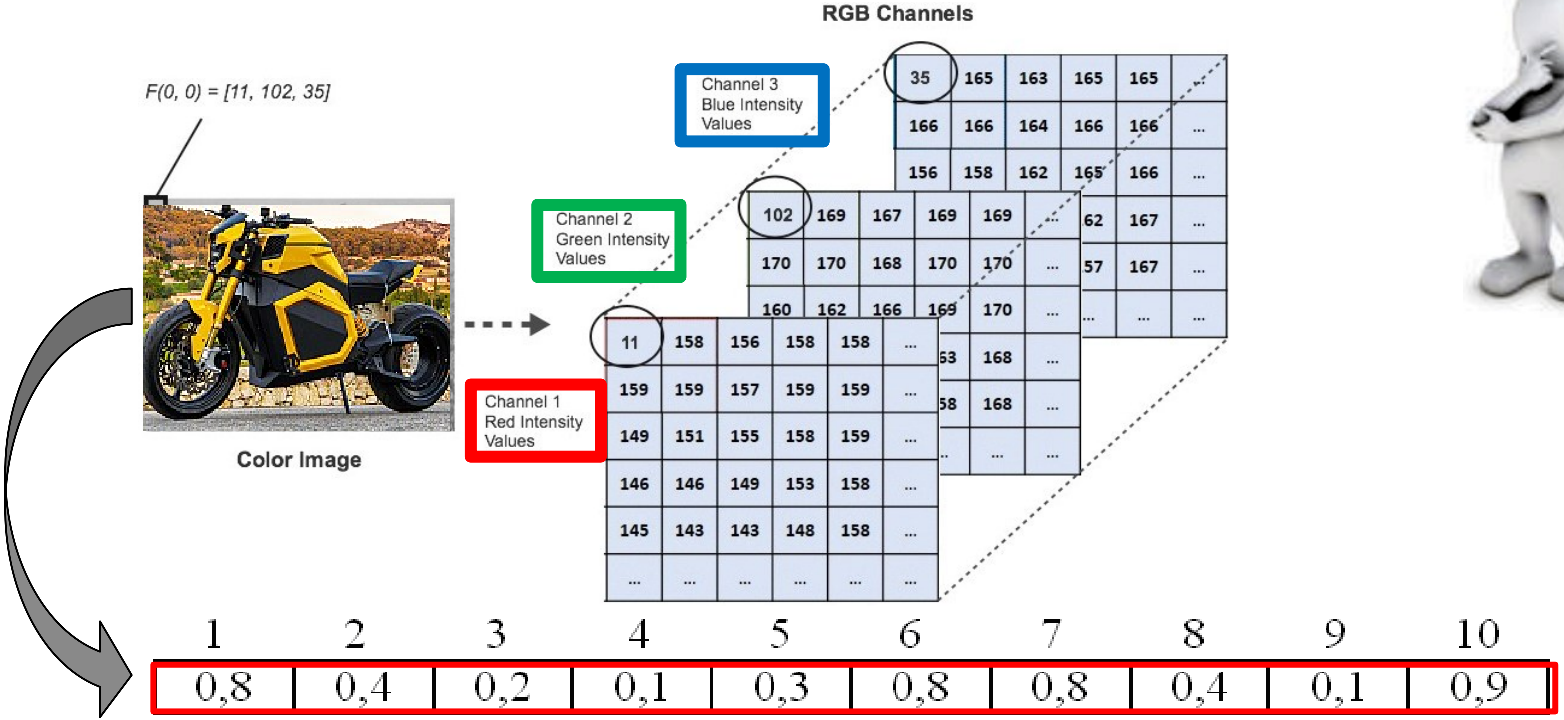
- Aims to highlight or accentuate some important characteristic of the image. Some examples:
    - Edge Filter (Canny)
    - Color Filter
    - Direction Filter (Sobel)



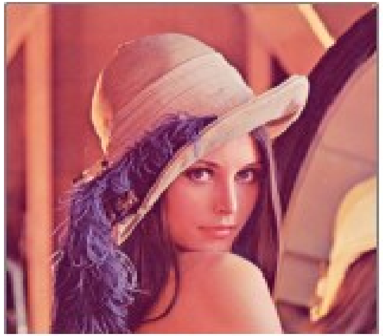
# 2- Feature extraction from images

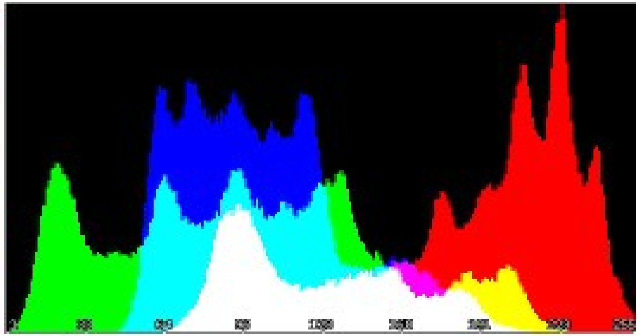
- It is the simple transformation of an image into numerical vector



# 2- Feature extraction from images

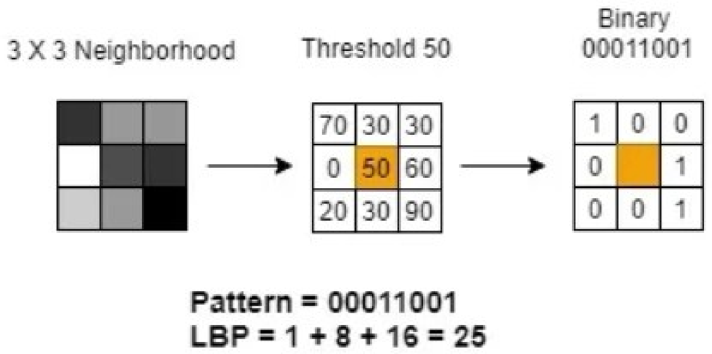
- Color Descriptors: they are invariant related to scales, translation/rotation of the image. Examples:
  - **RGB Histogram**: it is a combination of the R, G, and B color historgrams
  - Other: Color Moments, Color Coherent Vector, Color SIFT descriptors, etc.





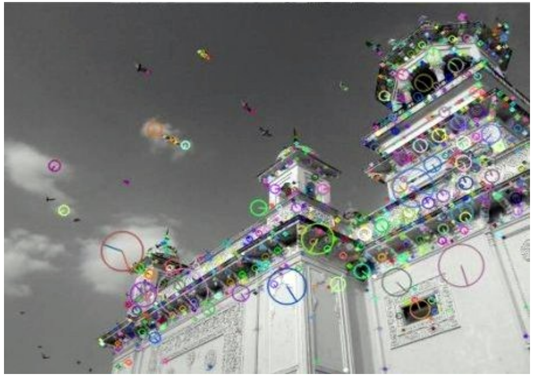
# 2- Feature extraction from images

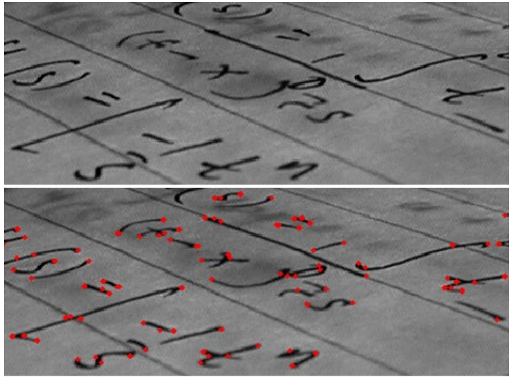
- Texture Descriptors: they measure smoothness and regularity of the image
  - **GLCM** (Gray-Level Co-occurrence Matrix): extracts statistical measures from the image: Second Angular Momentum, Correlation, Inverse Differential Moment and Entropy
  - **LBP** (Local Binary Patterns): it is very popular since it is computationally efficient, and it is robust to illumination changes. Good for face and object recognition
  - There are many variants of LBP: https://github.com/carolinepacheco/lbp-library

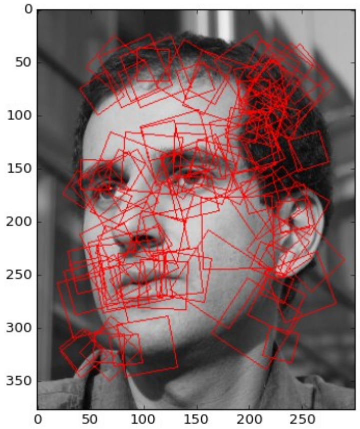



# 2- Feature extraction from images

- Frequency domain features: have low computational cost and they are based on the detection of interest points in the image
    - **SIFT** (*Scale Invariant Feature Transform*) has robustness to illumination changes and small positional variations
    - **SURF** (*Speed-Up Robust Features*) is a detector for interest points in an image and it is invariant to rotation or scaling
    - **ORB** (*Oriented FAST and Rotated BRIEF*): improved version of SURF

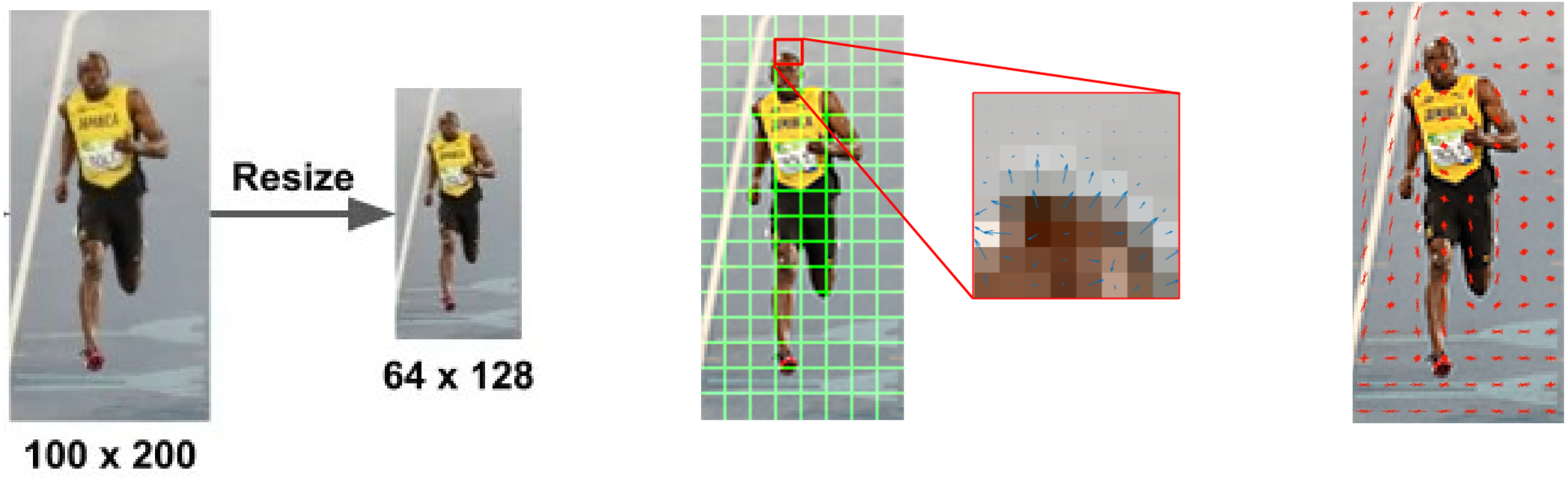






# 2- Feature extraction from images

- Histogram of Oriented Gradients (HOG)
    - It is inspired on SIFT and it is based on the distribution of gradient directions (derivative of x and y axes of the image)
    - Very useful for pedestrian, vehicles and animals detection in images



# Some Python libraries for feature extraction in images

- **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

# Some Python libraries for feature extraction in images

- **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 and Laplacian**: 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- Image classification and clustering

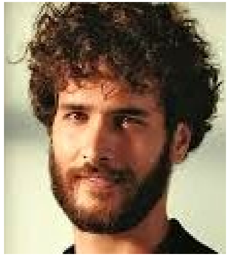
IMPORTANT

- The same concepts of **Descritive/Predictive Modelling** are valid for images
- A simple method based on image descriptors can be useful when:
  - The features that differentiate the images are very specific (e.g. color, shape, etc)
  - A model is desired to understand (describe) the relevant features of the images
  - A low computational cost is required
- A complex method (convolutional/transfer-learning) for feature extraction can be useful when:
  - The task has high semantic complexity
  - High accuracy is the priority, at the expense of high computational cost

# 3- Image classification and clustering

- Image classification use the same computational methods used for regular data
  - **Classification** (supervised learning): OneR, Decision Trees, SVM, Neural Networks...
  - **Clustering** (nonsupervised learning): K-means, Hierarchical learning...

| *f1* | *f2* | *f3* | *f4* | *f5* | ... | *fn* | classe |
|---|---|---|---|---|---|---|---|
| 0,123 | 0,946 | 0,856 | 0,168 | 0,02 | ... | 0,431 | **F** |



| *f1* | *f2* | *f3* | *f4* | *f5* | ... | *fn* | classe |
|---|---|---|---|---|---|---|---|
| 0,002 | 0,701 | 0,287 | 0,949 | 0,923 | ... | 0,581 | **M** |

# 4- Reverse Image Search

- It is a CBIR (Contents-Based Image Retrieval) where a given image is the query and the system searches an image database for similar/related images
- Reverse Image Search can be use for:
  - Locate the origen or author of the image
  - Find other versions of the image (better resolution, faked images...)
  - Find internet sites where the image appears.
- Many methods can be used:
  - SIFT, BoVW (Bag of Visual Words), etc
- Several internet sites use reverse image Search
  - Google image Search, Bing images, Picsearch, Pxsy, Pinterest, etc

# 5- Multimodal Image Retrieval

- It is a relatively new research area
- It is aimed to find, in a multimidia database, resources of a given modality (e.g. image) using a query of another modality (e.g. text, sketch, audio...)

Text

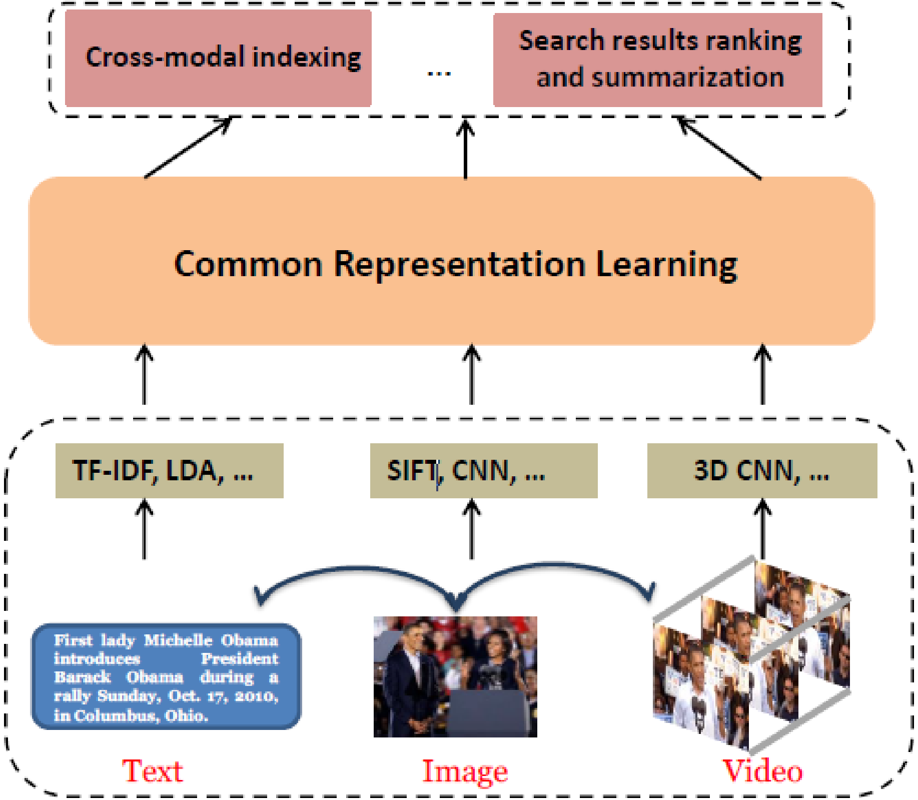

Image


Video

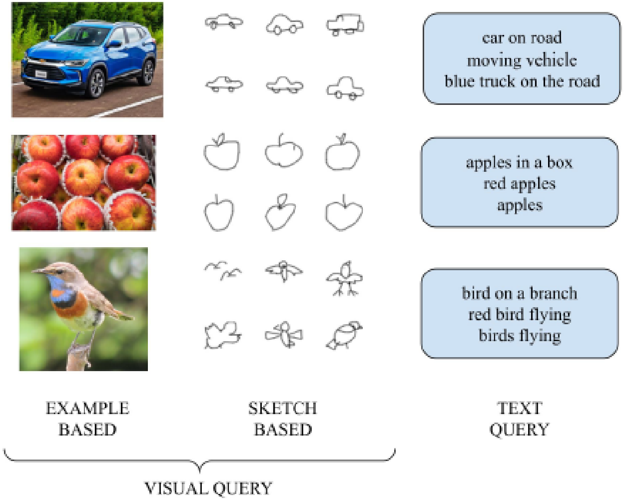

Audio

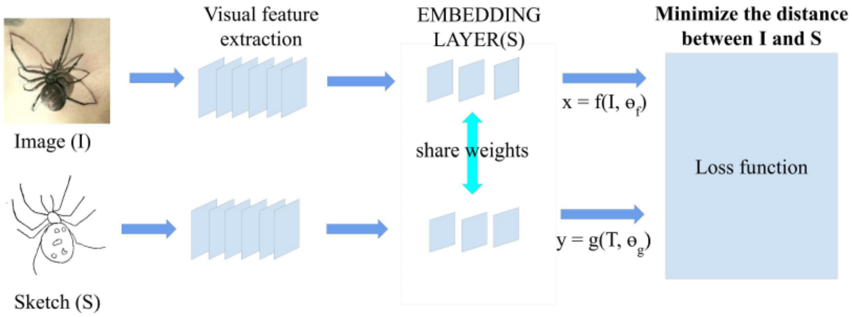
# 5- Multimodal Image Retrieval

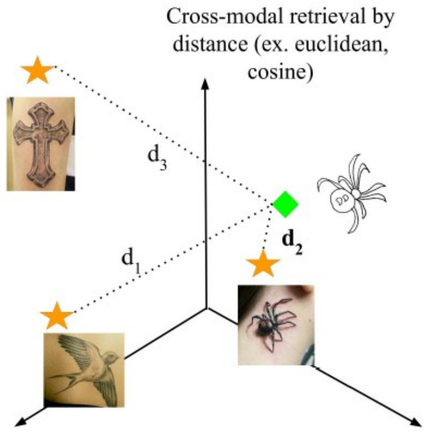
- Several features are extracted from each input, according to its modality
- All feature vectors are transformed into a common representation space before search



# 5- Multimodal Image Retrieval



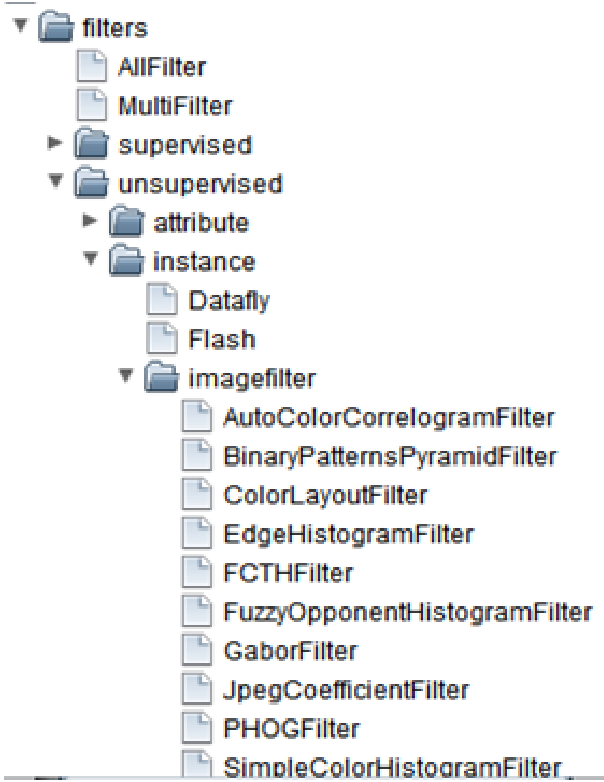




*Brenda C. S. Berno, Sketch-based multimodal image retrieval using deep learning, MSc. Dissertation, CPGEI, UTFPR, 2021*

# Image feature extraction using **Weka**

- O Weka (v. 3.8) has an *image filters* library for extracting features from images, for instance:
    - AutoColorCorrelogram
    - BinaryPatternsPyramid
    - ColourLayout
    - EdgeHistogam
    - FCTH
    - FuzzyOpponentHistogram
    - Gabor
    - JpegCoefficient
    - PHOG
    - SimpleColorHistogram
- The features extracted by Weka can be later used in other systems Python or Orange

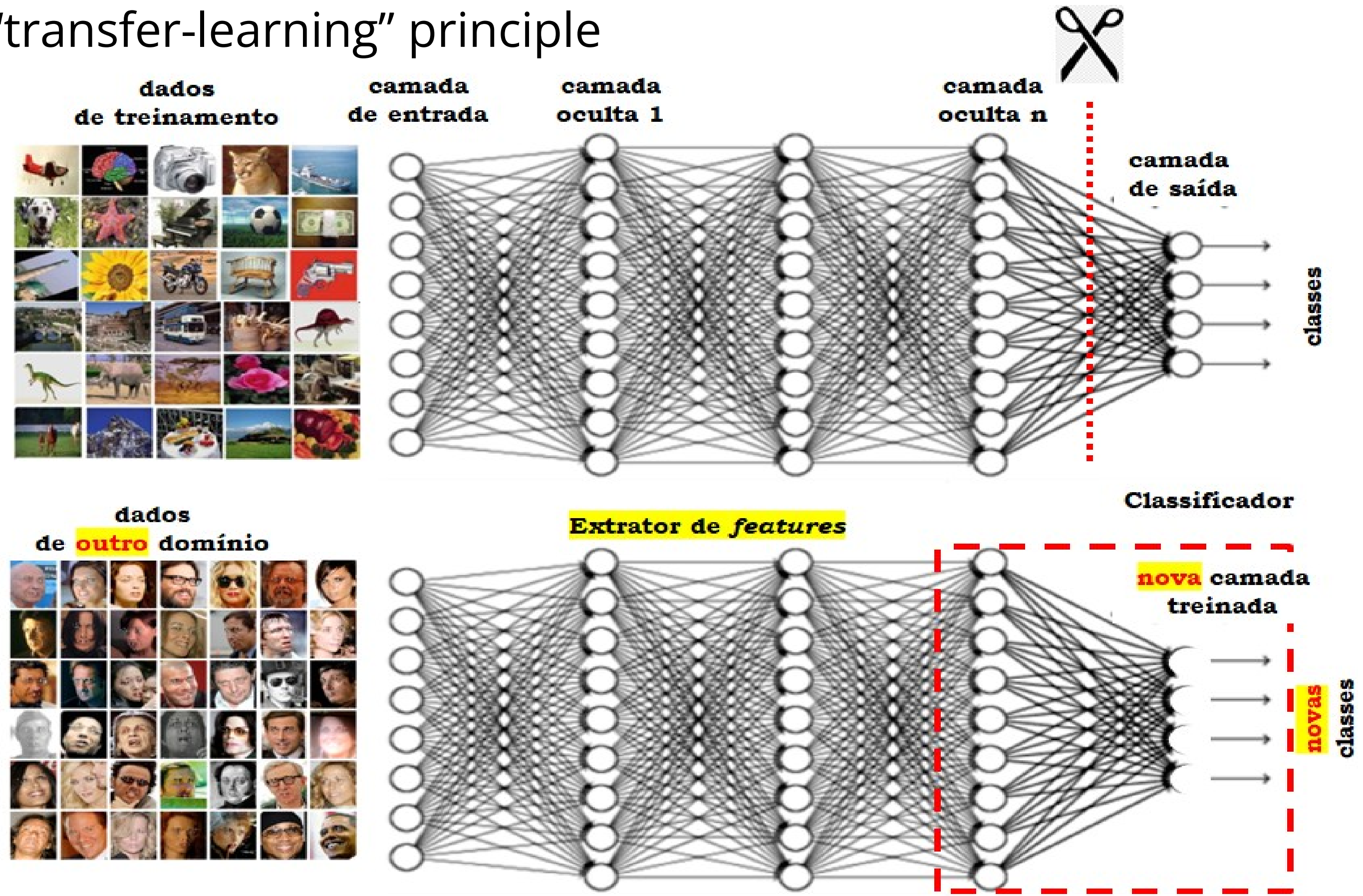


# Image feature extraction using **Orange**

- Orange does not have specific feature extractors for images
- It is possible to use a Python code to compute image features (e.g. LBP, HOG, RGB histogram, etc) and use them inside Orange
- Orange uses a pretrained convolutional neural network (CNN) for extracting image features. The following are available:
  - Squeeze-net (1000 features)
  - Inception-v3 (4096 features)
  - VGG-16, VGG-19 (4096 features)
  - Painters (2048 features)
  - DeepLoc (512 features)
  - Openface (128 features)
- All processing is done in the cloud, except for Squeeze-net which is locally processed

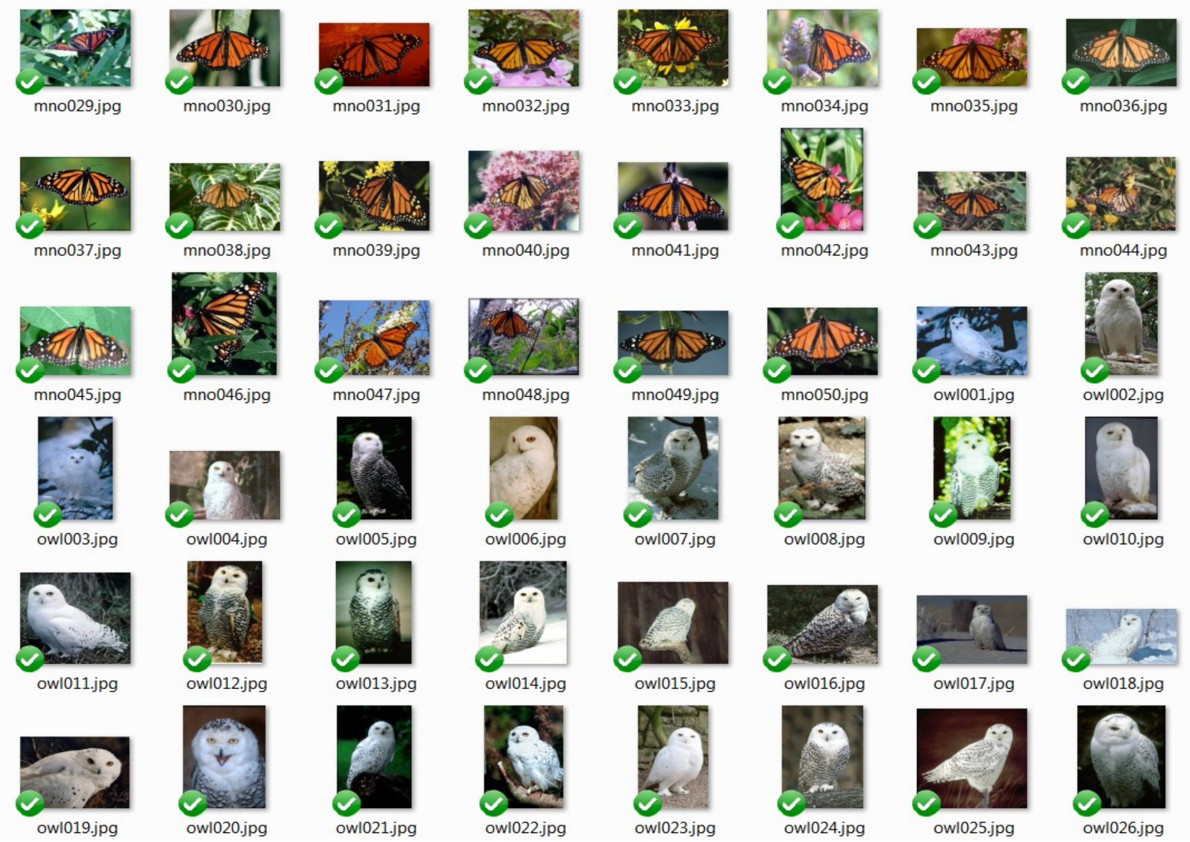
# Image feature extraction using **Orange**

- The “transfer-learning” principle



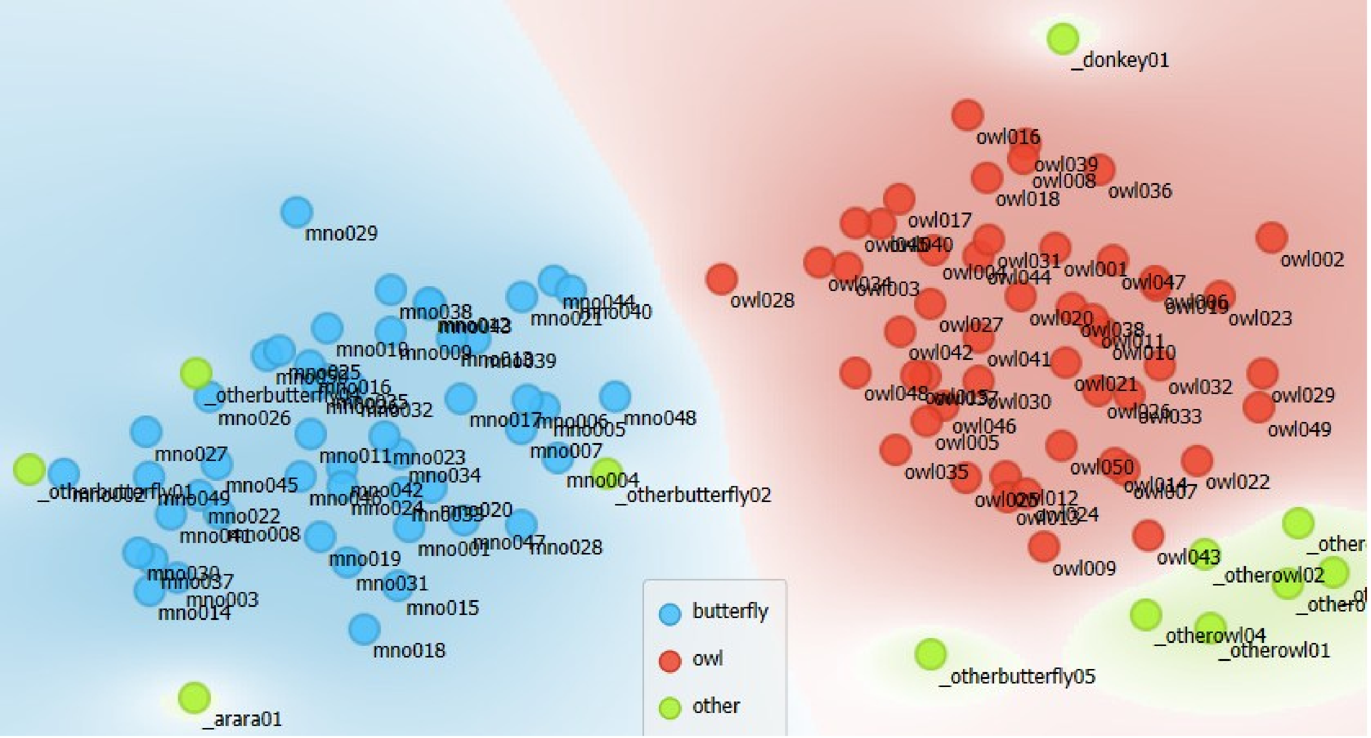
# Case study #1: Butterfly x Owl

- Objective: classify images of monarch butterflies and snowy owls
- Objects from other classes are inserted as “noise” to check the performance of the classifier
- Squeeze-net is used to extract a 1000-dimensional feature vector

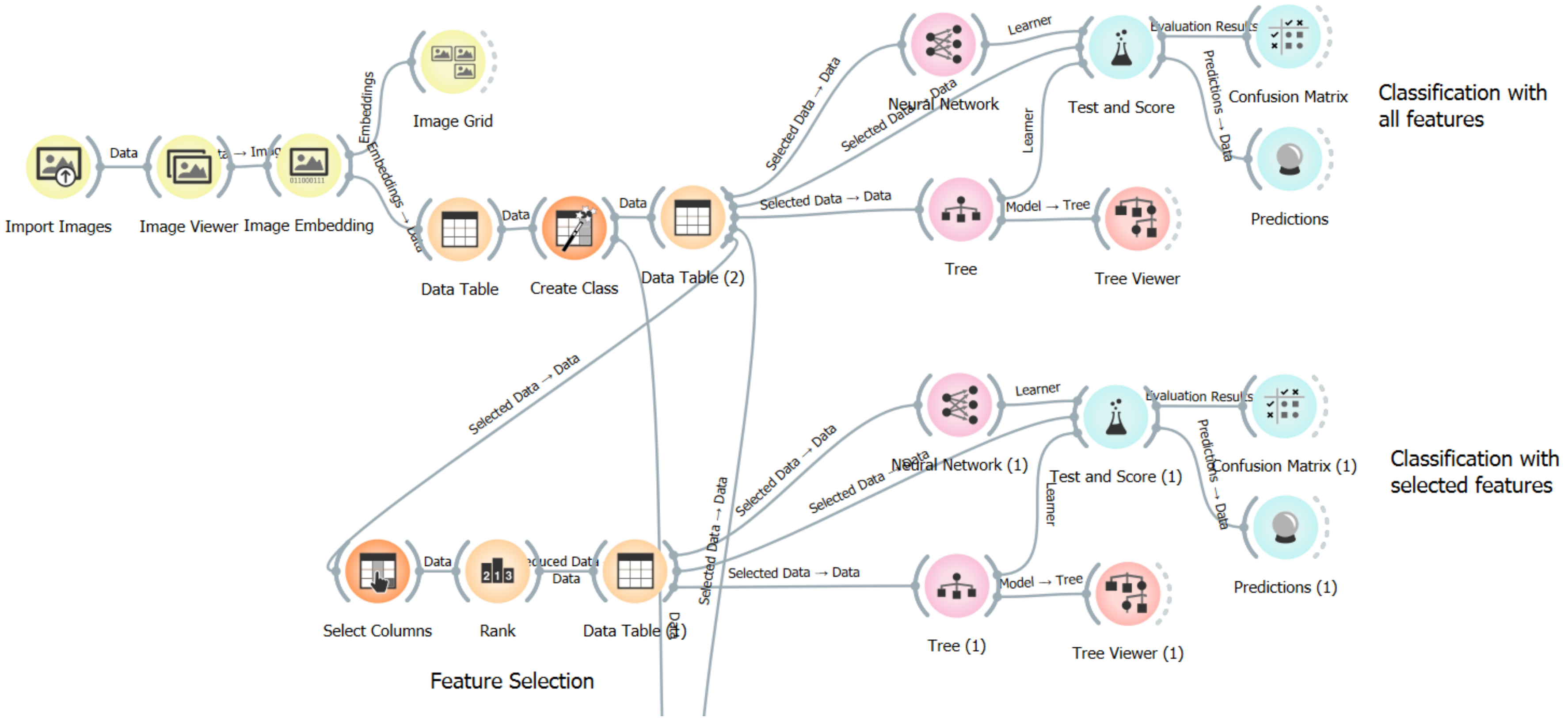


# Case study #1: Butterfly x Owl

- Clustering analysis with T-SNE shows a good separability between the two main classes



# Case study #1: Butterfly  x Owl



# Case study #1: Butterfly x Owl

- All features

Tree

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

Neural Network

| Actual \ Predicted | butterfly | owl | other | Σ |
|---|---|---|---|---|
| butterfly | 50 | 0 | 0 | 50 |
| owl | 0 | 50 | 0 | 50 |
| other | 2 | 0 | 10 | 12 |
| Σ | 52 | 50 | 10 | 112 |

- Top-100 features

Tree (1)

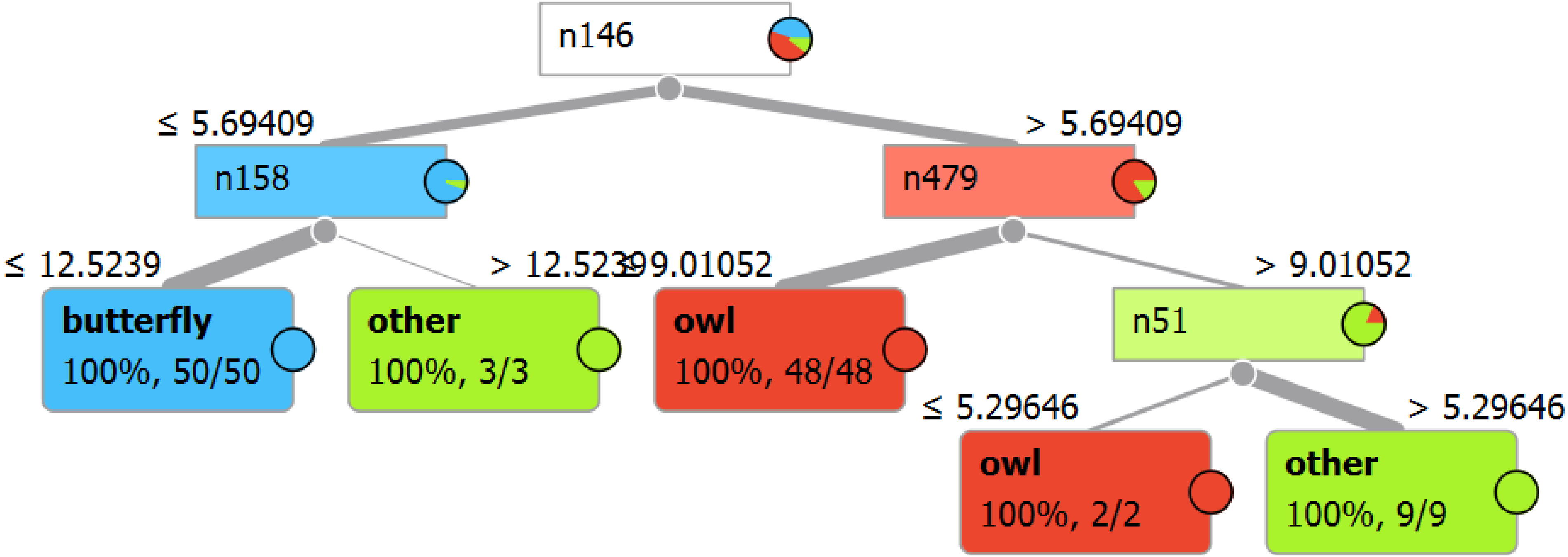
| Actual \ Predicted | butterfly | owl | other | Σ |
|---|---|---|---|---|
| butterfly | 45 | 1 | 4 | 50 |
| owl | 0 | 48 | 2 | 50 |
| other | 4 | 5 | 3 | 12 |
| Σ | 49 | 54 | 9 | 112 |

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 |

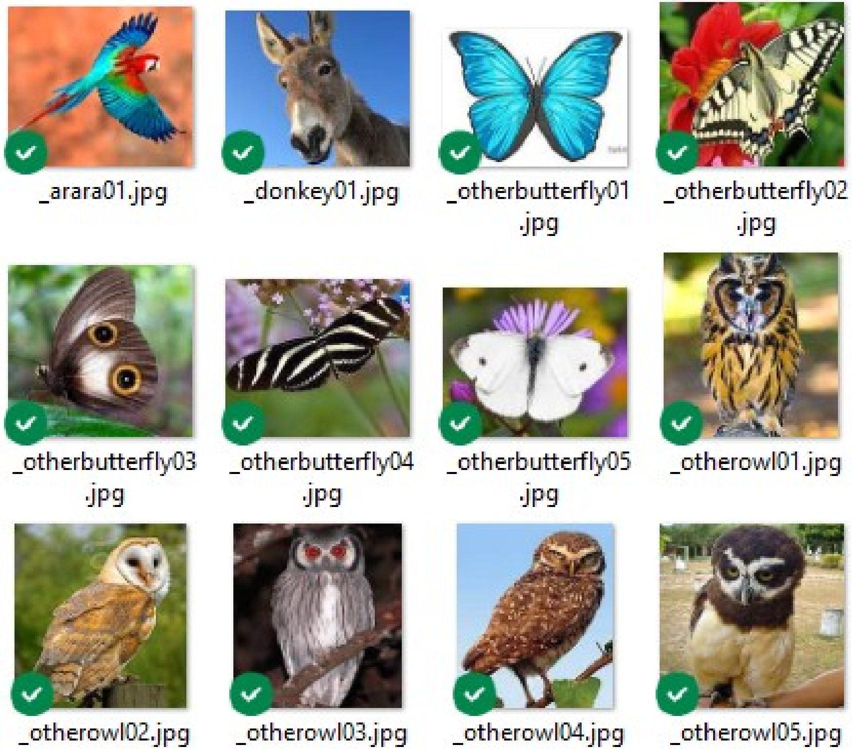
# Case study #1: Butterfly  x Owl

- A Decision-Tree classifier is quite simple and achieves 100% correct classification



# Case study #1: Butterfly x Owl

- "Open-world" test: training with the two known classes (owl and butterfly) and testing with unknown data



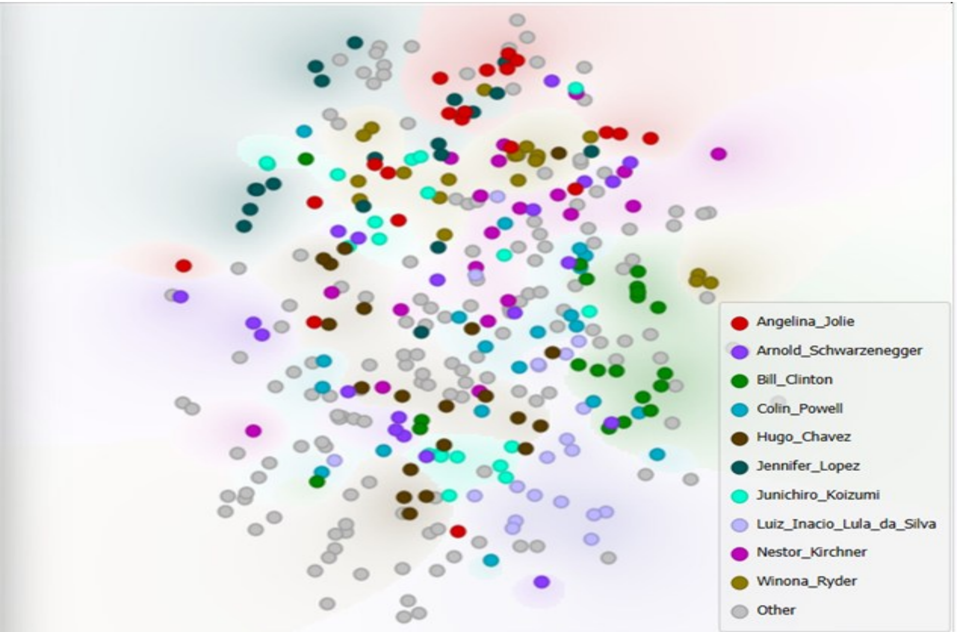

| image name | Tree (2) | Neural Network |
|---|---|---|
| _arara01 | owl ? | butterfly ? |
| _donkey01 | owl ? | owl ? |
| _otherbutterfly01 | butterfly | butterfly |
| _otherbutterfly02 | butterfly | butterfly |
| _otherbutterfly03 | owl | owl |
| _otherbutterfly04 | owl | butterfly |
| _otherbutterfly05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

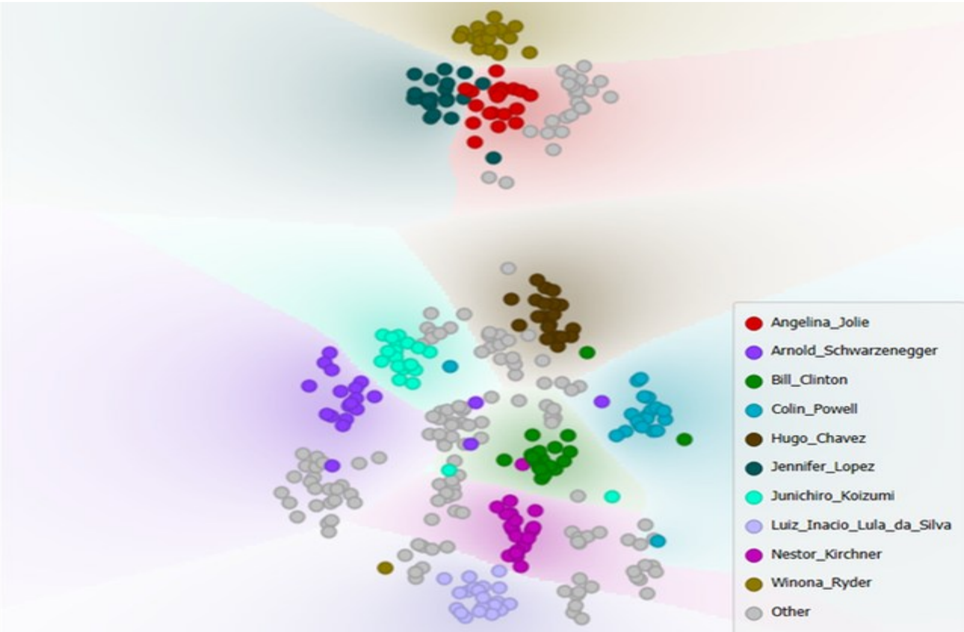
# Case study #2: Faces clustering/classification

- **Dataset**: 350 images from 23 different persons, WITH and WITHOUT class
- **Task 1**: Comparison of two CNN models as feature extractors (FaceNet X SqueezeNet) for a clustering task (NO class information)
- **Task 2**: Comparison of two sets of features for a classification task, with and without feature selection

# Case study #2: Faces clustering/classification

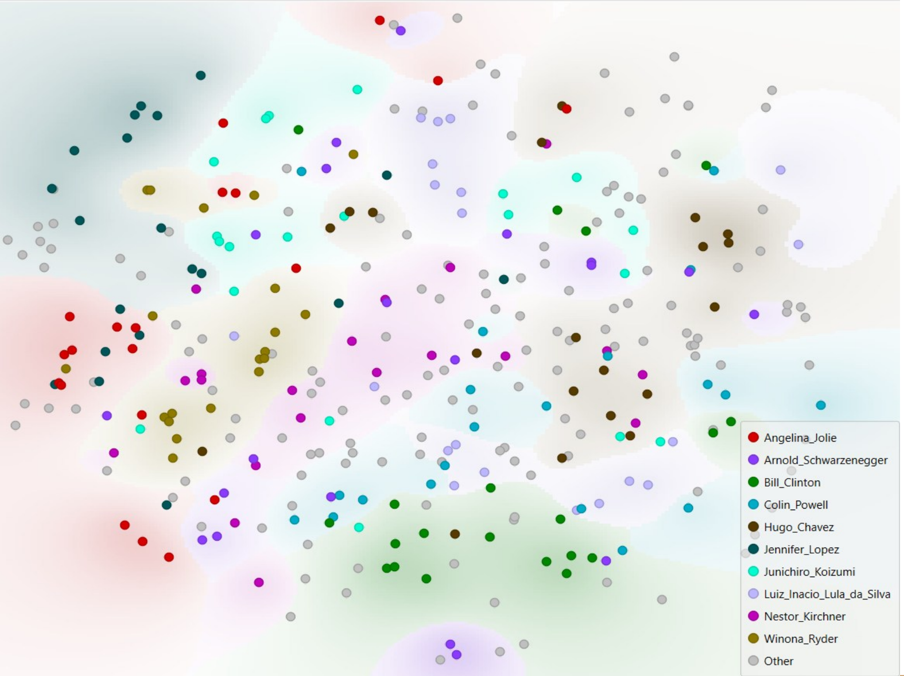
- **Task 1**: a preliminar study using T-SNE shows that the general-purpose CNN (SqueezeNet) cannot create clear clusters, while the specialized CNN (FaceNet) creates a concise clusters of the face images.

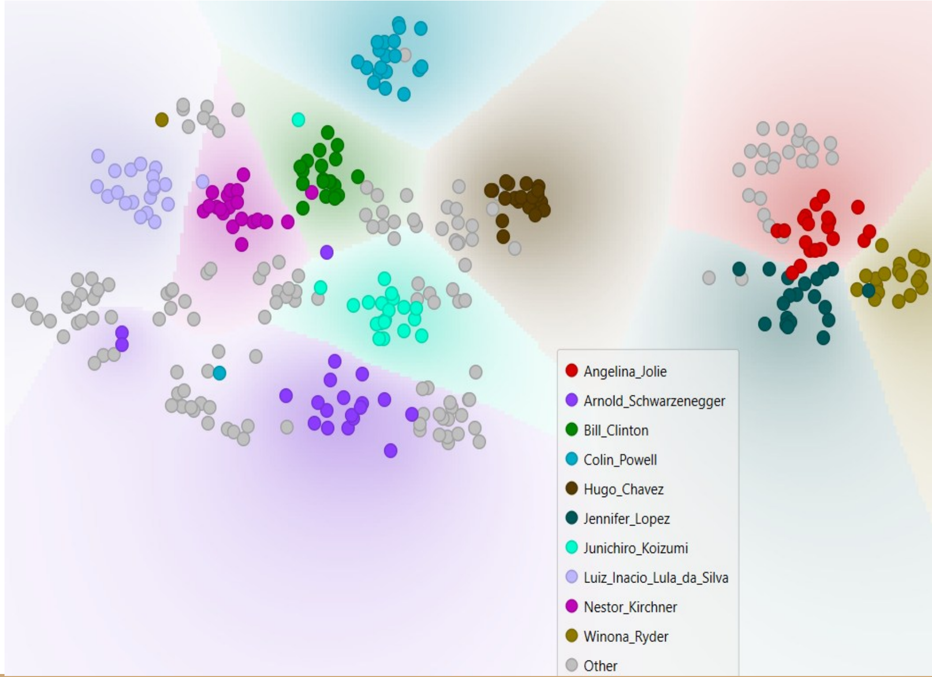


# Case study #2: Faces clustering/classification

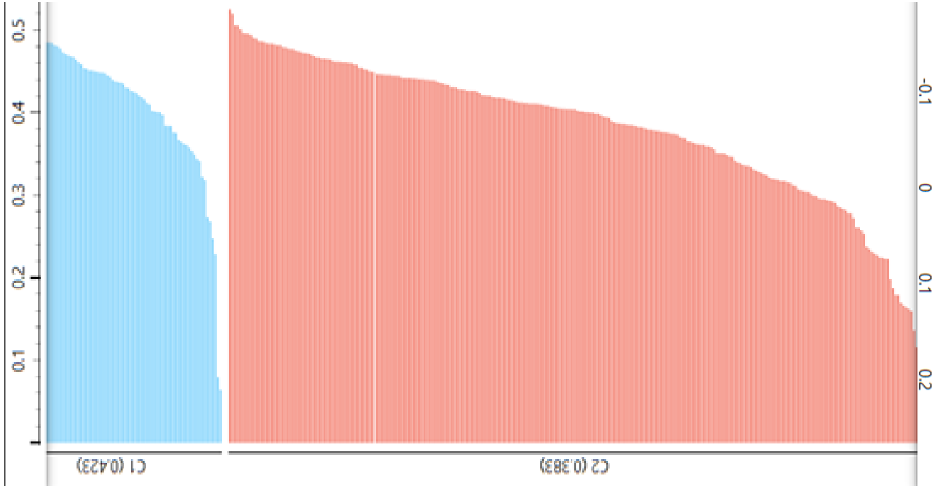
- Task 1: T-SNE shows better clustering capability by using features extracted from the specialized CNN, compared with the general-purpose CNN

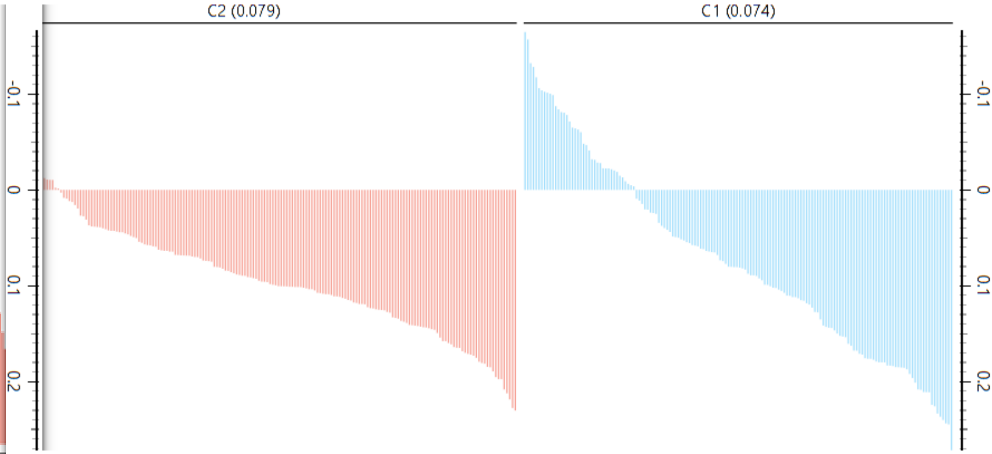


# Case study #2: Faces clustering/classification

- Task#1: The silhouette coefficient indicated 2 major groups in both cases.
    - However, for the general-purpose CNN there is no clear difference between groups.
    - For the specialized CNN groups are clearly separated by apparent gender
    - Hierarchical clustering corroborated with the above findings



# Case study #2: Faces clustering/classification

- Task 2: Classification task
  - The specialized CNN has only 128 features, but achieved better results

| Model | AUC | CA | F1 | Prec | Recall | MCC |
|---|---|---|---|---|---|---|
| Neural Network | 0.996 | 0.937 | 0.935 | 0.936 | 0.937 | 0.933 |
| SVM (1) | 0.992 | 0.939 | 0.933 | 0.929 | 0.939 | 0.936 |

  - The general-purpose CNN has 1000 features, but achieved results much lower than the previous case

| Model | AUC | CA | F1 | Prec | Recall | MCC |
|---|---|---|---|---|---|---|
| Neural Network | 0.943 | 0.609 | 0.599 | 0.610 | 0.609 | 0.588 |
| SVM (2) | 0.942 | 0.586 | 0.550 | 0.548 | 0.586 | 0.564 |