UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

PROGRAMA EXPERTO EN INTELIGENCIA ARTIFICIAL

TRABAJO FIN DE PROGRAMA

Face Recognition System Using Facenet in Keras and application of Super Learner Ensembles

NOMBRE:

Juan Luis Trejo Medina

CURSO 2019-2020

TÍTULO: Face Recognition System Using Facenet in Keras and application of Super Learner Ensembles

AUTORES: Juan Luis Trejo Medina

TITULACIÓN: Artificial Intelligence Expert Program

DIRECTOR DEL PROYECTO: María Borbones

FECHA: 21st June 2020

Abstract

The objective of this work is to develop the most efficient (considering accuracy in the probability of prediction and the computational requirement) architecture for face recognition classifier using Multilayer Convolutional Networks for face identification, Facenet in Keras which generates the embeddings and finally a classifier optimized using stacked and Super Learner ensembles comparing the performance of both with different datasets.

For that have been combined the best performance algorithms in each of the 3 steps, in face identification (MTCNN), creating the embeddings (Facenet) and finally creating a stacked classifier with machine learning (Stacked and Superlearner) looking as a combined accuracy higher than 90%

Facial recognition is key in sensible infrastructures like Nuclear, Water, Power plants, Petrochemicals, Refineries etc so the goal is creating an algorithm that can be used to fasten the access and the security in these key facilities which has a flow of thousands of people daily being key the accuracy and velocity as this will represent an enhancement of the productivity and security in installation that cannot afford any risk because the social and healthy risks that they represent.

As an average for a facility with 1000 operators in removing the input lathe we calculate a saving of 30 seg in getting in and 30 seg in getting out of the plant which means 1000 minutes per day, 22000 minutes per month and 264000 minutes per year (4400 hours)

As a conclusion has been created an algorithm with levels of accuracies higher than 90% which requires low computational charge as it leverages the most efficient algorithms (State of art on each step) and pre-trained models. This can be used not only for sensible infrastructures but as well for marketing campaigns in which personalize the customer experiences when faces are recognized with this level of accuracy in stores or any center that concentrate a limited number of people.

Index

Sumario

Figures index:

Table Index

1.- Introduction

Facial recognition consists in detecting identifying and verifying the faces targeted. In the face detection process the goal is to detect faces in a video and or photographs being static or dynamic motion in the detection of the face

The second step is the face digitization and in our project is the picture vectorization which create a representation through vectors of each pictures.

Finally it has to identify the face and recognize it among other faces provided.

Humans recognize humans looking faces naturally and what we will do through this project is to recognize faces simulating the actions performed by humans but doing through algorithms which will follow the same steps that humans do in face detection, identification and final matching.

The detection of the face will be done from the bigger picture to the bounding boxes which enclose the face, from that bounding will be detected some key features in the faces which will individualize them and create unique landmarks. This key features will be 5 key points in the two eyes, nose, two points in the mouth. After this vectorization will be performed the classification of the faces targeted being the goal of this project to investigate the performance of the classification of the faces through different models of machine learning instead of using the triple lost model developed in the Facenet algorithm.

Actually there are applications which leverage these models in surveillance cameras, Smart phones. In 2017 Apple created an app highly criticized in China as it was struggling to differentiate Chinese faces.

1.1 Most relevant development

There are different companies which are developing technologies being the top five companies (GAFAM), Google, Apple, Facebook, Amazon and Microsoft the ones which have progressed more plus other smaller companies which have

developed good performing solutions being the ones developed as performance as shown below:

Google: Facenet. Identification scores 99.63%

Facebook: DeepFace. Identification scores 97.25%

Amazon: Rekognition. Identification scores 97.25%

Chinese University of Hong Kong: Identification scores 98.52%

The score obtained by humans is 97.53% so the systems developed performs better than the human.

All these systems use a high computational charge and some of them struggle as well failing depending of the races. In this project the computational load will be challenged executing the classification with Machine learning models and testing the final model with different races. This is an area where companies like Microsoft, IBM, and China-based Megvii (FACE++) struggle to identify women with darker-skin women compared to lighter-skin men. As well Amazon model struggle to identify in 2018 28 members of US Congress as people which were jailed for crimes.

Facial recognition systems has kept improving during the last 4 years manly due to the evolvement of the convolutional network algorithms

1.2.- Face Recognition applications

Facial recognition has multiple applications which are growing as the technology keep improving in reliability and accuracy being some of the more relevant:

1.2.1 Security

Like Border checks, in airports with passport control which is an application getting more a more users

Another application is the face recognition of the staff in sensible infrastructures where thousands of people getting in certain facilities like power, water, nuclear

plants and petrochemical facilities. This represent an enhancement in the productivity of the operators because the saving of time and an increase in term of security and current Covid risk as it will facilitate quicker entrances avoiding the bottlenecks of people in the pick hours.

Police as well use facial recognition for driven licenses identification in most of the USA states and it is used for terrorism attack as the one executed in Brussels in 2016 where the FBI identified the terrorist tanks through their facial recognition software.

Through CCTV, facial recognition algorithms can help to find missed people, identify or track criminals movements and avoid future criminal activities as well.

1.2.2.- Marketing and retail

There are several applications in retail where face recognition algorithms is exploding the opportunity of driving the customer procurement process knowing their behaviors. Shopping with face recognition can identify customers and link them with their social media data, outcoming preferences and suggesting consequently products as per their identified preferences. This can be done placing cameras strategically in the stores manage through a SW that can provide strategic information.

Another application is selfie payment which facilitate the payment of users just with the face recognition. This is in development and testing process and is growing in parallel with the evolution of the face recognition algorithms.

1.3.- Project structure

Objective of the project is to architect a workflow which combine “state of the art” algorithms in face detection, and face identification using less computational requirement combining deep learning with machine learning algorithms and obtaining levels of accuracy comparable to Deep learning.

It has been structure following the flow of algorithms that has been used putting special emphasis in the development of the convolutional neural networks as

the key driver in the visual recognition and replication of the human vision. Convolutional Network is one of technologies in Artificial Intelligence with more potential considering all the applications not only in visual recognition but in other areas like Natural Language Learning being one of the areas that I have wanted to dig and learn deeper as part of this Postgrade.

Project is structured as follow:

1. Convolutional Neural Network. How it works and the characteristics to consider in the creation and optimization of a Deep Convolutional network, including as part of this project a detailed development of this technology key in the AI future evolvement.

2. MTCNN, Multilayer Convolutional Neural Network. The face detection part has been developed with the MTCNN algorithm which is the "state of the art" in face detection which optimize the face detection with face alignment

3. Facenet. It is a deep convolutional network designed by Google, Florian Schroff, in 2015 paper tittled, *"Facenet: A Unified Embedding for Face Recognition and Clustering"* trained to solve face verification, recognition and clustering problem with efficiently at scale. With this model are generated the embedding on the faces detected through the MTCNN framework. From Facenet we'll obtain the embedding which are the vectorization of the images evaluated and with from this numerical matrices rather than continue with the face identification using the Triplet loss model included as part of the Facenet model it has been used Machine learning algorithms (Stacked models)

4. Stacked Models (Deep Super Learner). The Super Learner is a Stacking ensemble algorithm which aim to improve the accuracy of the machine learning algorithms competing with the performance of deep neural networks. In this part of the project reside the main goal which is demonstrate accuracies higher than 90% and closer to the performance of Deep Neural Network through machine learning models which bring

benefits of Convergence, computational requirement, transparency, understandability and Interpretability of results.

Finally we summarize the conclusions obtained in the project and is detailed an application in a real environment.

2.- Problem statement. Convolutional Neural Network CNN

Convolutional Neural network is a supervised deep learning algorithm which process the information through several layers imitating the visual cortex of humans identifying different features of the images.

CNN contains several hidden layers being the first ones those which identify the simplest features line lines, curves… getting more a more specialized when getting closer to the deeper layers identifying more complex features like faces and animal's figures.

The neuronal network has to identify within an image objects and requires a lot of images through which learn and generalize well the objects.

As inputs of our neuronal network we will take the pixels of the image that will become neurons. For example, for an image of 28x28 pixels in white and black this is equal to 784 neurons as only we have one channel. In case that we manage colors scale the image will be 28x28x3 being the 3 channels, red, green and blue and the number of neurons in the input equal to 2352.

Transformation, in the CNN will be normalized the values dividing the number of pixels by 255 which is the total number of pixels.

Convnet, CCN, is formed by several layers as described in the next sections

2.1.- CONVOLUTION LAYER

Convolutional networks are built base on convolutions which aim to extract features from the images down evaluation.

Every image is converted to a matrix of pixeles which will be managed algebraically going from the pixels level to the numerical matrix level. An image can be extracted from different sources like photo, video cameras and could be captured statically or in movement like the video cameras.

Images have 3 channels, red, green and blue, that can be separated in parallel numerical matrices having range of pixels from 0 to 25.

As previously commented the convolutional network transforms images from pixels to numerical matrices which represent them and extract features that will allow the algorithms to perform images classification, clusterization, modification, tuning etc.

To the matrix created are applied filters multiplying them and through which are extracted and detected in a sole matrix features to process in the models. Filters are different depending of the features to extract. This is denominated a convolution and the number of filters are treated as parameters learned during the back-propagation.

For example in the example below for a two dimension picture (1 channel) is applied a filter to identify vertical features

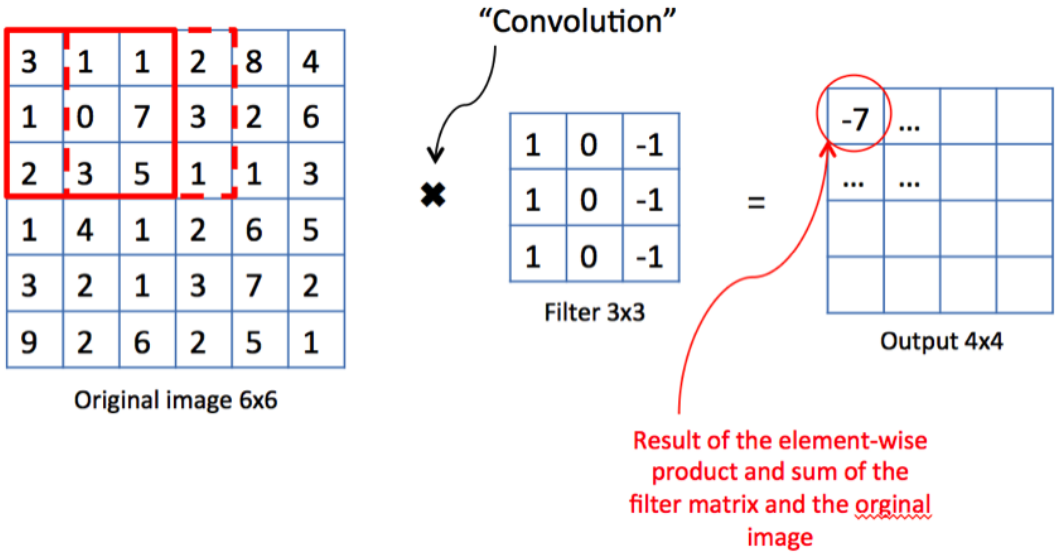


Figure 1. CONVNET Stride

In the filter shown in the figure 1 are used:

- 0 extract greys
- 1 filter brightness
- -1 extract areas in the image of darkness

As well in the next sections is explained other actions applied to the matrix that tune the outcome of the convolution like:

- Stride: Horizontal and vertical movement within the matrix. This drives the multiplication of the matrix with the filters. In the Figure 1 has been applied a stride of 1, denominated “valid convolution” as all the movements stays within the boundaries of the original matrix

- Padding: Adding rows and columns of zeros around the matrix to evaluate the areas of the picture located in the boundaries which may have useful information that otherwise wouldn’t be well evaluated

2.2- Padding

As explained one of the challenge in an image evaluation are the edges evaluation and how to not lose the information that they provide. When it is used a stride of one, as in the figure 1, the pixels that are in the boundaries of the image are less processed by the filters losing valuable insights in some cases.

To solve this, it is applied padding which is basically the addition of rows and columns of Zeros around the matrix as previously anticipated being the result shown in the picture 2.

0	0	0	0	0	0	0	0
0	3	1	1	2	8	4	0
0	1	0	7	3	2	6	0
0	2	3	5	1	1	3	0
0	1	4	1	2	6	5	0
0	3	2	1	3	7	2	0
0	9	2	6	2	5	1	0
0	0	0	0	0	0	0	0

Figure 2. CONVNET Padding

2.3.-Pooling

Pooling is a hiperparameter that outcome the values of the filter application. There are two types:

- Average Pooling: In the application of the filter in the image vectored, it is taken the average value as outcome of the operation
- Max Pooling: In the application of the filter with the image vectored, it is taken the maximum value as shown in Figure 3 as outcome of the operation

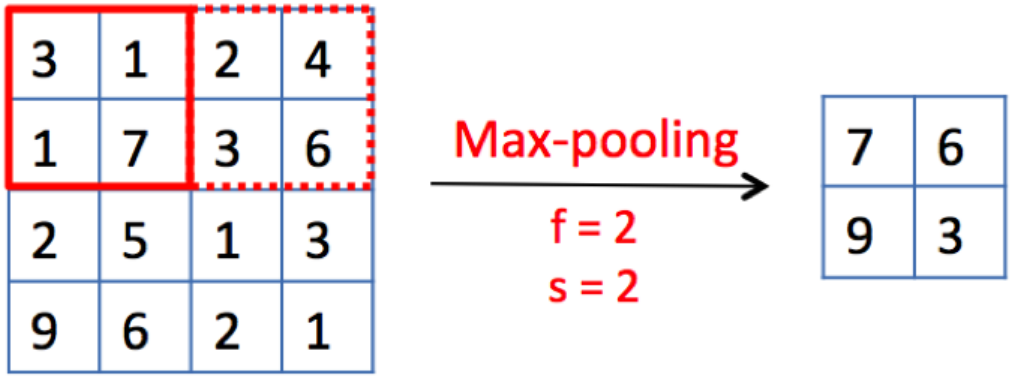


Figure 3. CONVNET Pooling

2.4.- Stride

Stride as anticipated in previous section is the number of pixels stepped in each operation between the vectored image with the filter applied.

Output size of the operation is calculated through the formula below being the variables

- filter size (f),
- stride (s),
- pad (p),
- input size (n): As image are composed by three channels, red, green and blue this is composed by three matrices once per channel as shown in figure 4 where is shown the three channel in the vectored image with a filter of 27 parameters divided in 3 matrices of 3x3x3 once per channel

$$Output\ size = \left(\frac{n + 2p - f}{s} + 1\right) x \left(\frac{n + 2p - f}{s} + 1\right)$$

Formula 1: CONVNET Output Size

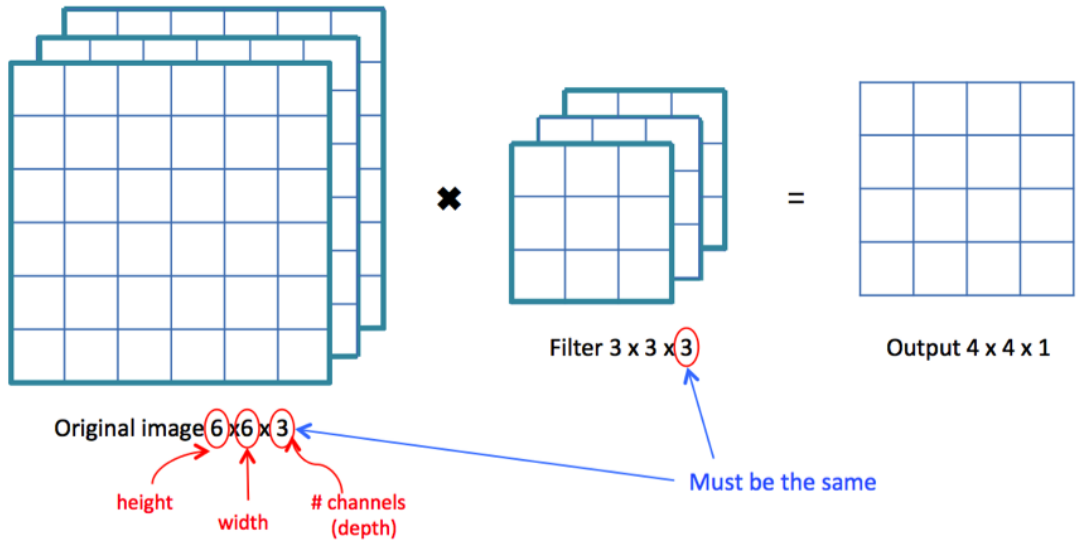


Figure 4. CONVNET 1 Filter

The number of filters can be extended as shown in figure 5 where are applied two filters where the result obtained in each are combined in the convolutional model.

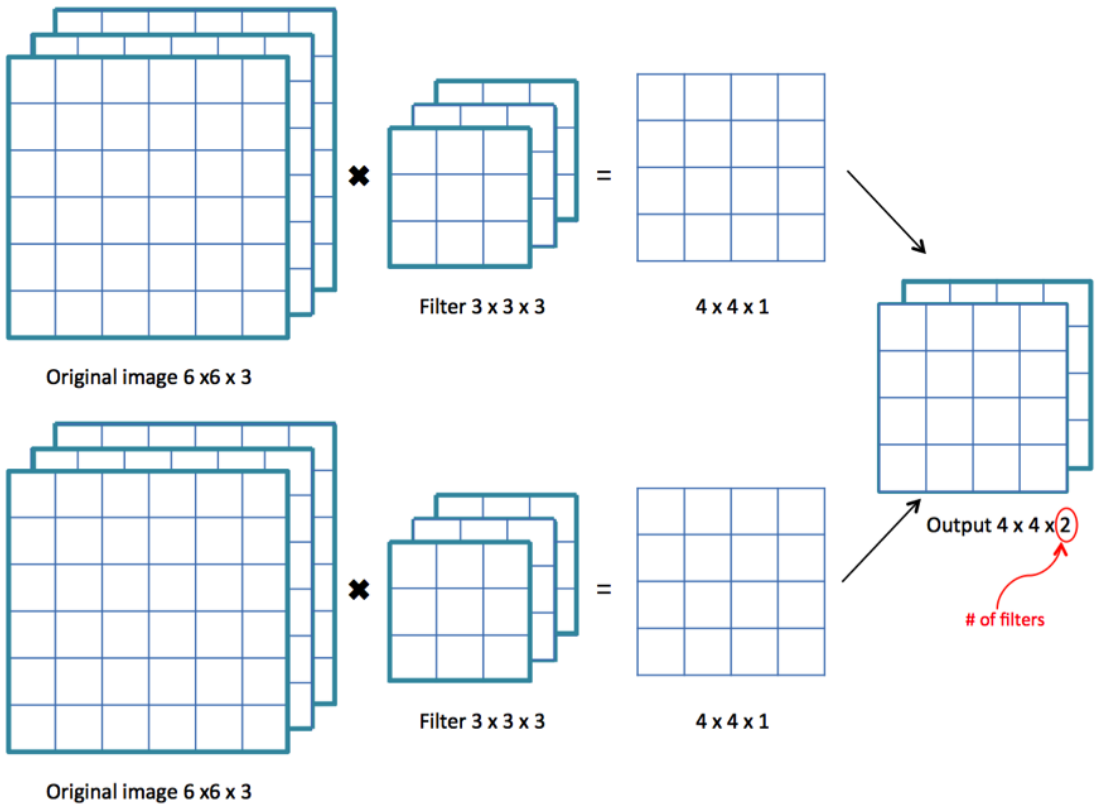


Figure 5. CONVNET 2 Filters

So considering more than one filter the matrix dimension output will follow the formula 2

Input:	Filter:	Output:
$(n\ x\ n\ x\ n_c)$	$(f\ x\ f\ x\ n_c)$	$\left(\left[\frac{n+2p-f}{s}+1\right]\ x\ \left[\frac{n+2p-f}{s}+1\right]\ x\ n_c'\right)$

Formula 2: CONVNET Output Size

- nc' as the number of filters

2.5.- Convolutional neural network.

2.5.1.- Layer

As shown in figure 6 it is added the bias and the non linear functions which will drive the convergence of the neural network, selecting a ReLU function in the layers.

The way to calculate the number of parameter per layer depend of the input matrix so following the example as explained before considering 5 filters of a size 3x3x3 pus 1 bias gives 140 parameters.(5x28)

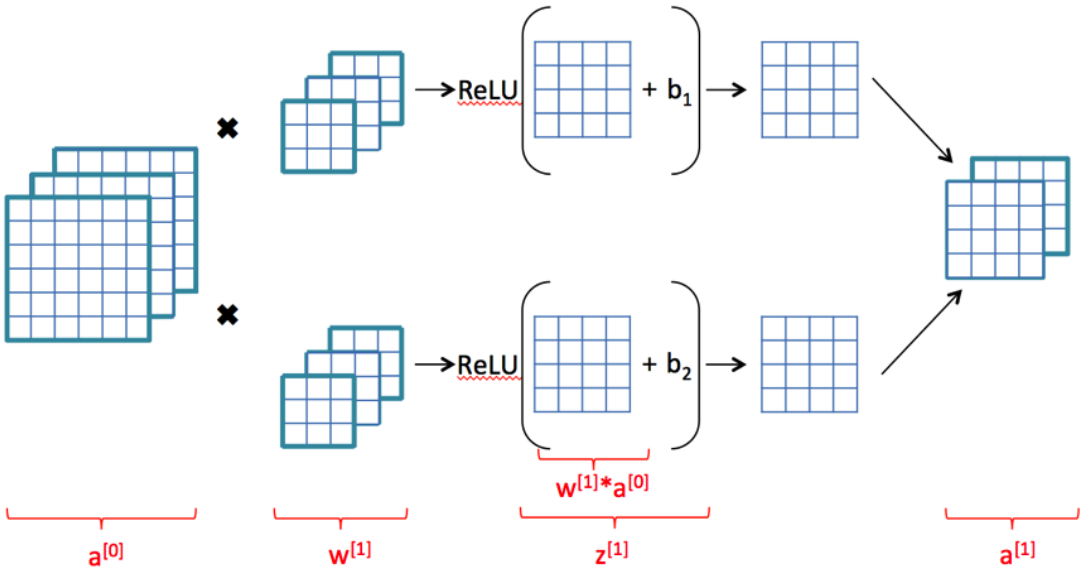


Figure 6. CONVNET One CNN Layer

2.5.2.- Complete Deep Convolutional Network

In figure 7 is consolidated all the info detailed in the previous sections with a complete deep convolutional neural network, including pooling within each layer flattening the results obtained and closing it with a Softmax.

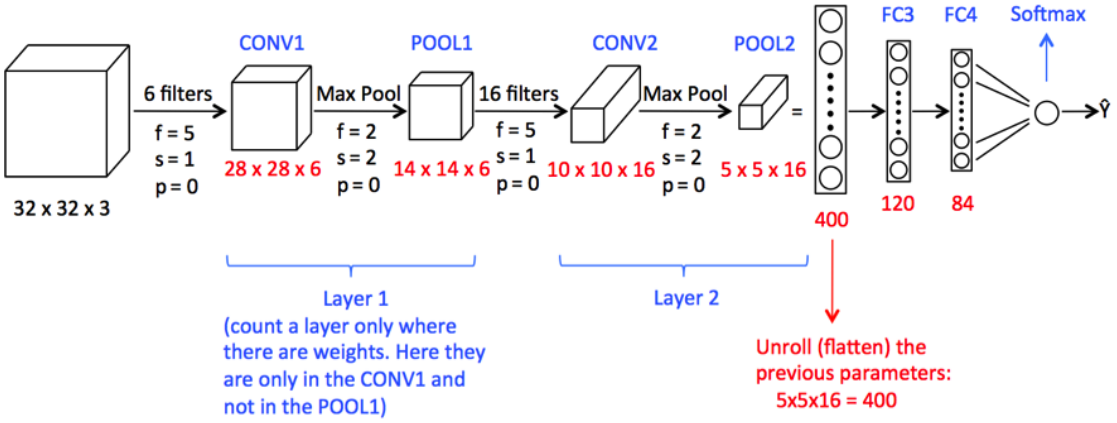


Figure 7. CONVNET Complete Deep CNN

The architecture shown in figure 7 is based by LeNet-5 as one of the classical architectures developed in convolutional networks which include Weights and Bias acting as a neural network.

As final key highlight to soften the computational requirement in images management, this convolutional networks can use as weights, the weights trained in previous models which is denominated transfer learning, saving time and computational requirement during the training.

3.- Face Detection With Deep Learning

3.1.- Multi-Task Cascaded Convolutional Neural Network

For our project we have selected the "state of art" algorithm in face detection "**Multi-Task Cascaded Convolutional Neural Network", MTCNN,** described by Kaipeng Zhang, et al. in the 2016 paper titled "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks." [4]

The network uses a cascade structure with three networks as shown in the figure 8 taken from the paper referenced [4] and as summarized in the next 3 bullet. In the next section this will be deeper developed:

- First step: It's build the image pyramid which rescale the image in different sizes being the first model the Proposal Network or P-Net which proposes candidate facial regions
- Second step: Denominated Refine Network or R-Net which filters the bounding boxes individualizing the one to use in the next step
- Third step: Denominated Output Network or O-Net which proposes facial landmarks, 5 points in the face that detect exactly the face position

One of the challenges of all the models developed previous to the multitask Cascaded Convolutional Network is the challenge of integrating the face detection with face alignment.

As stated in the paper referenced [4]:

" In this paper, we propose a new framework to integrate these two tasks using unified cascaded CNNs by multi-task learning. The proposed CNNs consist of three stages. In the first stage, it produces candidate windows quickly through a shallow CNN. Then, it refines the windows to reject a large number of non-faces windows through a more complex CNN. Finally, it uses a more powerful CNN to refine the result and output facial landmarks positions."

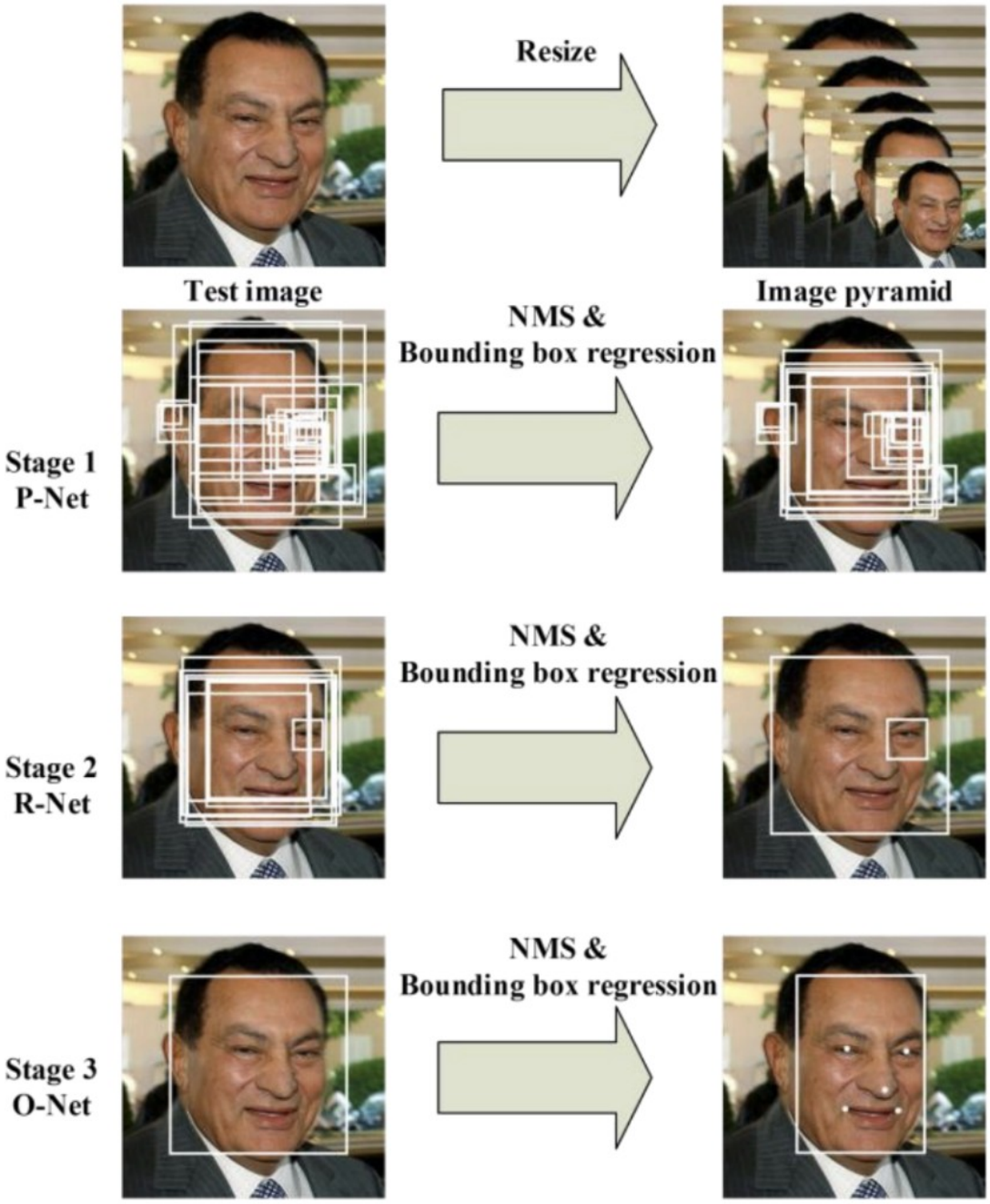


Figure 8. MTCNN Stages

3.2.- Summary of the MTCNN paper

3.2.1.- Overall framework

Stage 1: P-Net, (Proposal Network) to obtain the candidate windows and the bounding box regression vectors. After identifying them it´s used the bounding box regression vectors to calibrate the candidates.

In order to reduce the overlapping bounding boxes it is being used the NMS (Non Maximum Suppression) which remove/merge bounding boxes on the same faces

Stage 2: R-Net, (Refine Network), the output of the P-Net is the input of the R-Net where is further reduced the number of false candidates through extra calibration with bounding box regression and NMS candidate merge.

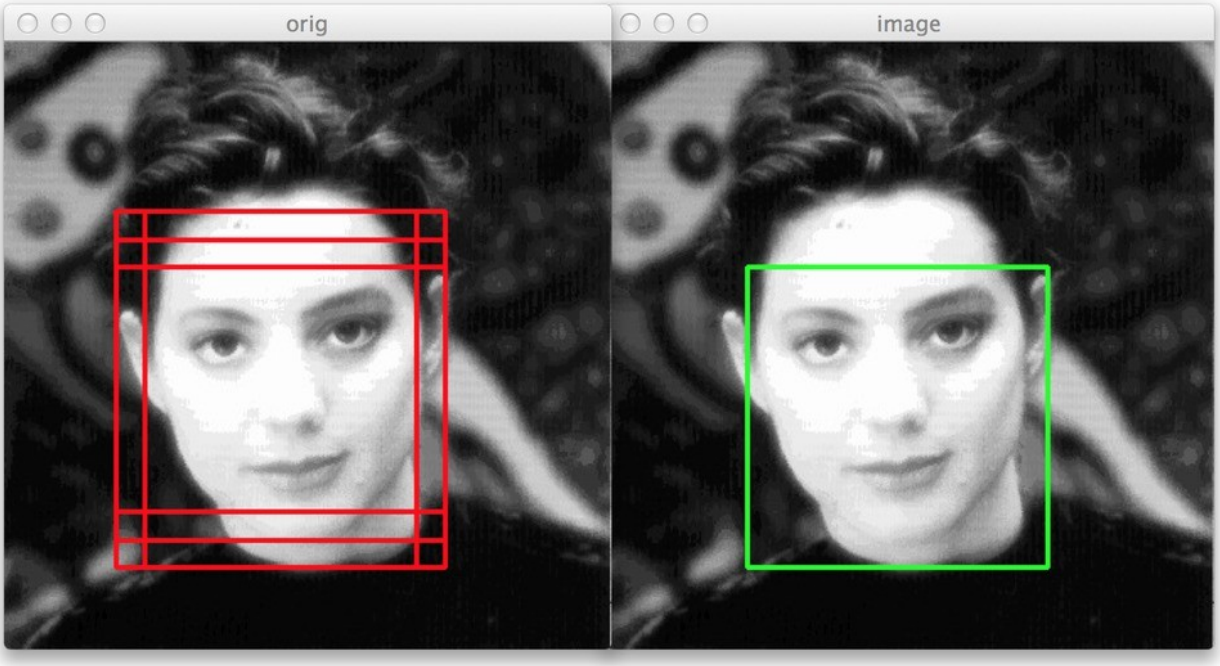


Figure 9. MTCNN R-Net 2nd Stage

Stage 3: O-Net, (Optimal Network) This last stage has the same approach of the Refine Network, describing the face in more detail and outcoming five facial landmark's positions

3.2.2.- CNN Architecture

Due to the complexity of the multiples CNN each of them will use filters of 3x3 rather than 5x5 as were the previous face detections algorithms ([19] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in IEEEf Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325-5334) the MTCNN use filters of 3x3 (Figure 11)

reducing the computing and increasing the depth obtaining a better performance as shown in the (Figure 10).

TABLE I
COMPARISON OF SPEED AND VALIDATION ACCURACY OF OUR CNNS AND PREVIOUS CNNS [19]

Group	CNN	300 Times Forward	Accuracy
Group1	12-Net [19]	0.038s	94.4%
Group1	P-Net	0.031s	94.6%
Group2	24-Net [19]	0.738s	95.1%
Group2	R-Net	0.458s	95.4%
Group3	48-Net [19]	3.577s	93.2%
Group3	O-Net	1.347s	95.4%

Figure 10.- Computing performance [4]

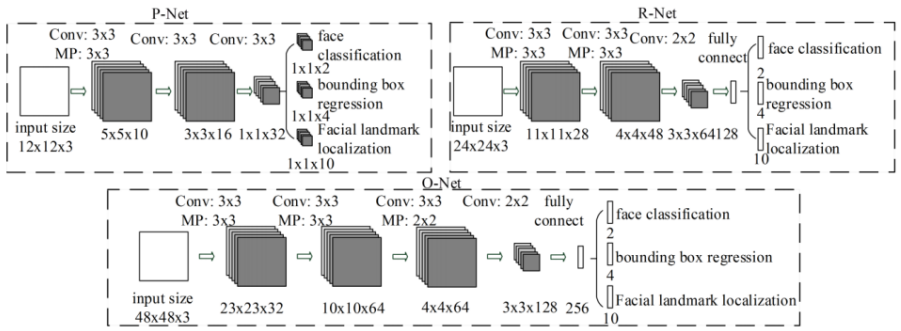


Fig. 2. The architectures of P-Net, R-Net, and O-Net, where "MP" means max pooling and "Conv" means convolution. The step size in convolution and pooling is 1 and 2, respectively.

Figure 11.- MTCNN Architecture [4]

3.2.3.- Training

Three tasks are performed during training the face identifications CNN:

1. Face classification: Its is faces as a binary classification of Face/non Face which use for each sample the cross entropy loss
2. Bounding Box regression: For each candidate window is being predicted the offset between it and the nearest ground creating an Euclidean space in which is used for the regression the Euclidean loss
3. Facial Landmark location: Similar to the Bounding Box Regression, it is a regression problem which use the Eulidean loss. In this step are identified 5 facial landmarks
    - Left eye
    - Right eye

- Nose
- Left mouth corner
- Right mouth corner

3.2.4.- Project implementations

In the project is being used the implementation provided by Iván Centeno in the ipazc/mtcnn project installing the version 0.1.0

The output of the face detector MTCNN is the input of the Facenet where is created the face embeddings

4.- Facenet

FaceNet is a deep convolutional network designed by Google, Florian Schroff, in 2015 paper tittled, *"Facenet: A Unified Embedding for Face Recognition and Clustering"* [5] trained to solve face verification, recognition and clustering problem with efficiently at scale.

1. Directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity.
2. Optimize the embedding face recognition performance using only 128-bytes per face.
3. Achieves accuracy of 99.63% on Labeled Faces in the Wild (LFW) dataset, and 95.12% on YouTube Faces DB.

The Facenet can be used thanks to 3rd parties open sources implementations of the model and the availability of pre-trained models. It extracts from a given picture high quality features from faces, called faces embeddings which can be used to train a face identification system.

Basically the embeddings are the vectorization of the images and with that we are able to run classification algorithms in which identify faces in images.

Facenet model is a deep convolutional neural network trained via triplet loss function which identifies within the Euclidean space the vectors more similar which are those with smaller distance and the faces are more differences that are the ones with vectors in the Euclidean space with larger distances between them.

The focus of Facenet is to optimize the embeddings and from them to use classifiers through which identify the faces

4.1.- Pre-Trained model

There are several projects that provide tools to pre-train the model and package through which train the Facenet model:

1. Openface: Provide Facenet models built and trained using PyTorch deep learning framework. The challenge with this model is only available in Python version 2

2. Another project is “Facenet by David Sandberg” which use Tensorflow to train and build Facenet models. Constraint with this project is the lack of a structure library or workable API, though there are a number of projects which convert David’s model to Keras

3. Finally the project used is “Facenet by Hiroki Taniai”. His project convert the inception ResNet v1 model from TensorFlow to Keras also providing a pre-trained Keras model ready to use

In this project is being used the option 3 Facenet by Hiroki Taniai project that has been trained on MS-Celeb-1M which expect colored images, pixels values whitened and square shape of 160x160 as input and return a face embedding 128 element vector.

4.2.- Method

As explained and shown in the paper [5], Facenet uses a deep convolutional network that train the CNN using Stochastic Gradient Descent (SGD) with standard backdrop and AdaGrad. The architecture followed is shown in the figure below:

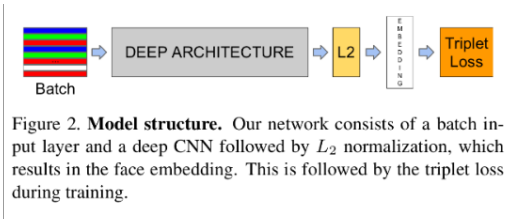


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

- The blackbox Deep Architecture shown the CNN trained
- L2 is a normalization applied which provides the embeddings
- Concluding with the minimization of the Loss function, Triplet Loss

Figure 12: Facenet Architecture

Below the high level representation of the Triplet Loss architecture:

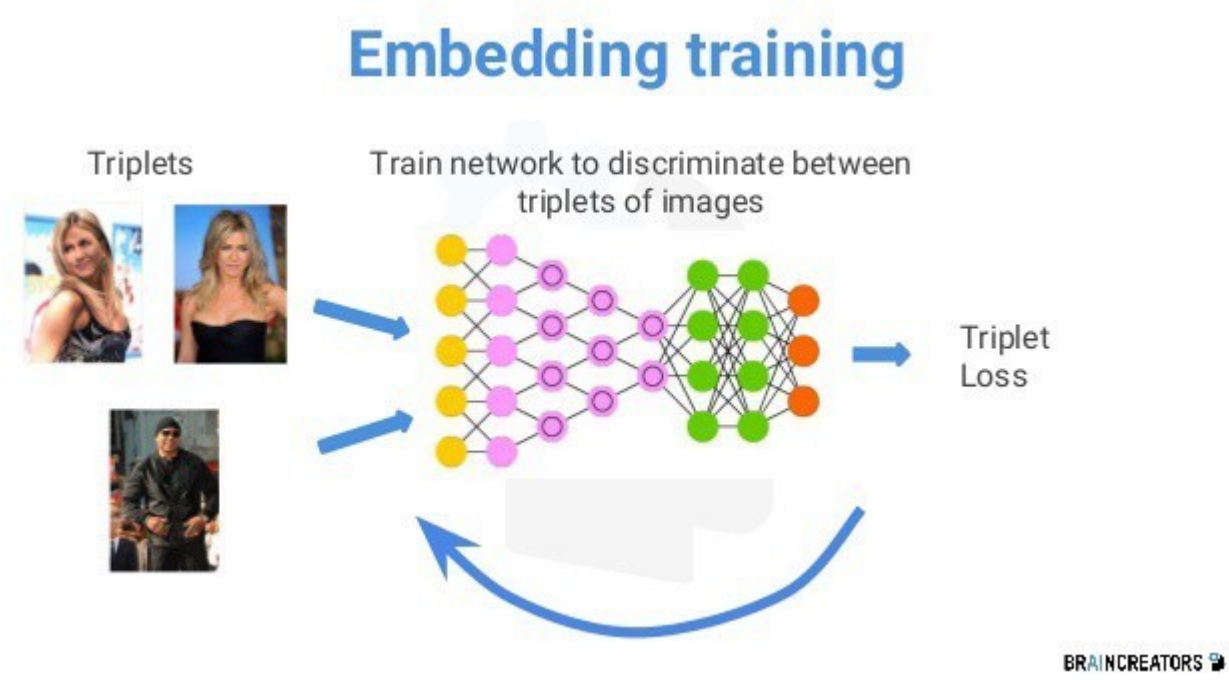


Figure 13: Facenet Triplet Loss architecture

4.3.- Triple Loss

It considers 3 images categorized as below taking the embeddings and calculating the distances and minimizing the distance between the anchor and the positive image and maximizing the distance between the anchor and the negative

1. Anchor being the image to check
2. Positive being the match
3. Negative different to the match

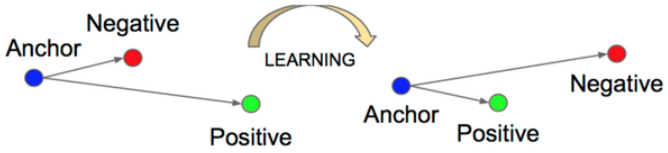


Figure 14: Facenet Triplet Loss

The embedding are represented in the Euclidean space and is there where will be identified the different triplets used to train the model through the loss function below:

Euclidean space generated

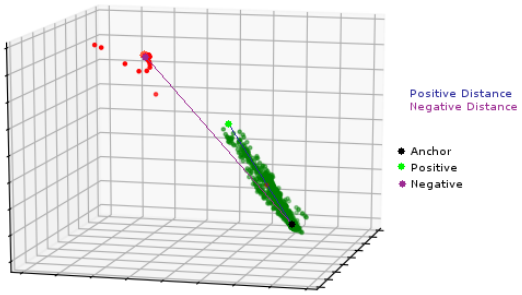


Figure 15: Facenet Loss Function

Loss Function

$$Loss = \sum_{i=1}^{N} \left[\| f_i^a - f_i^p \|_2^2 - \| f_i^a - f_i^n \|_2^2 + \alpha \right]_+$$

Schroff et al.

Formula 3: Facenet Loss Function

Alfa is the margin, a Hyper-parameter that generates a minimum space between the positive and negatives.

4.3.1.- Triplet Loss Selection

In the triplets generated we can differentiate in:

1. Easy satisfied Triplet. These triplets are the ones with longer separation between the Anchor and the positive and the anchor and the negatives:

   d(A,P) + Alfa ≤ d(A,N)

   || f (A)- f(P) || ^ 2 + Alfa ≤ || f (A)- f(N) || ^ 2

   Formula 4: Easy satisfied Triplet

2. Hard satisfied triplet. This are the triplets where the differences between anchor and positive and anchor and negatives are minimals and therefore are the ones which we should train as they are the most difficult to categorize

   $d(A,P) + Alfa \approx d(A,N)$

   $|| f(A) - f(P) ||^2 + Alfa \approx || f(A) - f(N) ||^2$

   Formula 5: Hard satisfied Triplet

In order to improve the convergence and the computational charge of the Gradient descent Procedure (GDP) in the back propagation the model has to select Hard Satisfied Triplet which breach the Formula 5 as it can work in a range of values that can be improved quicker and more efficiently the GDP that if we use Easy Satisfied triplet identified randomly.

In the example below is calculated the Loss function in two different scenarios:

Example A:

Alfa = 0.4

Distance between Anchor and positive = 1.2

Distance between Anchor and negative = 2.4

Loss Function = -1.6

Example B:

Alfa = 0.4

Distance between Anchor and positive = 0.2

Distance between Anchor and negative = 2.4

Loss Function = -2.6

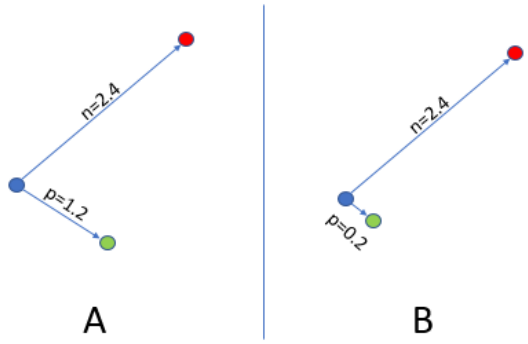


Figure 16: Triple Loss Example

$$n = \| f_i^a - f_i^n \|_2^2$$
$$p = \| f_i^a - f_i^p \|_2^2$$
$$\alpha = 0.4$$
$$A = 1.2 - 2.4 - 0.4 = -1.6$$
$$B = 0.2 - 2.4 - 0.4 = -2.6$$

Formula 6: Triple Loss Example

As can be observed the Triplet loss has to be maximized as :

loss = K.maximum (basic_loss,0.0) being in case A and B the result equal to 0 losing a wide range of information as in both cases the function give the same result, 0, but it is clear that the case A is better than the case B. This is an area of further improvement of the Triplet Loss Function, computational and from a performance prospective.

5.- Face Classification

5.1.- Introduction

As last stage of the process, it has to be performed the classification of the face embeddings of the Celebrity Faces Dataset used.

Doing this we have selected several Machine Learning algorithms classification models configured using K-fold Cross validation

The Super Learner is an ensemble Machine learning algorithm which consolidate the k-fold outcome (out-of-fold) of each of them as the input of the super leaner ensemble algorithm.

The Super Learner is a Stacking ensemble algorithm which aim to improve the accuracy of the machine learning algorithms competing with the performance of the deep neural network. DNN performs better that individuals ML models and the Superlearner ensemble results demonstrate that it is competitive in term of:

- Accuracy
- Loss
- Transparency and understandability
- Interpretability of results
- Hyper-parameter management. Reduced number of hyperparameters
- Convergence
- Computational requirement

5.2.- Ensemble methods

It combines several machine learning models, training them and generating a single learner which outcome higher accuracy than a single learner.

Stacking models involve more than one machine learning base models which will be named Level 0 (Base model) models being the out-of -fold results the input of the Level 1 (Meta model) which use all the predictions of the base models to train itself.

The base models will be executed via k-fold cross validation using the same splits of the data and this dataset trains the meta model in isolation as a separate model. As an option and to provide more context the meta model can be added to the meta model training dataset the data of the base models.

As stated in the paper from Steve Young, Tamer Abdou and Ayse Bener "Deep Super Learner [10]: A Deep Ensemble for Classification Problems" dated on 6th March.

"Super learning is an ensemble method proposed by Van der Laan et al. that optimizes the weights of the base component learners by minimizing a loss function given cross-validated output of the learners. Super learning finds the optimal set of weights for the learners and guarantees that performance will be at least as good as the best base learner [11]. The proposed algorithm, DSL, is an extension of the super learner ensemble."

One of the key aspects of the performance and the capacity of an ensemble is the diversity among the base models. The super learner adapts to several problems following the base models diversity and how those optimize the weights on each model for the same problem.

5.2.- Super Learner architecture

The super learner ensemble consists in a layer by layer process which works in a cascade approach taking the output of one layer as the input of the following layer.

The architecture shown in the figure 17, extracted from the paper [10], uses the next hyper-parameter structure:

- j classes
- l features
- m base learners
- n records in the training setbacks

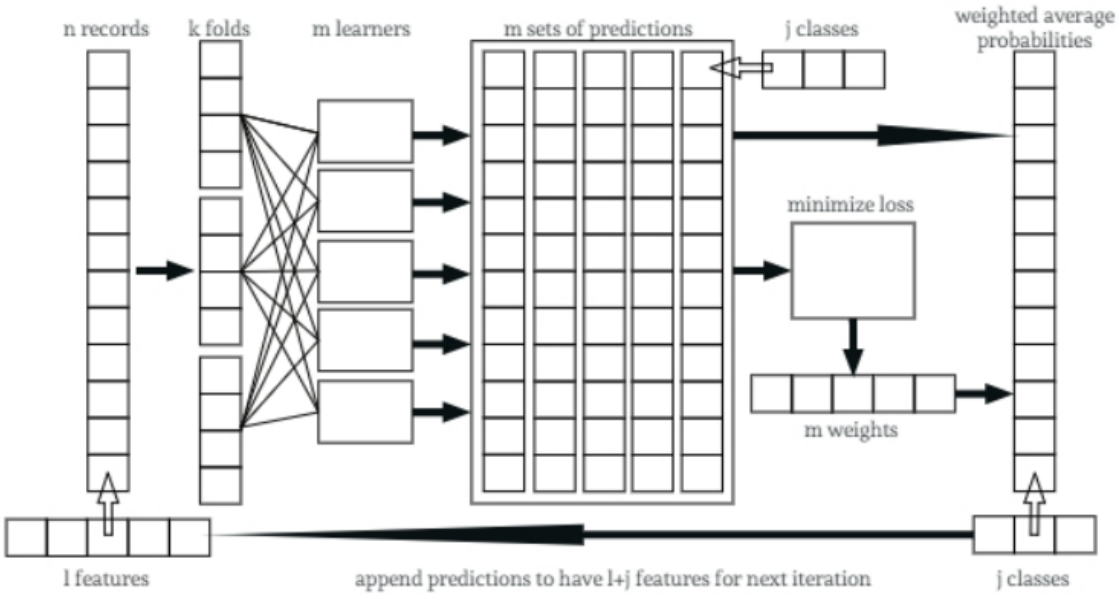


Fig. 1. Overall procedure of DSL with j classes, k folds, l features, m learners, n records

Figure 17: Super Learner Algorithm Architecture

The phases in the DSL are:

1. Cross Validation:

   Dataset is randomly split into training and test sets where a maximum percentage of the data is taken into the training set. Though the test dataset is small, there is still some chance that we left some important data in there that might have improved the model. And there is a problem of high variance in the training set. To solve this problem it is used the K-fold cross-validation.

Cross-validation is a technique that is used to evaluate machine learning models by resampling the training data for improving performance.

In K-fold Cross-Validation, the training set is randomly split into K(usually between 5 to 10) subsets known as folds. Where K-1 folds are used to train the model and the other fold is used to test the model. This technique improves the high variance problem in a dataset as we are randomly selecting the training and test folds

2. Build and train each base model on each k-fold
3. Optimize the loss function obtaining the optimized weights
4. Taking the optimized weights calculate the weighted average of the predictions across learners to obtain overall predictions for each record
5. As optional step is required a re-train of all the base models (Required by Van der Laan) though with a correct number of k-folds the model performs well and it may not need the retrained steps unless that the number of k-folds is reduced or in those models like KNN were the testing is more expensive computational than the training
6. Append the predictions obtained in the step 4 to the original dataset. Predictions generated are concatenated as additional features with the original ones generating l+j features
7. As shown in the architecture the concatenated training dataset feeds the model reducing the loss on each iteration till get the optimum value

6.- Implementation

As industrial implementation this project aims to enhance the capabilities of the surveillance and access control of sensible infrastructures such as Nuclear, Power, Water plants, Refineries and Petrochemical installations

This facilities usually has installed a surveillance software with a net of CCTV which may enjoy this new enhancement upgrading just the operating Software and installing additional video cameras in the entries of the facilities

As example of implementation we will consider the Software of surveillance provided by Honeywell which include analytics, Intrusiontrace and Loitertrace:

- Intrusiontrace: provides high performance intrusion detection using video analytics specifically designed for 24/365 outdoor operation.
- Loitertrace: provides high performance loiter detection using video analytics specifically designed for indoor/outdoor operation.

As first implementation step it has to be presented to the Honeywell TRACE product manager explaining the features and capabilities of the algorithm developed and evaluate the investment in upgrading the product estimating a low effort considering the framework developed in Python which is composed for three algorithm tested largely.

Return of the investment for Honeywell: Commercially the orders and revenue for the Video systems in Industrial facilities may increase in 25% considering that this is a differentiator in the market that others vendors are not still offering.

Return of the investment for Honeywell customers: With the installation of additional cameras in the entry of the facility can be removed the input lathes avoiding bottlenecks of people in the entrances which will improve the productivity and more considering the current measures taken due to the Covid19 which requires a minimum social distances being in the morning slot when all the staff get in the plant and in the afternoon when the staff concludes the shift the timing of biggest risk of contamination which will become all the accesses slower and difficult to manage. With the installation of an access

system with facial recognition the input lathes can be removed and get more space and quicker access that facilitate the flow of the people, accurate recognition and safer process.

Cameras to use:

The cameras compatibles with the Software are:

COMPATIBLE HONEYWELL CAMERAS	
HBL6GR2	6 MP Rugged IR Network Bullet Camera
HBD8GR1	4K Rugged IR Network Bullet Camera
H4L6GR2	6 MP Rugged IR Network Minidome Camera
H4D8GR1	4K Rugged IR Network Minidome Camera
HCD8G	4K Network Box Camera
HFD6GR1	6 MP Rugged IR Network Fisheye Camera
HFD8GR1	12 MP Rugged IR Network Fisheye Camera
HEPZ302W0	1080p Explosion-Proof Network PTZ Dome
HDZ302DE	1080p 6" Indoor/Outdoor Network PTZ Dome
HDZ302D	1080p 5" Indoor/Outdoor Network PTZ Dome
HDZ302DIN	1080p Indoor In-Ceiling Network PTZ Dome

Figure 18: CCTV. Cameras

6.1.- Budget required

As budget required it has been divided in the two applications identified, surveillance and access control:

Description	Unit Price	Uds	Total Price	Timing	Comment
Software Upgrade Intrusiontrace	10,000 $	1	10,000 $	3 months	
Software Upgrade Loitertrace	10,000 $	1	10,000 $	3 months	
Additional cameras in the facilities entrance	4,000 $	5	20,000$	1 month	
Additional surveillance cameras in the perimeter of the facility	4,000 $	20	80,000$	1 month	To leverage all the capabilities of the Face recognition App

Table 1: Budget

7 Conclusions

In order to outcomes conclusions it has been tested 3 scenarios combining real pictures and cartoon in the training datasets. Three of them has a common structure through which will be trained and tested the mentioned scenarios and see compare results base on same design conditions.

7.1 Notebook structure

The notebook has been developed in Google Colab being the datasets saved in Google Drive and following the next steps in the notebook elaboration:

1.- First we it is performed the Google Drive activation and change to the directory where are saved the datasets

2.- Multilayer Convolutional Network (MTCNN): Face Detection. installing the library ‘mtcnn’ and verifying the version 0.1.0. It’s tested the library with a picture where are identified the faces and after isolated in 5 faces picture verifying the good performance of the MTCNN algorithm

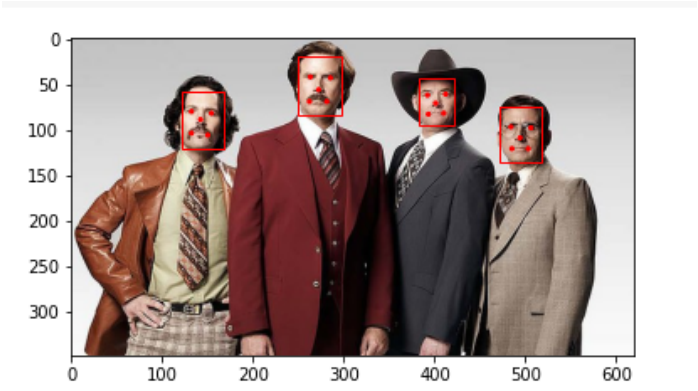


Figure 19: Faces detection

Figure 20: Faces isolation

3.- Face Embeddings: As next step it is created with the Facenet algorithm the embeddings of the images which has been uploaded and saved in a compressed file. The face embeddings are vectors which represent the features extracted from the faces

4.- Face Classification: FACENET algorithm executes the face classification creating an Euclidean space with all the vectors, classifying faces through triplet loss selection which is highly required computationally as it has to dig the entire euclidean space evaluating vectors distances.

In order to explore a less computational requirement solution, as commented in previous sections, it has been used machine learning classification models combined in stacked models and a super-learner ensemble in which we evaluate three scenarios tested.

7.1.1 First testing scenario:

This scenario use in both in the training and validation datasets Real faces of celebrities:

Train Dataset('5-celebrity-faces-dataset/train/')

Ben Afflek (14 pictures)

Elton John(17 pictures)

Madonna(19 pictures)

Jerry Seinfeld (21 pictures)

Mindy Kaling(22 pictures)

Test Dataset('5-celebrity-faces-dataset/val1picture/')

Ben Afflek (1 picture)

Elton John(1 picture)

Madonna(1 picture)

Jerry Seinfeld (1 picture)

Mindy Kaling(1 picture)

As shown in the dataset the best performance model is the Superlearner which give the highest probability of prediction with values higher than 90% improving in a 4% the result obtained in the stacked ensemble

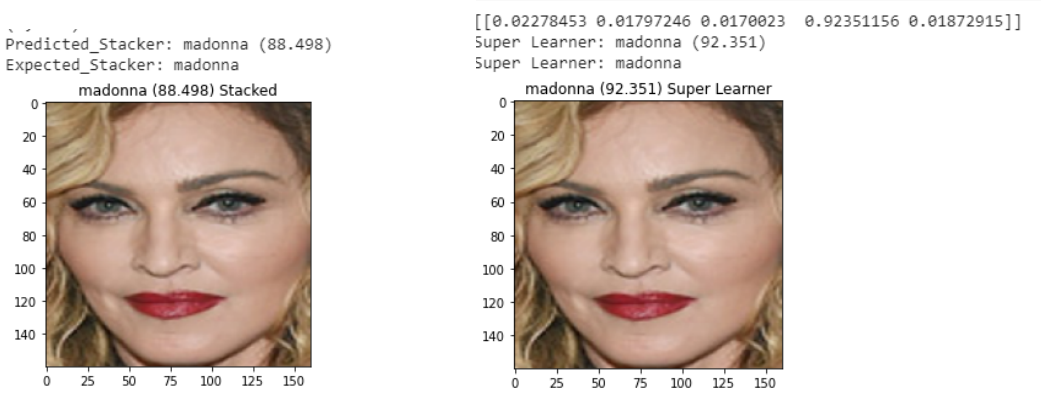


Figure 21: Scenario 1 Faces identification probabilities Madonna

Accuracy of the Super-learner model:

```
---Training---

LogisticRegression: 100.000

DecisionTreeClassifier: 100.000

SVC: 100.000 GaussianNB: 100.000

KNeighborsClassifier: 100.000

AdaBoostClassifier: 66.667

BaggingClassifier: 100.000

RandomForestClassifier: 100.000
```

```
ExtraTreesClassifier: 100.000

---Validation---

LogisticRegression: 100.000

DecisionTreeClassifier: 100.000

SVC: 100.000

GaussianNB: 100.000

KNeighborsClassifier: 100.000

AdaBoostClassifier: 60.000

BaggingClassifier: 100.000

RandomForestClassifier: 100.000

ExtraTreesClassifier: 100.000
```

Super Learner: 100.000

Accuracy of the Stacked model:

```
>lr 1.000 (0.000)
>knn 1.000 (0.000)
>cart 0.924 (0.084)
>svm 1.000 (0.000)
>bayes 0.990 (0.031)
>stacking 1.000 (0.000)
```

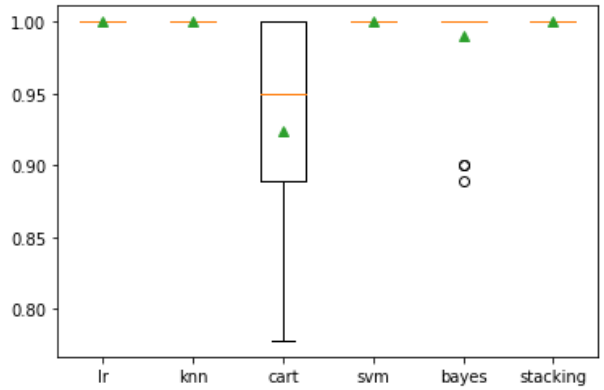


Figure 22: Scenario 1 Stacked Ensemble boxplot accuracy

7.1.2.- Second testing scenario

This scenario uses in the training dataset a reduced number of cartoon pictures of celebrities of differemt races and real faces of celebrities in the validation dataset.

Train Dataset('5-celebrity-faces-dataset/traincaricatura14/')

Beyonce (14 cartoon pictures)
Barack Obama (14 cartoon pictures)
JFK (14 cartoon pictures)
Dalai Lama (14 cartoon pictures)
JayZ (14 cartoon pictures)

Test Dataset('5-celebrity-faces-dataset/val1picturecaricatura/')

Beyonce (1 picture)
Barack Obama (1 picture)
JFK (1 picture)
Dalai Lama (1 picture)
JayZ (1 picture)

In this testing scenario has been loaded a training dataset with 14 cartoon pictures and a validation dataset with 1 real picture of 5 celebrities. The challenge tested in this scenario is the reduced number of pictures to train and the quality being cartoon pictures which influence the result obtained in the models used.

We have simulated a situation where the quality of the picture in term of face distortion and how far is the match with the real face plus the reduced number of cartoon used to train which become this scenario the most challenging one. Despite of all the constraints added to the model It still performs well which demonstrates that the architecture selected is strong in providing high accuracy of the model.

Predict probability:

Stacker: 88.6% Super-Learner: 83.8% SVC: 85.2%

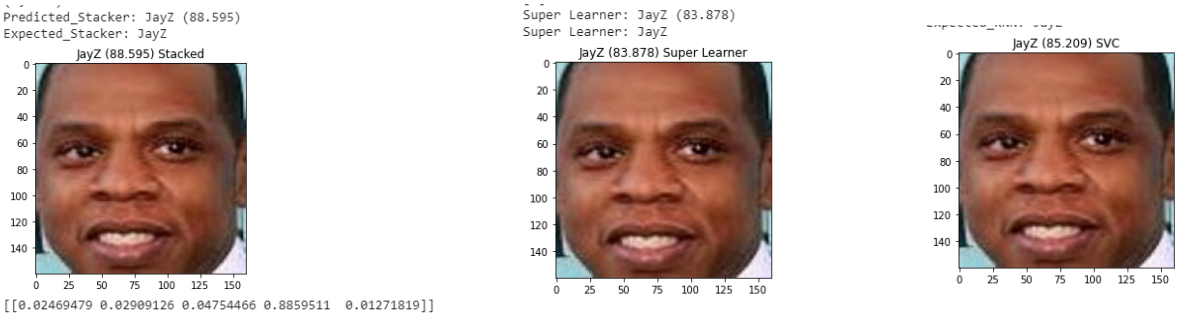


Figure 23: Scenario 2 Faces identification probabilities. JayZ

The model which provides higher probability is the stacked model vs the superlearner and SVC being the prediction probability of the superlearner the lowest.

Accuracy of the Super-learner model:

```
---Training---

LogisticRegression: 97.143

DecisionTreeClassifier: 100.000

SVC: 100.000

GaussianNB: 98.571

KNeighborsClassifier: 98.571

AdaBoostClassifier: 40.000

BaggingClassifier: 100.000

RandomForestClassifier: 98.571

ExtraTreesClassifier: 100.000
```

```
---Validation---

LogisticRegression: 100.000

DecisionTreeClassifier: 60.000

SVC: 100.000

GaussianNB: 100.000

KNeighborsClassifier: 100.000

AdaBoostClassifier: 40.000

BaggingClassifier: 100.000

RandomForestClassifier: 100.000

ExtraTreesClassifier: 100.000

Super Learner Val: 100.000
```

Accuracy of the Stacked model: The accuracy with the 14 cartoon pictures dataset is lower than the results obtained in the previous scenario which used real pictures

```
>lr 0.943 (0.079)
>knn 0.943 (0.070)
>cart 0.638 (0.168)
>svm 0.924 (0.088)
>bayes 0.924 (0.088)
>stacking 0.919 (0.102)
```

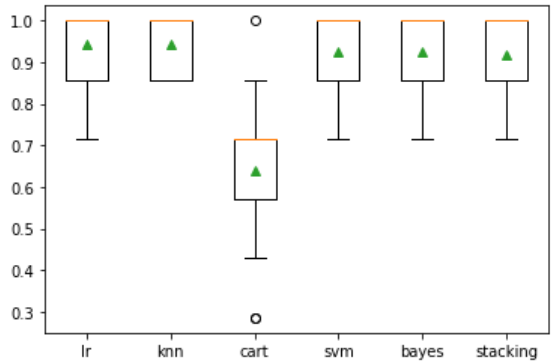


Figure 24: Scenario 2 Stacked Ensemble boxplot accuracy

Further to that we have tested how the model perform with different races showing the model equally a high probability of prediction in each case:

Dalai Lama: The best performance model is again the SVC being closer to the stacked model and again the worst the Super-learner ensemble:

Predict probability:

Stacker: 90.6% Super-Learner: 78.8% SVC: 93.5%

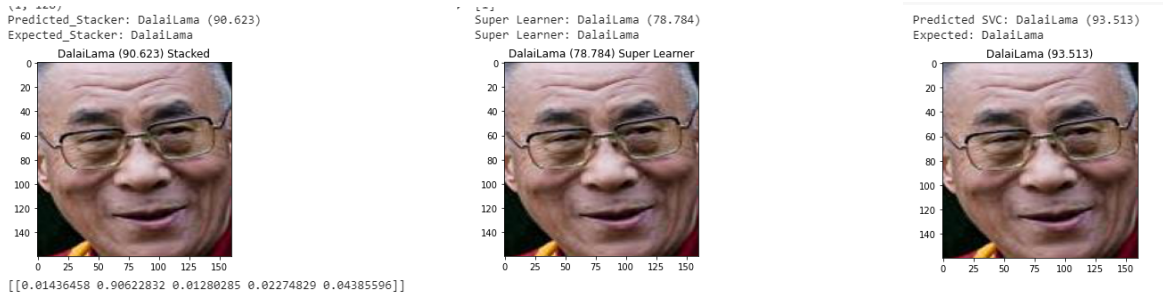


Figure 25: Scenario 2 Faces identification probabilities. DalaiLama

Beyonce: The best performance model is the Super-learner in this case having the SVC and the stacked ensemble pretty similar prediction probabilities. The factions of Beyonce which are well defined and the good quality of the 14 cartoons selected provide clearner embeddings that help the Superlearner model to perform better:

Predict probability:

Stacker: 87.7% Super-Learner: 90.3% SVC: 88.4%

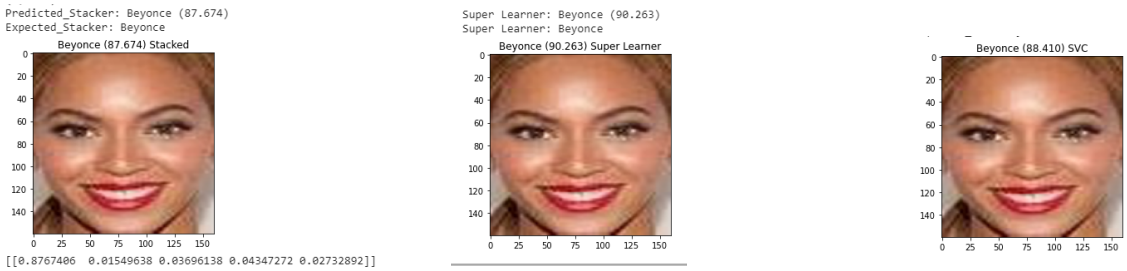


Figure 26: Scenario 2 Faces identification probabilities. Beyonce

JFK: With JFK images the highest performance is the stacked ensemble and SVC being the superlearner ensemble the less performing one which likely is due to the quality of the cartoons

Predict probability:

Stacker: 93.5% Super-Learner: 88.9% SVC: 93.3%

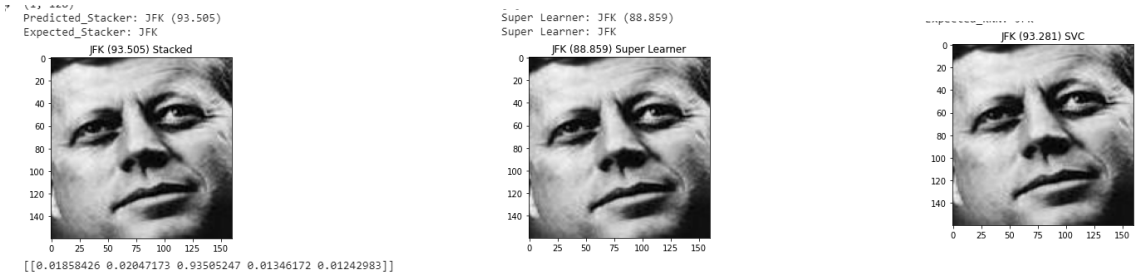


Figure 26: Scenario 2 Faces identification probabilities. JFK

It is concluded that for dataset with reduced number of samples and low quality of cartoons the usage of ensemble models like the Surperlearner and the Stacked ensemble is not showing considerable advantages in front of a model like the Linear Support Vectorial Classifier which performs well.

Further it has been tested the model modifying the hyper-parameters like the number of splits improving the prediction probability in a window between 2-4% in the ensemble models.

7.1.3.- **Third testing scenario**

In this scenario are trained a longer number of cartoons images and still one real image in the validation dataset.

Train Dataset('5-celebrity-faces-dataset/traincaricatura14/')

Beyonce (42 cartoon pictures)

Barack Obama (40 cartoon pictures)

JFK (34 cartoon pictures)

Dalai Lama (26 cartoon pictures)

JayZ (35 cartoon pictures)

Test Dataset('5-celebrity-faces-dataset/val1picturecaricatura/')

Beyonce (1 picture)
Barack Obama (1 picture)
JFK (1 picture)
Dalai Lama (1 picture)
JayZ (1 picture)

In this testing scenario has been loaded a training dataset with multiple cartoon pictures and a validation dataset with one real picture of five celebrities. The challenge tested in this scenario is using for training a wider number of cartoon pictures and how the quality of those and how representative they are of the faces tested influence the results in the models tested.

We have simulated a situation with cartoon faces where they are distorted but with a larger number of cartoon used to train which become this scenario a better performing one. Despite of all the constraints added to the model It still performs really well which demonstrates that the architecture selected is strong in providing high accuracy of the model

Predict probability:

Stacker: 87.7% Super-Learner: 90.3% SVC: 88.4%

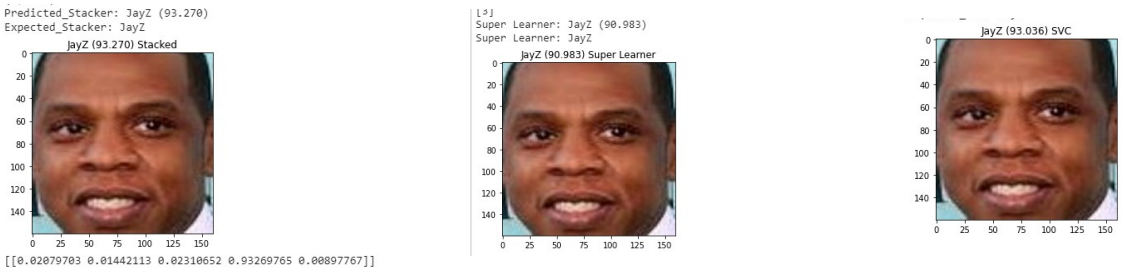


Figure 27: Scenario 3 Faces identification probabilities. JayZ

Accuracy Super-Learner Model:

```
---Training---

LogisticRegression: 96.610

DecisionTreeClassifier: 100.000

SVC: 99.435

GaussianNB: 96.045

KNeighborsClassifier: 97.175

AdaBoostClassifier: 74.011

BaggingClassifier: 100.000

RandomForestClassifier: 100.000

ExtraTreesClassifier: 100.000

---Validation---

LogisticRegression: 100.000

DecisionTreeClassifier: 100.000

SVC: 100.000

GaussianNB: 100.000

KNeighborsClassifier: 100.000

AdaBoostClassifier: 80.000

BaggingClassifier: 100.000

RandomForestClassifier: 100.000

ExtraTreesClassifier: 100.000

Super Learner: 100.000
```

Accuracy of the Stacked model: The accuracy of the stacked ensemble in this last scenario is pretty similar in all the models and comparing with previous scenarios is higher in 2 points.

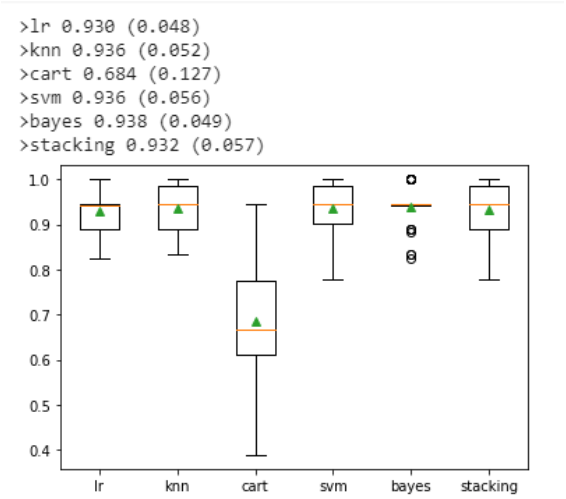


Figure 28: Scenario 3 Stacked Ensemble boxplot accuracy

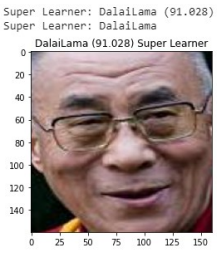
As evaluated in the previous scenario we have tested how the model performs with different races showing the model equally a high probability of prediction in each case:

Dalai Lama: The best performance model is again the SVC being closer to the stacked model and again the worst the Super-learner ensemble:

Predict probability: Three models tested perform pretty similar being the less performing again the Super-learner ensamble. The stacked ensemble performs equal to the SVC model.

Stacker: 93.8%

Super-Learner: 91.0%



SVC: 94.0%

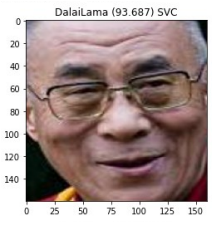


Figure 29: Scenario 3 Faces identification probabilities. DalaiLama

Beyonce:

Predict probability: Three models perform pretty similar being the less performing again the Super-learner ensamble. The stacked ensemble performs equal to the SVC model. Likely with the addition of further cartoons the quality of the training package is lower than in the previous scenario where the 14 cartoons selected had better quality.

Stacker: 98.0% Super-Learner: 94.0% SVC: 97.9%

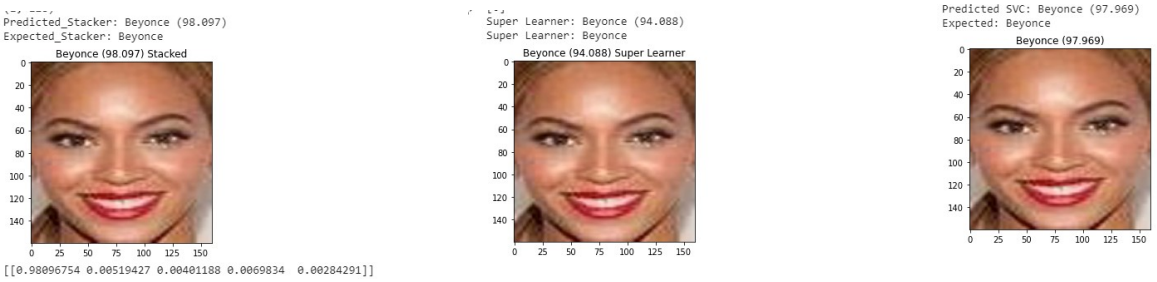


Figure 30: Scenario 3 Faces identification probabilities. Beyonce

JFK: With JFK pictures the three models perform pretty similar being the less performing again the Super-learner ensamble. The stacked ensemble performs equal to the SVC model reaching percentages close to 99%. Same as in the previous case with Beyonce likely the package of cartoons have a low quality and the superlearner even with a great performance is not able to reach the levels of the stacker and SCV ML models

Predict probability:

Stacker: 98.5% Super-Learner: 94.6% SVC: 98.7%

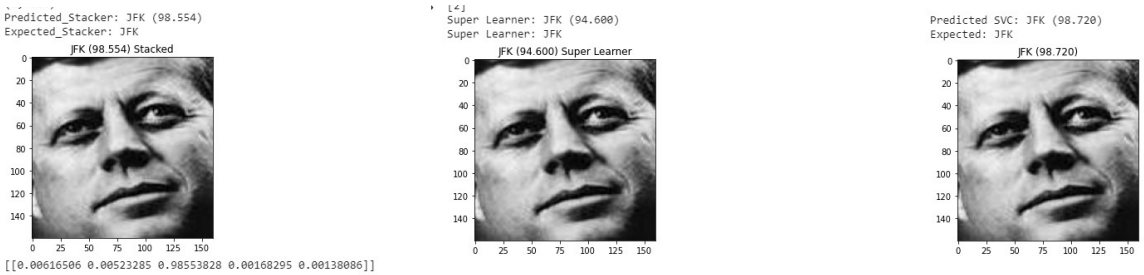


Figure 31: Scenario 3 Faces identification probabilities. JFK

7.2.- Final conclusions

As final conclusion for models trained with real picture the best performing model is the Super-learner which take advantage of all the models used in the layer 1 and the retro-feed of the training dataset with the meta-model predictions which make the model depend of the quality of the training dataset.

This might be the root cause because the superlearner model is the best performing with real pictures in the scenario 1 and for scenarios 2 and 3 that use cartoon datasets for training the best performing models are SVC and the Stacker model created.

We have seen an exception with the picture of Beyonce in the second scenario where we have in the training dataset 14 cartoons and where the Superlearner model performs slightly better than the SVC and the stacking model. This might be due to the quality of the 14 cartoons of the dataset which helped the superlearner model to perform well when the trained values are added on each iteration to the training dataset.

Same exception is seen in the scenario 3 in the case of JayZ where the training dataset is composed by 35 pictures being the best performing model the superlearner.

We can conclude that executing the classification with machine learning models is computationally more efficient than executing the classification through the triple lost research in the Euclidean space generated in the Facenet algorithm. The most efficient one when used real faces in the training dataset is the Superlearner model being a better fit either the stacked ensemble or the Linear Support Vectorial Classifier model when the training dataset has low quality pictures.

In all the cases we can get probabilities higher than 90% with 14 real pictures with a Superlearner classifier or either SVC or Stacked Ensemble if we use 14 cartoon or low quality pictures.

Bibliography

Facial Recognition

1.-https://www.gemalto.com/govt/biometrics/facial-recognition

2.-https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524

NMS (Non Maximum Suppression) Algorithm

3.-https://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/

MTCNN (Multilayer Convolutional Network Paper)

4.-https://arxiv.org/ftp/arxiv/papers/1604/1604.02878.pdf

Facenet paper

5.-https://arxiv.org/pdf/1503.03832.pdf

6.-https://zhuanlan.zhihu.com/p/35560666

Triplet lost

7.-https://omoindrot.github.io/triplet-loss

8.-https://www.coursera.org/lecture/convolutional-neural-networks/triplet-loss-HuUtN?utm_source=linkshare&siteID=je6NUbpObpQ-S1xMKLbAcUkl9pCT5IN27Q&ranEAID=je6NUbpObpQ&utm_content=10&ranMID=40328&ranSiteID=je6NUbpObpQ-S1xMKLbAcUkl9pCT5IN27Q&utm_campaign=je6NUbpObpQ&utm_medium=partners

9.-https://towardsdatascience.com/lossless-triplet-loss-7e932f990b24

Super Learner

10.-Paper from Steve Young, Tamer Abdou and Ayse Bener “Deep Super Learner: A Deep Ensemble for Classification Problems” dated on 6th March.https://arxiv.org/pdf/1803.02323.pdf

Honeywell Surveillance Softwares and CCTV

11.-https://www.security.honeywell.com/All-Categories/video systems/analytics/licensed-analytics

Formulas