Image Classification for Inspection Machines:

a comprehensive review and application study

Università Cattolica del Sacro Cuore,

Applied Data Science For Banking and Finance

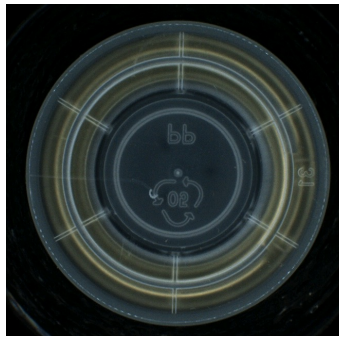
Elisa Salvi

AGENDA

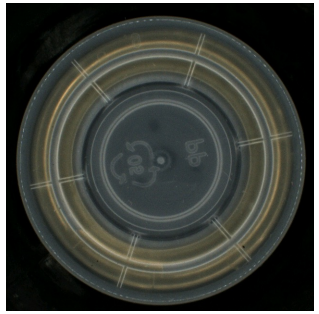
- ✓ Dataset with 9 types of Images
- ✓ Image Pre-processing:
  - Edge Detection
  - Circle Hough Transform
- ✓ Features Extraction and Selection:
  - RGB levels
  - SIFT
  - HOG
- ✓ Classification Algorithms for Computer Vision:
  - Decision Tree, Random Forest, SVM, KNN
  - Convolutional Neural Network
  - Vision Transformer
- ✓ Results
- ✓ GUI

MOTIVATION

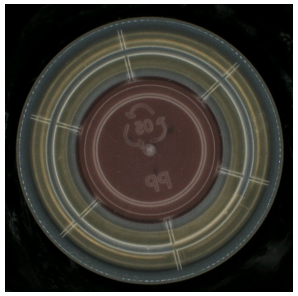
My internship project's main scope is to ***classify Flip-off's images*** acquired by the Inspection Machine department of ***Antares Vision S.p.A.***



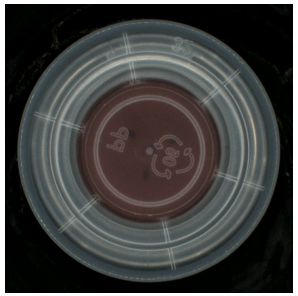
Trasparente-Nero
Oro-250 ml



Trasparente-Nero-Oro-1000 ml

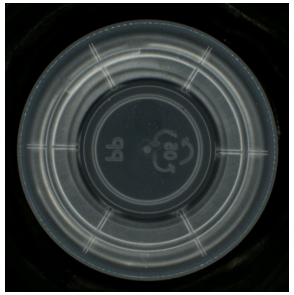


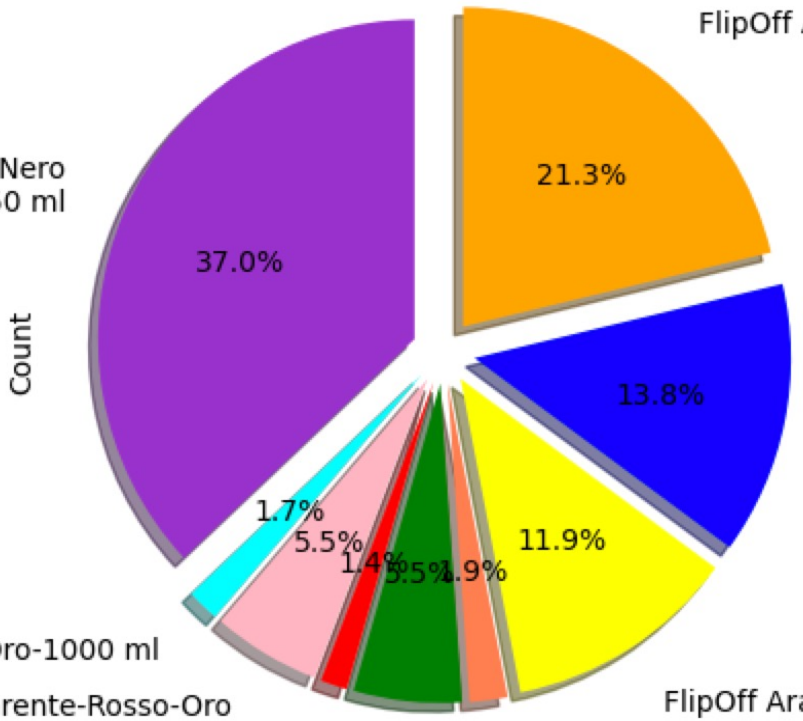
Trasparente-Rosso-Oro



Trasparente-Rosso-Argento

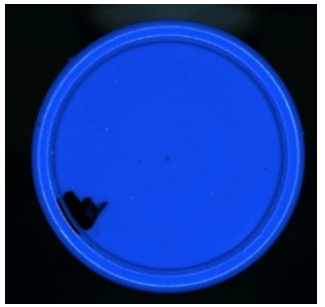
Trasparente-Nero-Argento



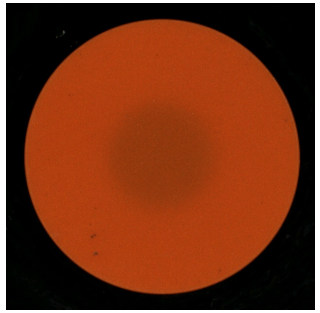


FlipOff Arancio con scritta

Flipoff Blu



FlipOff Arancio

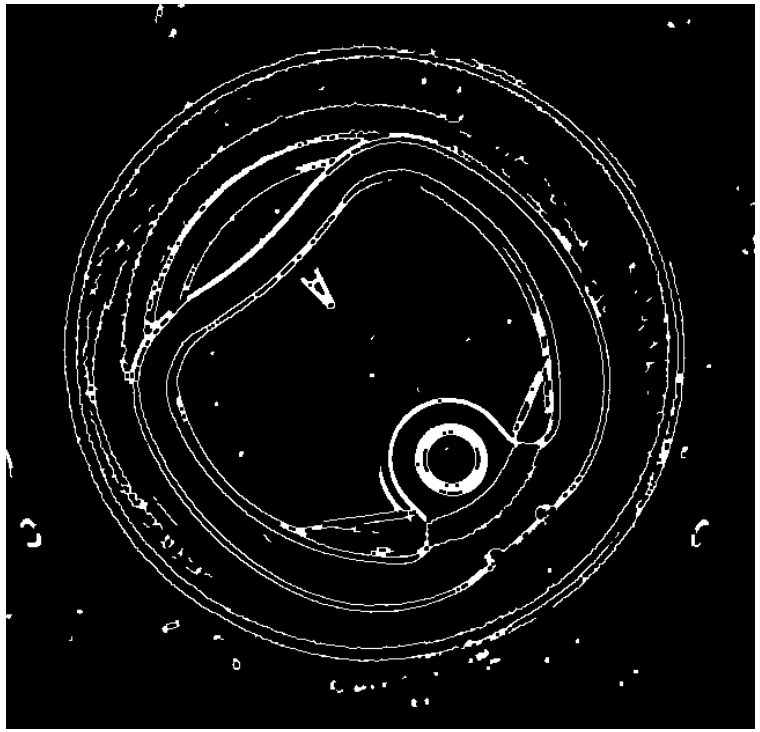


Alluminio-Anello-Alluminio

EDGE DETECTION

Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, such as sharp changes in the image brightness.

The points where the image brightness varies sharply are called the ***edges*** (or ***boundaries***) of the image.

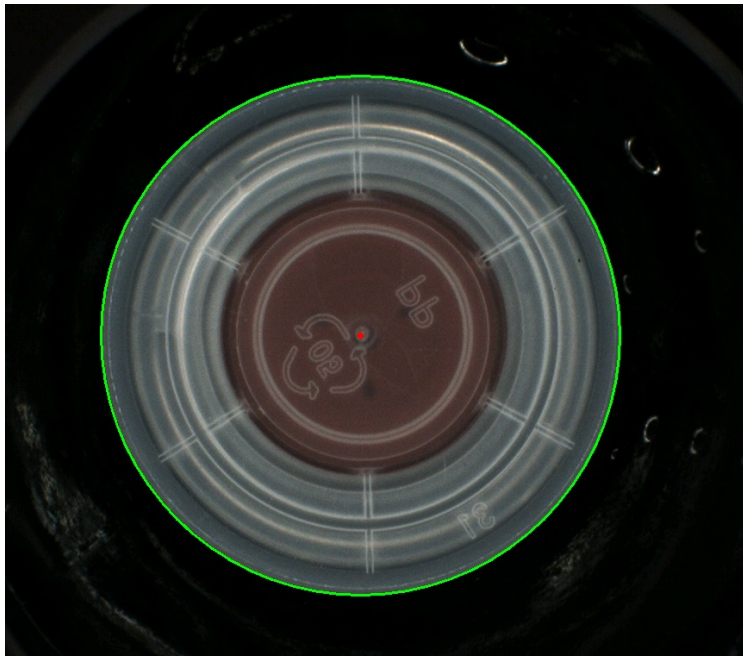


CIRCLE HOUGH TRANSFORM

The ***Hough Transform*** is an extraction technique to find the shape class of objects in an image even if in boundaries are present some imperfections or gaps.

The ***Circle Hough Transform*** is a feature extraction technique used for detecting circles in imperfect images.

The circle candidates are produced by "voting" in the Hough parameter space and then selecting local maxima in an accumulator matrix.



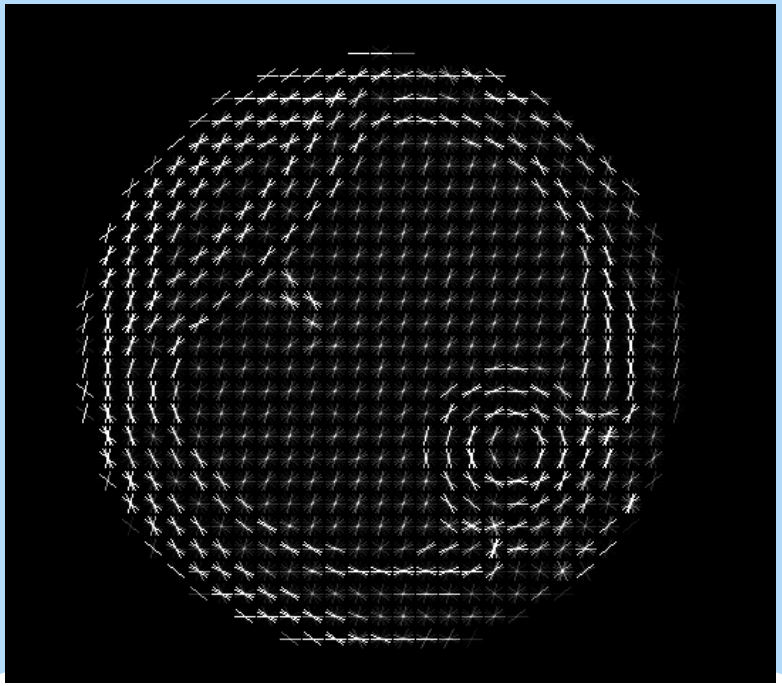
FEATURE EXTRACTION

- ***RGB*** mean, variance and standard deviation
- ***HOG*** min, max, mean and standard deviation
- ***SIFT*** mean and standard deviation

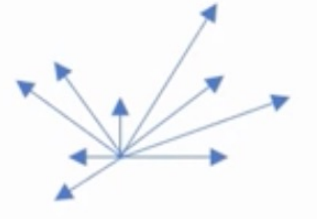
The features consider the internal circle area (80% of the initial radius) in order to expunge from the images useless shades and the background.

NOT ALL the features extracted are used for the models to avoid overfitting. Two feature selection algorithms facilitate this choice ranking the features importance: a XGBtree and a regression.

HISTOGRAM OF ORIENTED GRADIENTS



- HOG calculates the ***gradient*** of every 8x8 pixels cell in the image.
- The magnitude is higher when there is a sharp change in pixels' intensity, such as around the edges.
- Using the gradients and the Pythagoras theorem, the ***magnitude*** and ***direction*** for each pixel value are calculated.
- With this two information, an ***histogram*** is generated for each cell. The 9 bins represent the direction (0°-180°) and the height of the bins is the magnitude of the gradient.
- Each histogram is then transformed in ***a vector representation.***

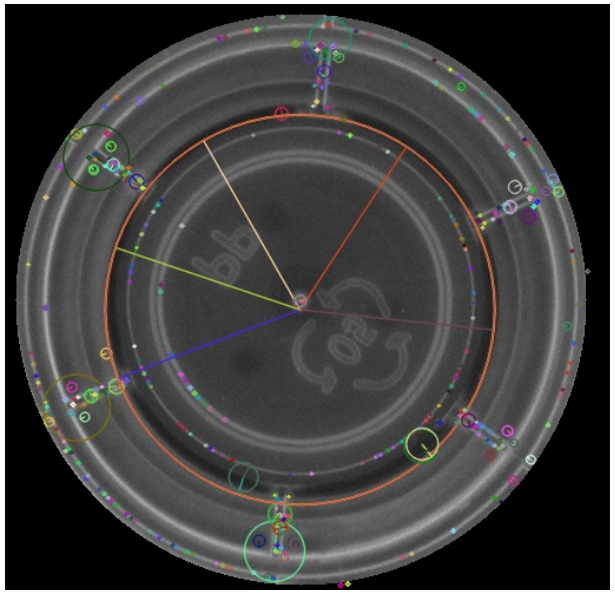


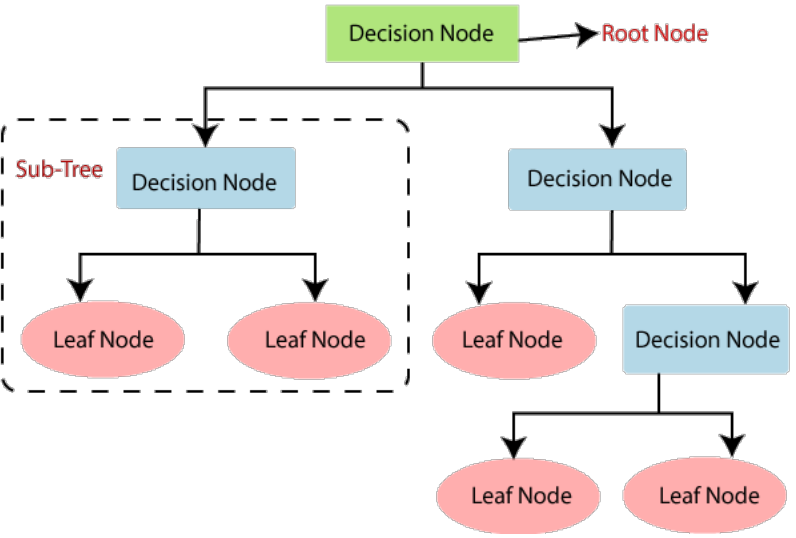
SCALE INVARIANT FEATURE TRASFORM

The ***SIFT*** is an algorithm used to detect and describe local features in digital images, invented by D. Lowe in 1999.

It locates certain ***key points*** and then describes them with quantitative information (so-called ***descriptors***) which can be used for object recognition.

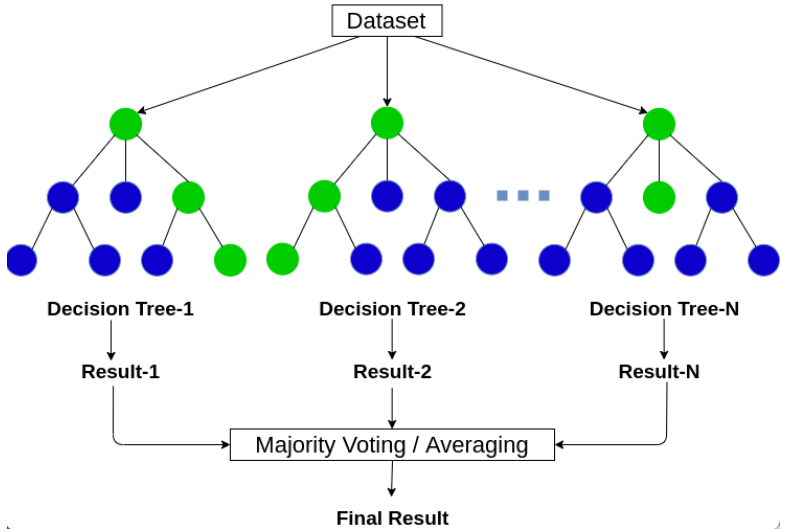
The descriptors are supposed to be ***invariant*** against various transformations (rotation, scaling, lightening, …) which might make images look different, although they represent the same object.

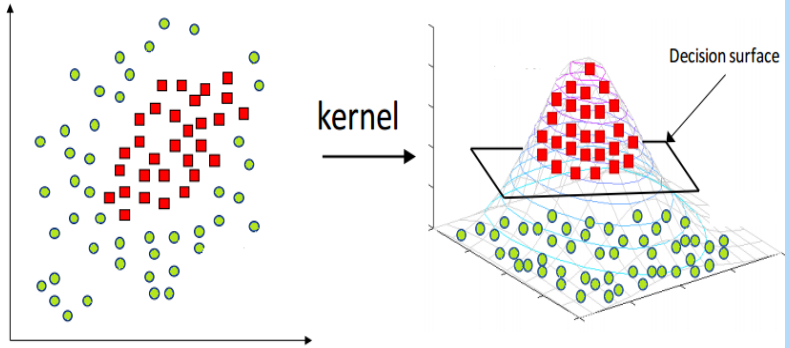


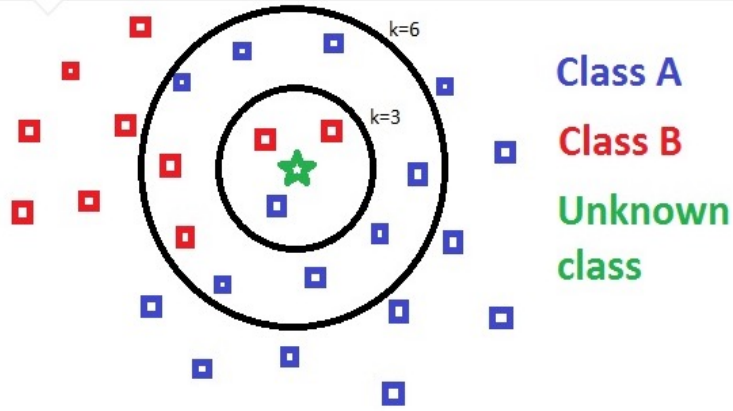


CLASSIFICATION ALGORITHMS

- *Decision Tree*
- *Random Forest*
- *Support Vector Machines*
- *K Nearest Neighbor*







PROBLEM DIVIDED IN TWO SUB-PROBLEMS

- Black-Gold Transparent 250 ml;
- Black-Gold Transparent 1000 ml;

} Black-Gold

- Red-Gold Transparent;
- Red-Silver Transparent;
- Black-Silver Transparent;

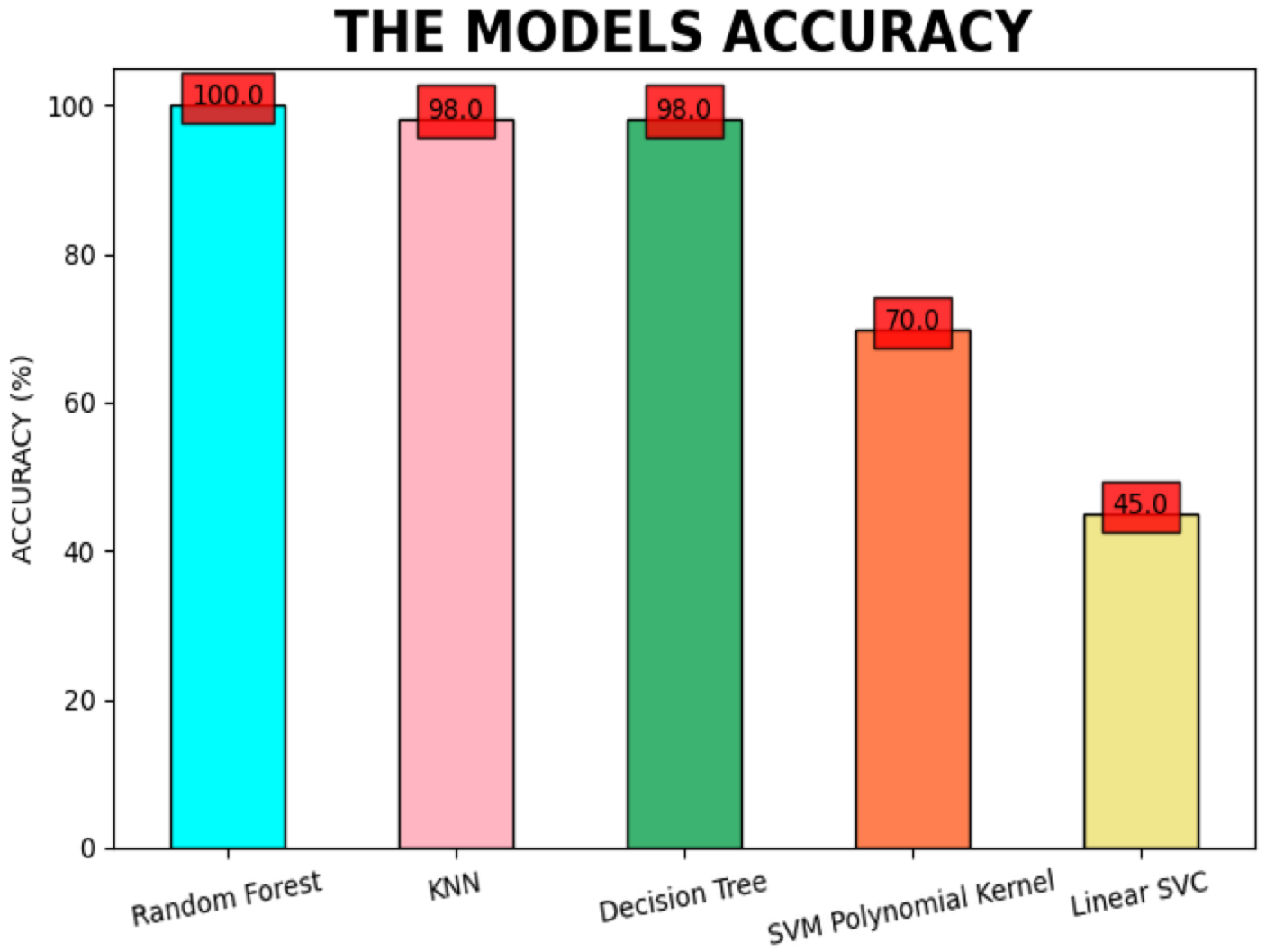
} Transparent

- Aluminium;
- Orange;
- Blue;
- Orange with text.

CLASSIFICATION GRUPPING TRANSPARENT FLIPOFF

FEATURE USED:

- ***RGB variance***
- ***HOG standard deviation***
- ***Circle Area***
- ***SIFT mean***

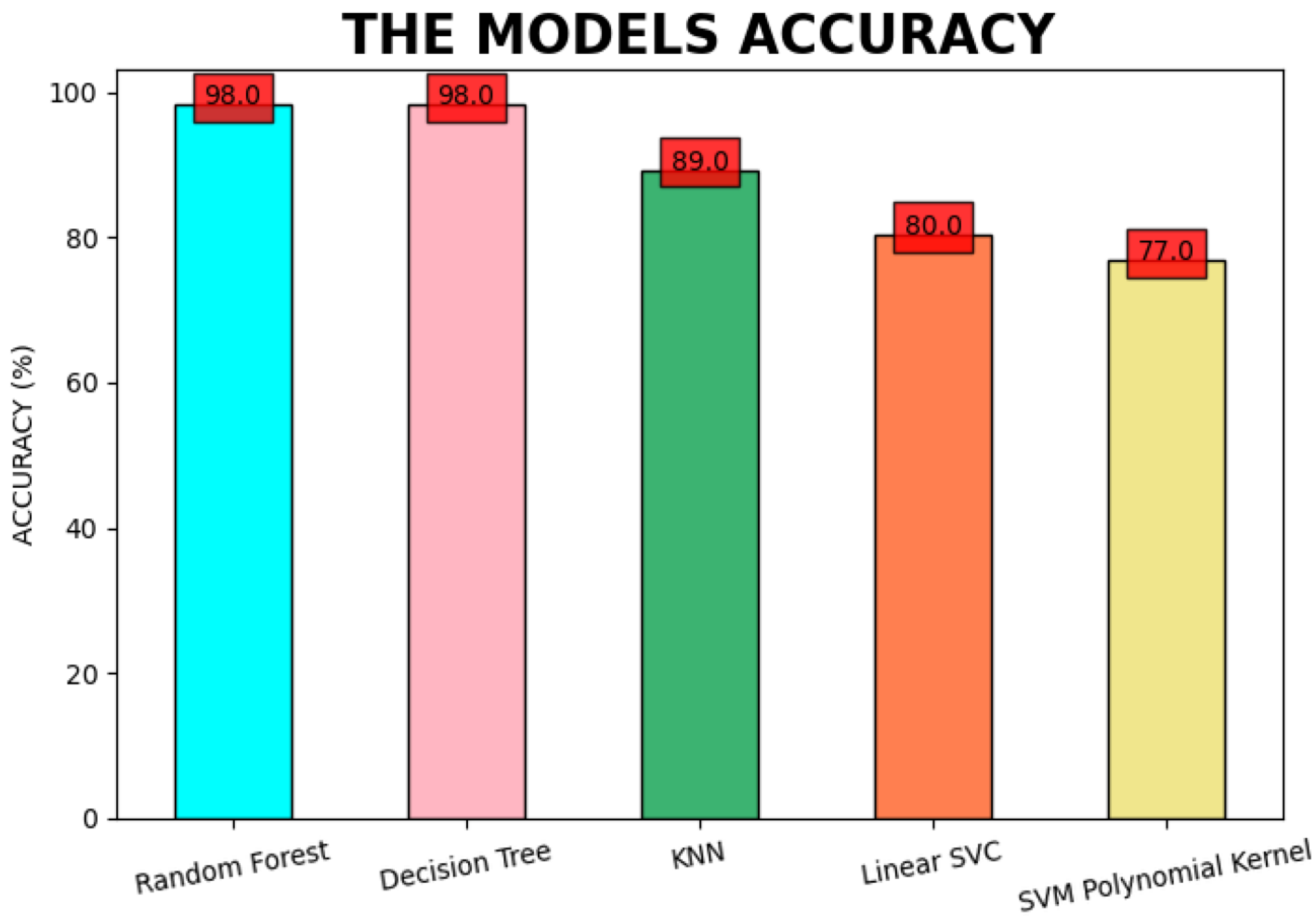


NB: $\text{Accuracy} = \dfrac{\text{Number of correct predictions}}{\text{Total number of predictions}}$

CLASSIFICATION TRANSPARENT FLIPOFF

FEATURE USED:

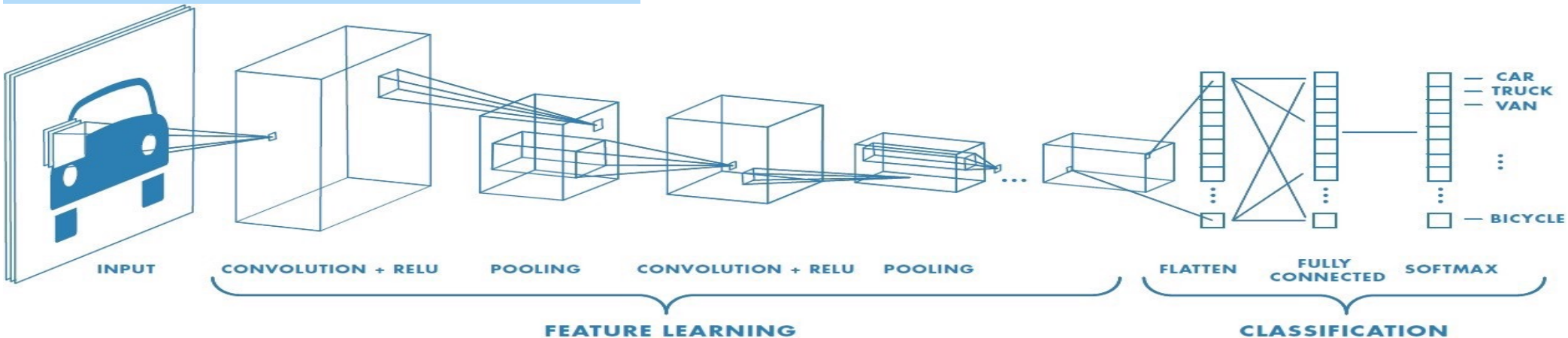
- ***R and G mean***
- ***B standard deviation***
- ***R variance***
- ***HOG minimum***
- ***SIFT standard deviation***



CONVOLUTIONAL NEURAL NETWORK

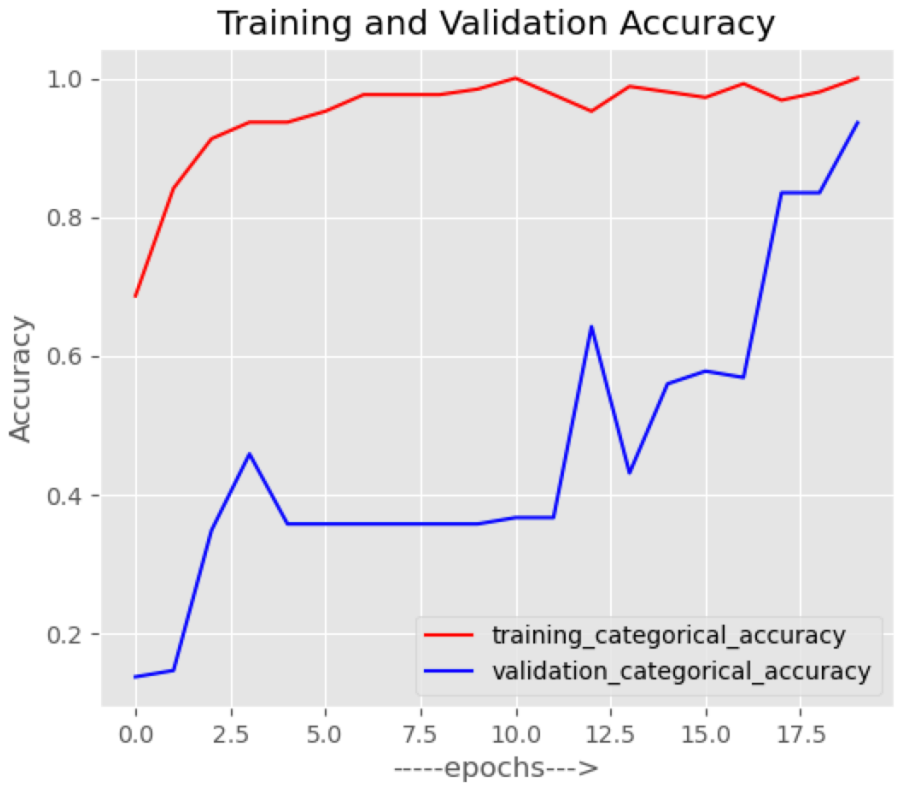
Multiple, sequential layers that are directly fed with raw data.

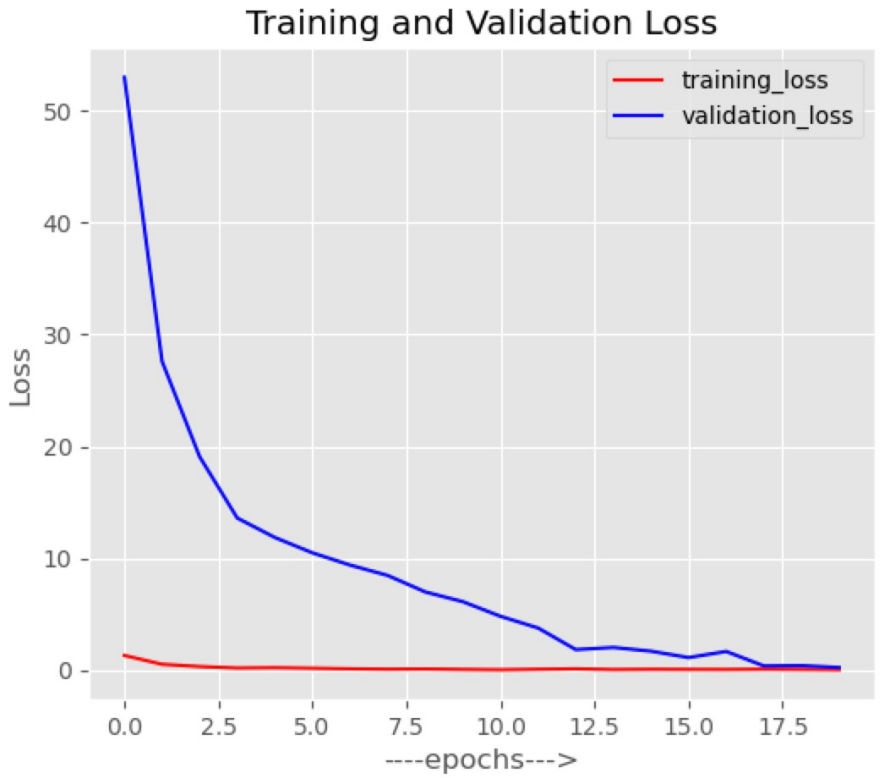
- ***Convolutional*** layers: apply Filters to extract features and discard irrelevant information with the ReLu.
- ***Max Pooling*** layer: reduces the dimensionality of the output.
- Layers are repeated to extract higher-level features.
- ***Dense*** layer: aggregates information before classification.
- The "***SoftMax***" activation function is used to convert the output into the membership probability.



CNN CLASSIFICATION:

accuracy set from the GUI at 90%





Accuracy:	Number of epochs:	Time:
30%	4	35 sec.
50%	14	2 min.
60%	17	2 min. and 25 sec.
70%	13	2 min.
80%	16	2 min. and 20 sec.
90%	20	3 min.

macOS Big Sur
Versione 11.6.8

MacBook Pro (16-inch, 2019)
Processore 2,4 GHz Intel Core i9 8 core
Memoria 64 GB 2667 MHz DDR4
Scheda grafica Intel UHD Graphics 630 1536 MB

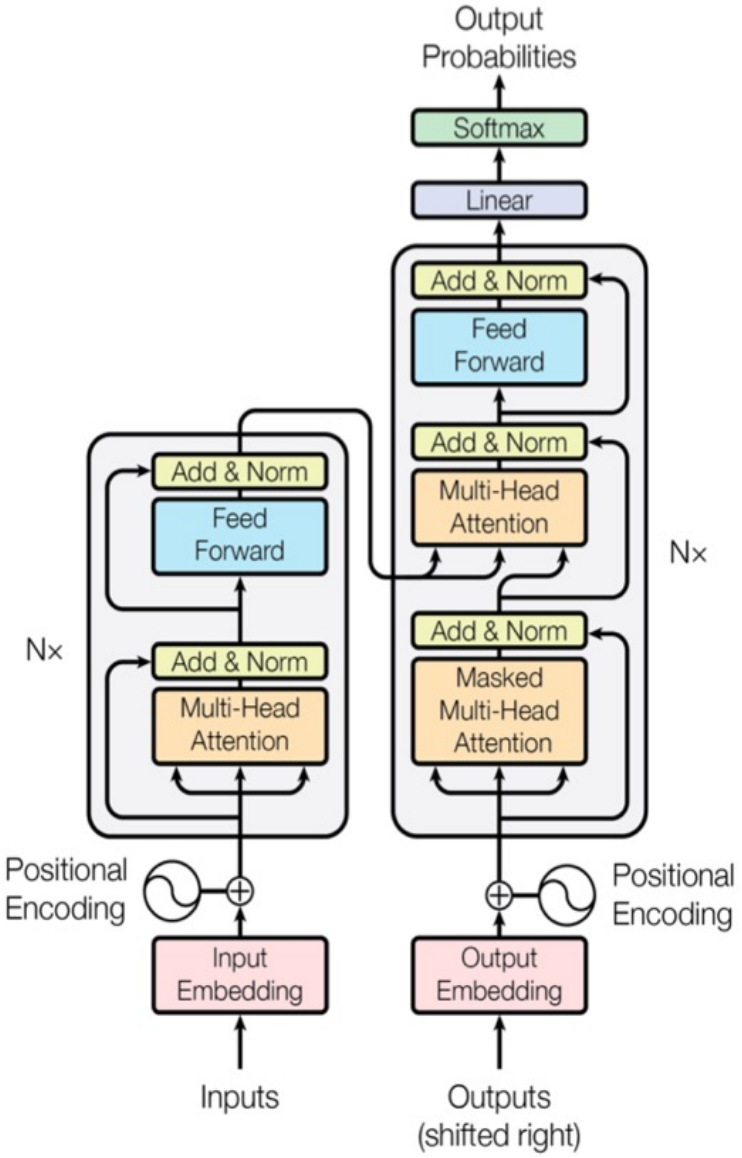
TRANSFORMER

CNNs assume that neighbour pixels are more important than distant pixels.

"***Self-Attention***" mechanism captures long-range relations between pixels.

In 2021 Bazi used a ***ViTransformer*** for image classification for the first time.

- ***INPUT EMBEDDING:*** input is broken down in ***token*** that are represented by vectors, called "***embeddings***". These vectors lie on a a multi-dimensional space, called ***"embedding space***", that aims to capture relationship between the pixels based on their distance.



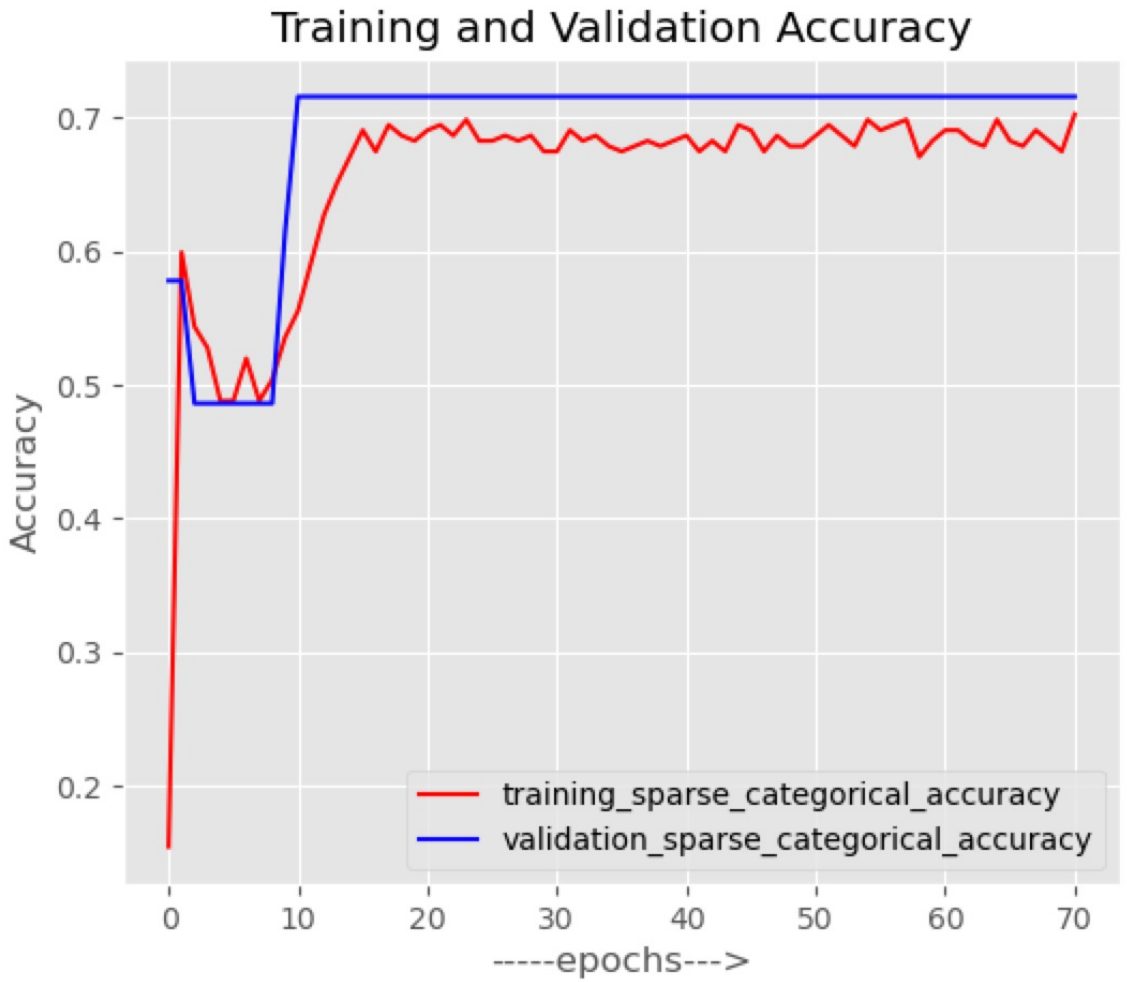
- ***THE ENCODER:*** takes as input the images and the pixels positions. is composed of ***Multi-Head Attention layers*** applies multiple linear transformations to find information from different representation subspaces at different positions and several characteristics of the pixels.

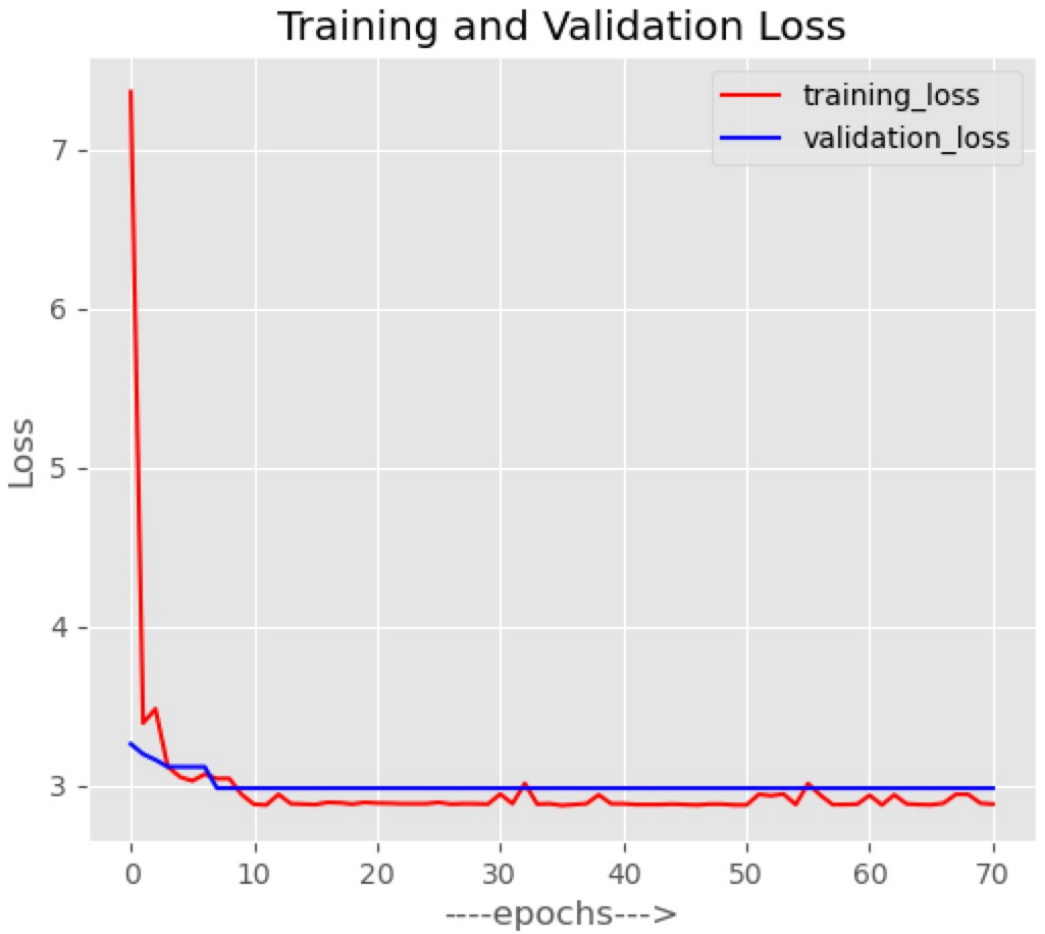
- ***THE DECODER:*** takes as input the target pixels (shifted one time step to the right) and the output of the encoder for N times.

- Finally, a "***Softmax***" classification layer is used for classification.

VI-TRANSFORMER CLASSIFICATION

accuracy set from the GUI at 70%





Accuracy:	Number of epochs:	Time:
30%	2	18 sec.
50%	2	18 sec.
60%	9	1 min.
70%	70	8 min.

GRAFICAL USER INTERFACE WITH "TKINTER"

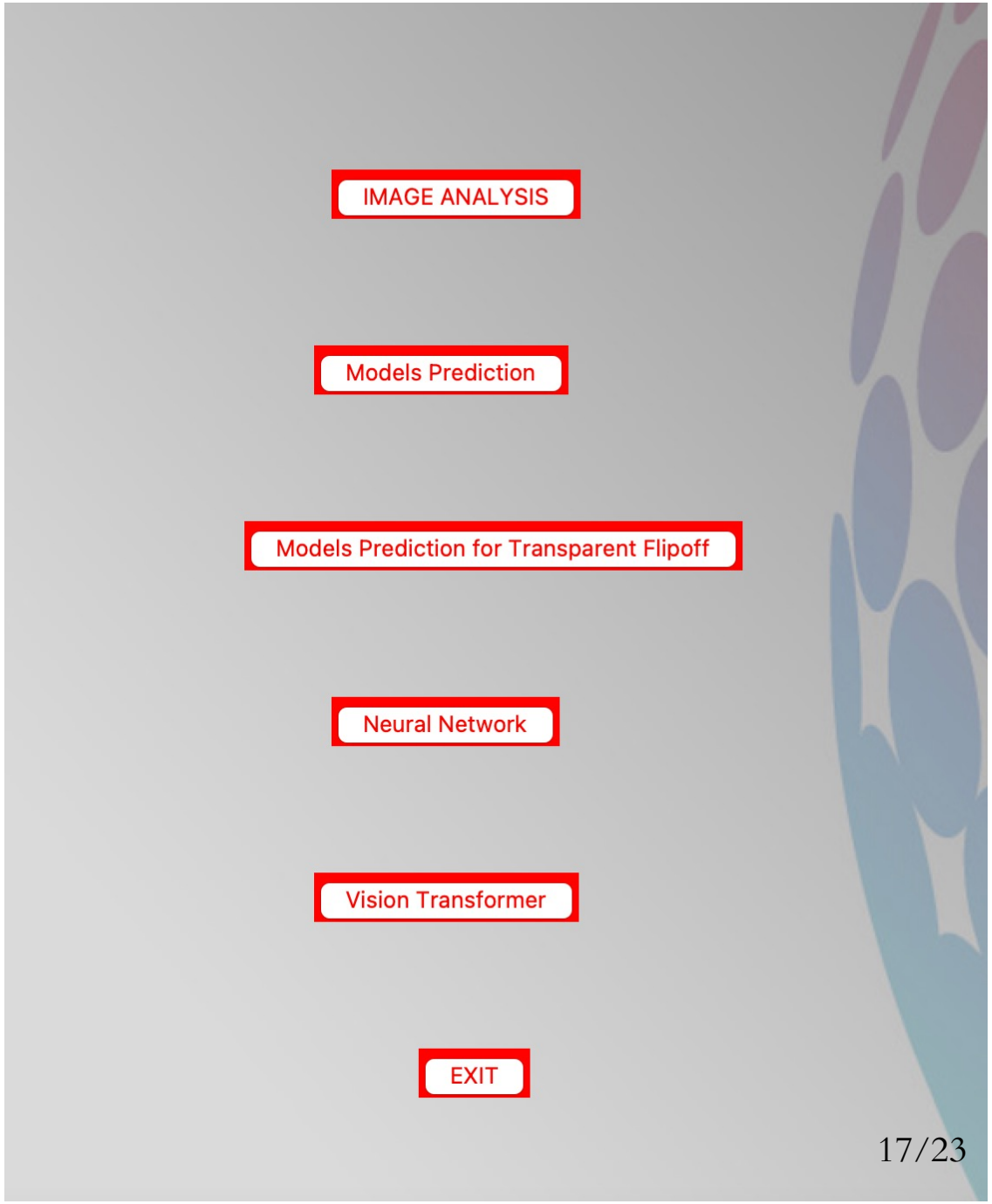


IMAGE ANALYSIS

THE DATASET

EDGE DETECTION

HOUGH CIRCLE

80% MASK

HOG?

HISTOGRAM OF ORIENTED GRADIENTS

SIFT?

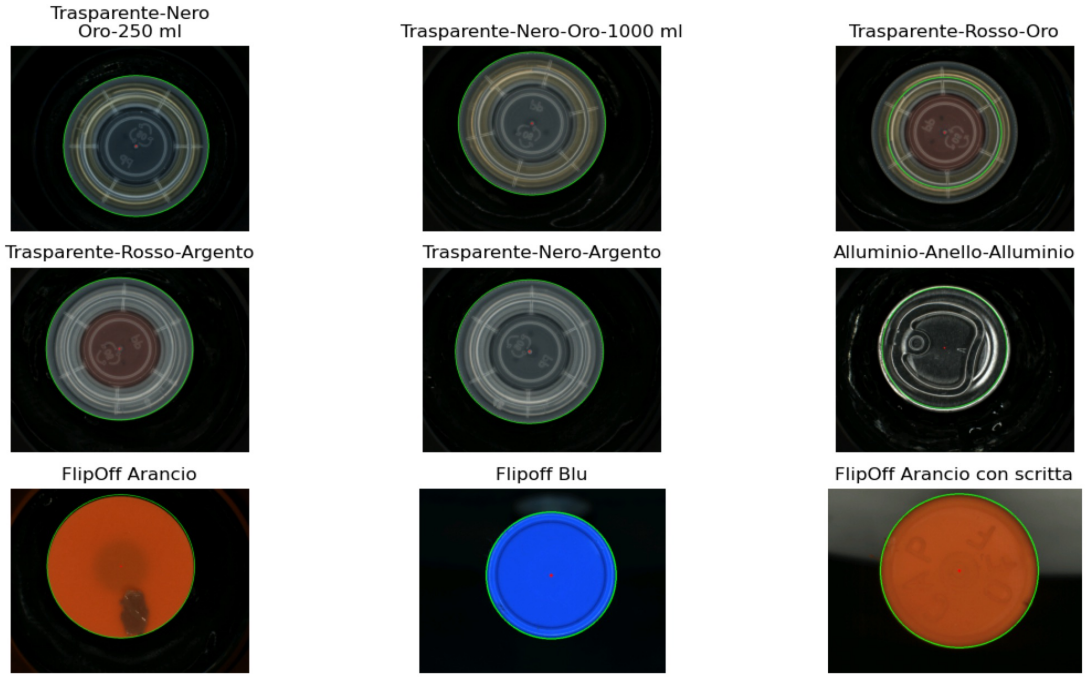
SCALE INVARIANT FEATURE TRANSFORM

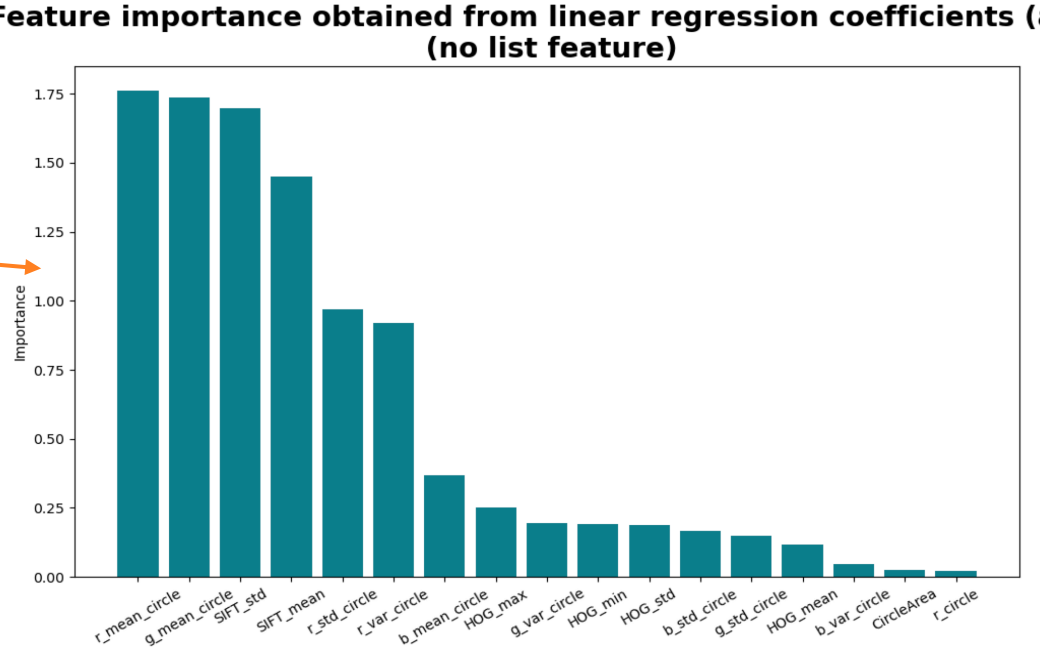
FEATURE EXTRACTION

FEATURE IMPORTANCE lin_reg

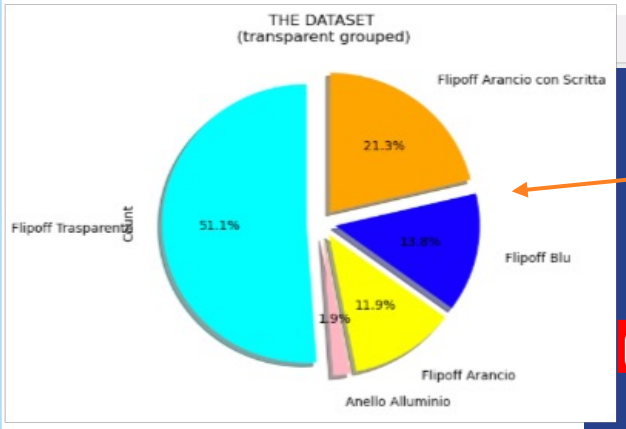
FEATURE IMPORTANCE XGBClassifier

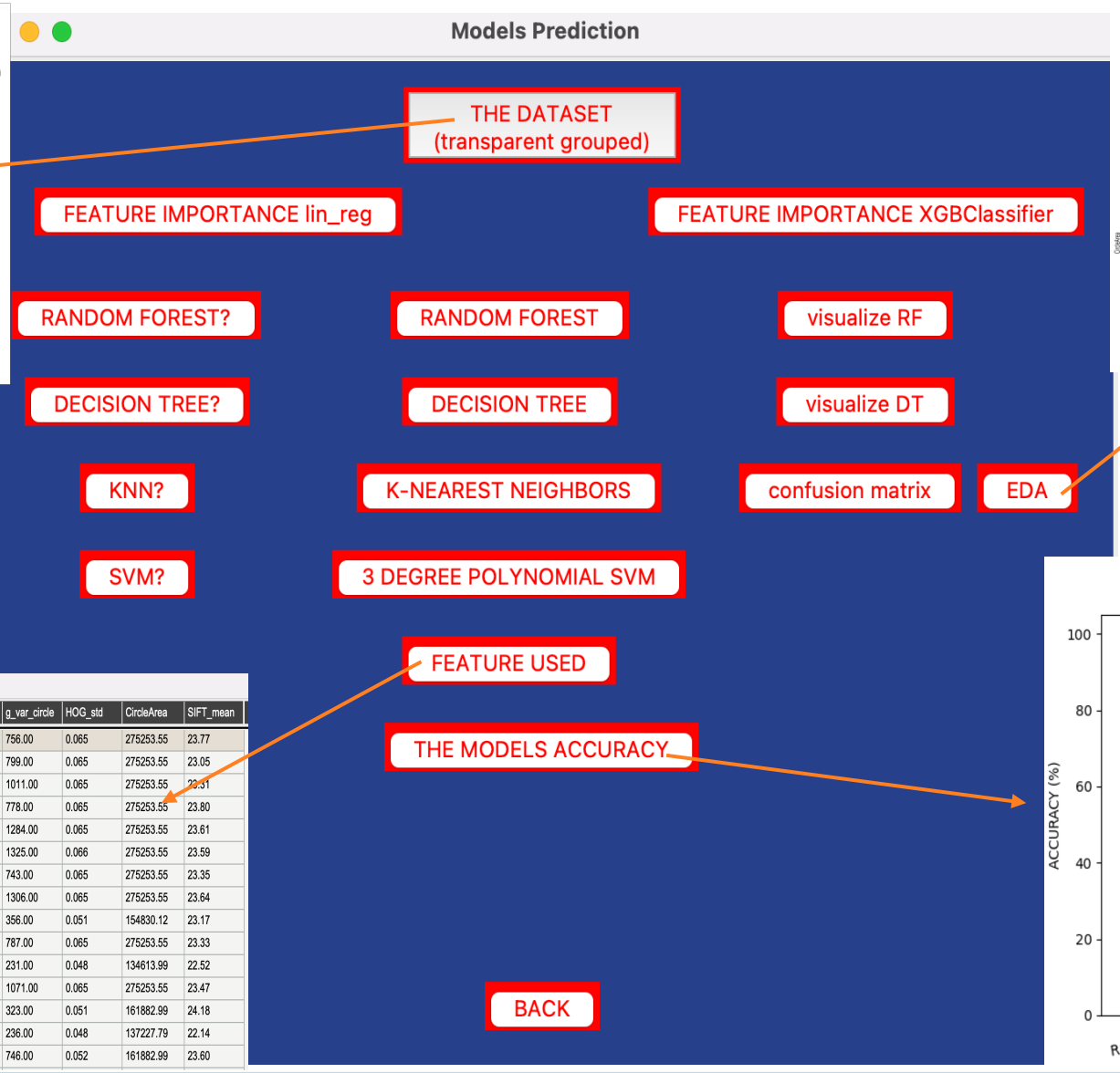
BACK

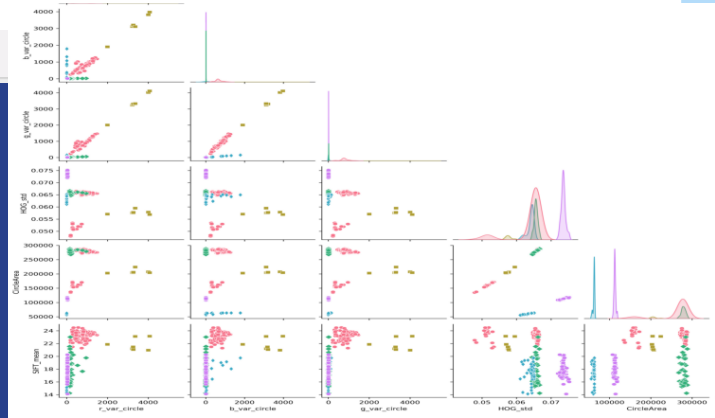




THE SUB-PROBLEMS VISUALISATION

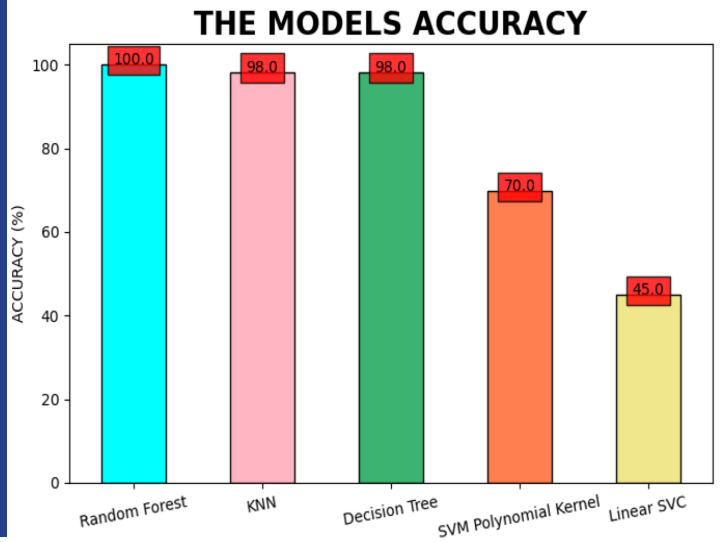




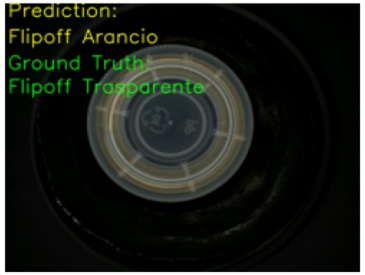


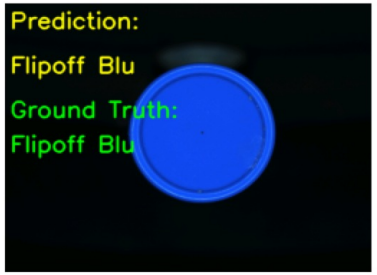
FEATURE USED

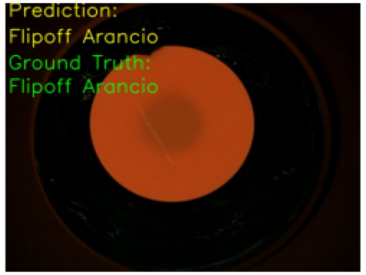
	New Target Name	Path	r_var_circle	b_var_circle	g_var_circle	HOG_std	CircleArea	SIFT_mean
1	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	764.00	624.00	756.00	0.065	275253.55	23.77
2	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	803.00	652.00	799.00	0.065	275253.55	23.05
3	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	986.00	878.00	1011.00	0.065	275253.55	23.31
4	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	758.00	668.00	778.00	0.065	275253.55	23.80
5	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	1271.00	1093.00	1284.00	0.065	275253.55	23.61
6	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	1288.00	1147.00	1325.00	0.066	275253.55	23.59
7	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	740.00	610.00	743.00	0.065	275253.55	23.35
8	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	1269.00	1149.00	1306.00	0.065	275253.55	23.64
9	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	311.00	360.00	356.00	0.051	154830.12	23.17
10	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	806.00	623.00	787.00	0.065	275253.55	23.33
11	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	194.00	250.00	231.00	0.048	134613.99	22.52
12	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	1083.00	848.00	1071.00	0.065	275253.55	23.47
13	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	287.00	317.00	323.00	0.051	161882.99	24.18
14	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	198.00	255.00	236.00	0.048	137227.79	22.14
15	Flipoff Trasparente	/Users/davidesalvi/Desktop/stage_project_py/images	664.00	721.00	746.00	0.052	161882.99	23.60

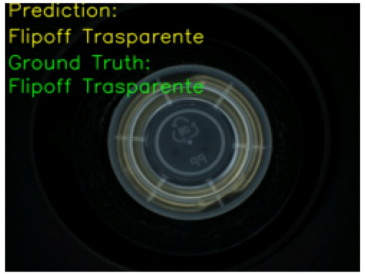


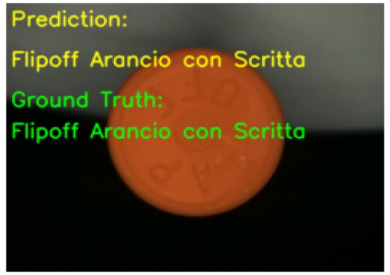
DECISION TREE

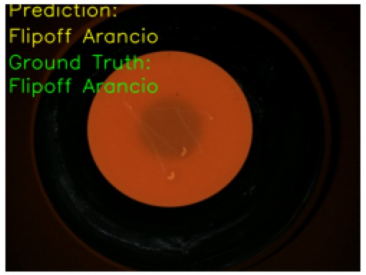


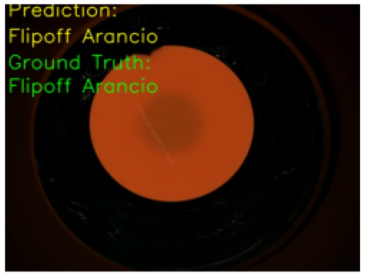


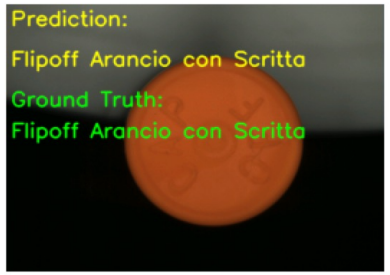












CNN and ViTransformer GUI

Insert the accuracy in the box above
(0.1 for 10%)
and PRESS this button

How does it work?

BACK

ACCURACY ANALYSIS

Training and Validation Accuracy

Training and Validation Loss

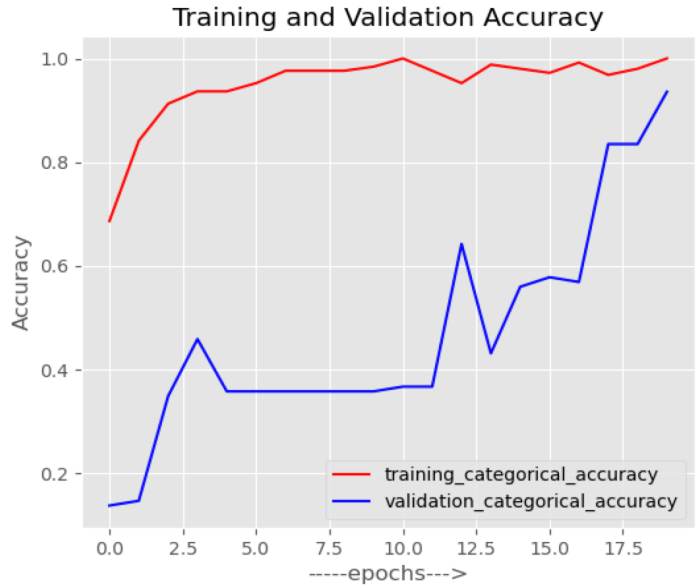
TIME

The time needed for the CNN training with an accuracy of 90% is:
173.4 sec.

CNN model prediction

GT vs CNN PREDICTIONS

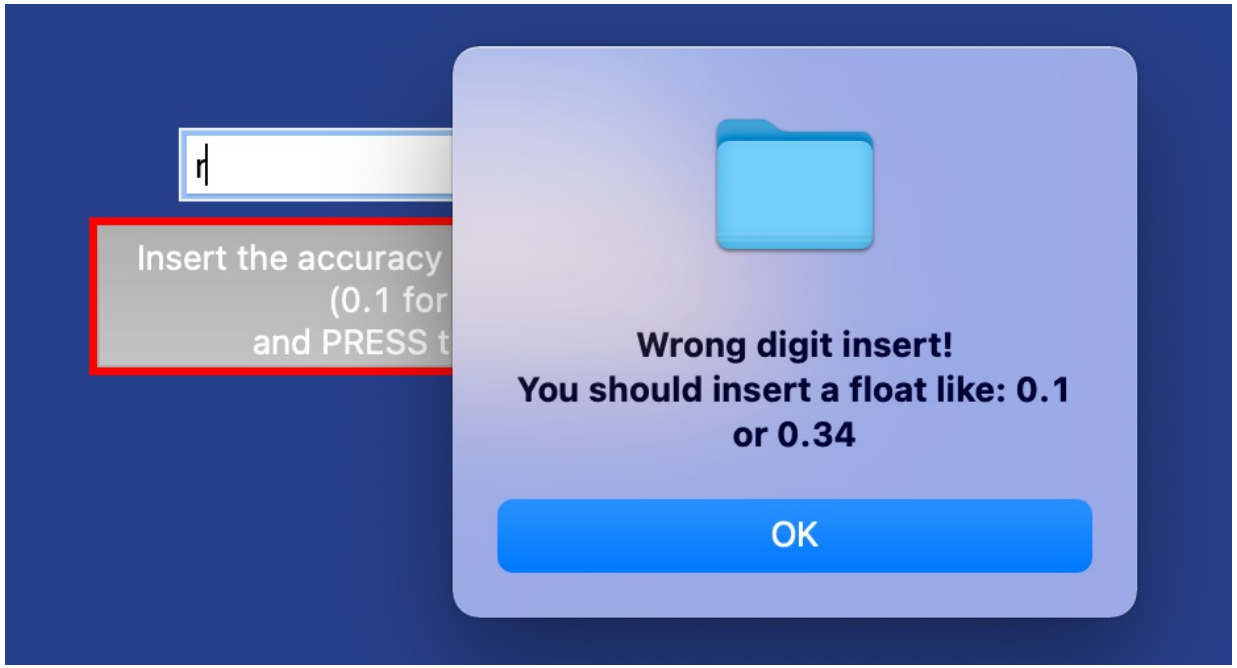
BACK

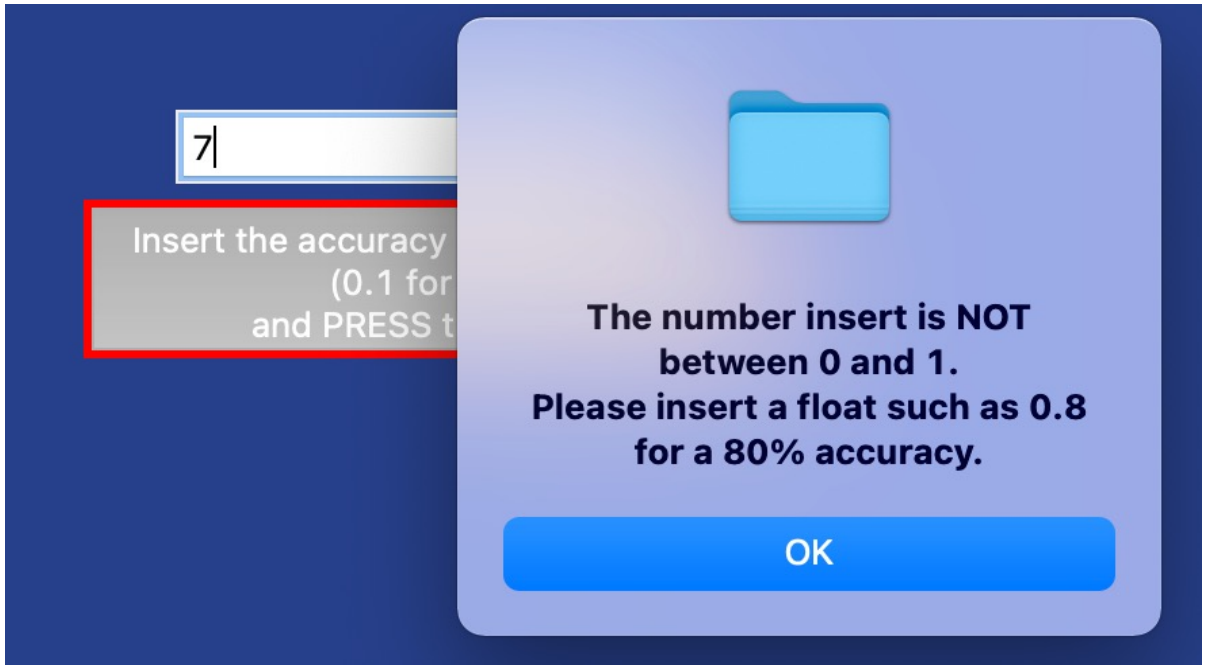


	category	GroundTruth	file_name
38	0	0	Product_Prod0171_Cam00_Pre00_Img000_ResM031
39	0	0	Product_Prod0172_Cam00_Pre00_Img000_ResM031
40	0	0	Product_Prod0173_Cam00_Pre00_Img000_ResM031
41	1	1	Product_Prod0009_Cam00_Pre00_Img000_ResM001
42	4	1	Product_Prod0104_Cam00_Pre00_Img000_ResM001
43	8	2	Product_Prod0042_Cam00_Pre00_Img000_ResM001
44	2	2	Product_Prod0045_Cam00_Pre00_Img000_ResM001
45	2	2	Product_Prod0069_Cam00_Pre00_Img000_ResM001
46	2	2	Product_Prod0095_Cam00_Pre00_Img000_ResM001
47	8	2	Product_Prod0097_Cam00_Pre00_Img000_ResM001
48	2	2	Product_Prod0103_Cam00_Pre00_Img000_ResM001
49	8	3	Product_Prod0018_Cam00_Pre00_Img000_ResM001
50	4	3	Product_Prod0062_Cam00_Pre00_Img000_ResM001
51	4	4	Product_Prod0024_Cam00_Pre00_Img000_ResM001
52	4	4	Product_Prod0027_Cam00_Pre00_Img000_ResM001

TOOL ROBUSTNESS

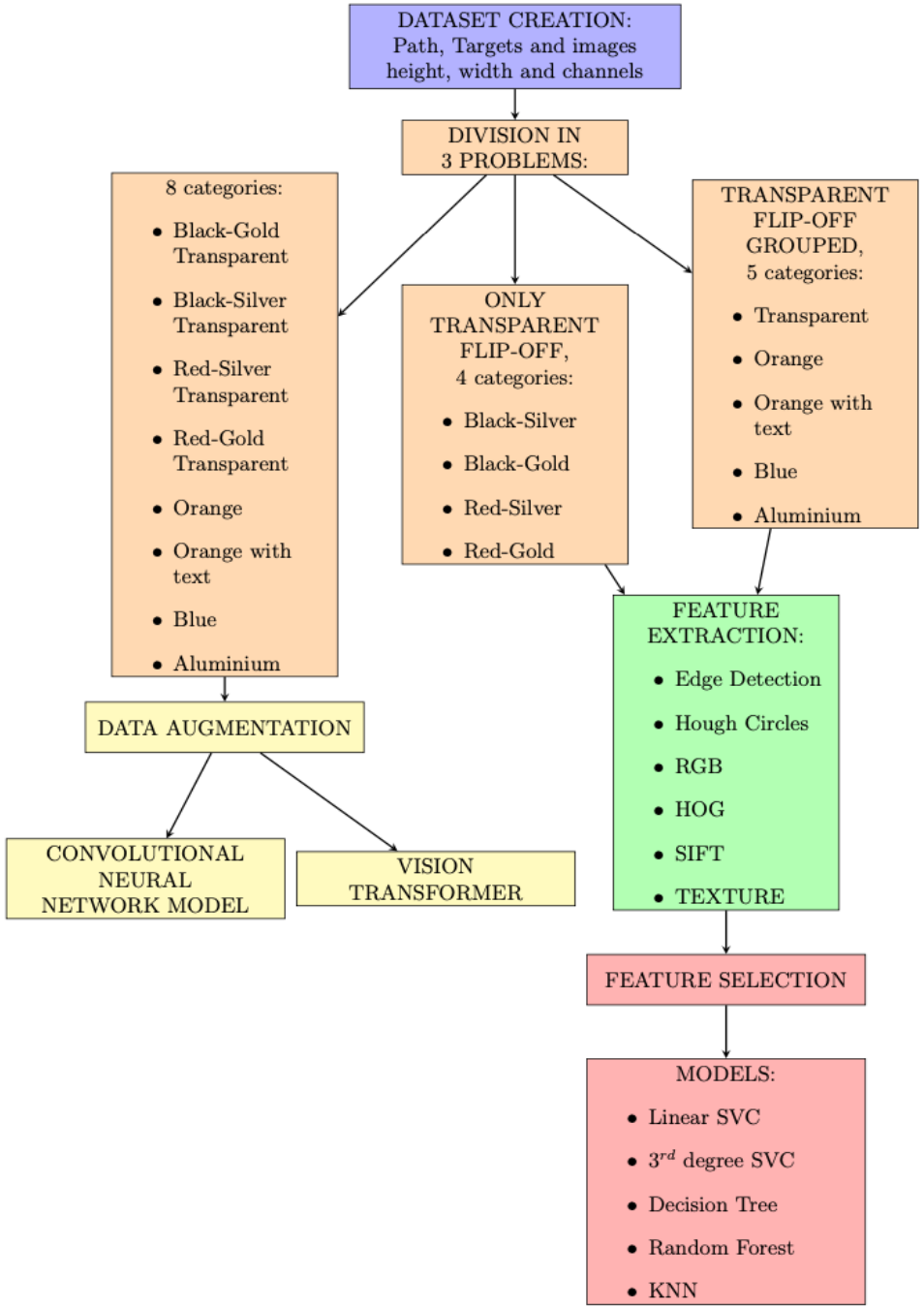
Exceptions / Warnings






```
except ValueError:
    tk.messagebox.showerror(title="ERROR", message="Wrong digit insert! \n You should insert a float like: 0.1 or 0.34")
```

THESIS PROJECT FLOWCHART



THANK YOU!

Elisa Salvi