Automated Face Recognition

Artificial Neural Network (ANN)

Fall 2022

Repository Link: https://github.com/mahmoudhaney/FaceRecognition

Team ID 99

	ID	NAME	LEVEL	Department
1	202000186	آية عماد طلعت حرب	3	CS
2	202000376	ساره وليد سيد	3	CS
3	202000649	فيروز جمعه منصور محمد	3	CS
4	202000677	كريمه ابو حسيبه محارب	3	CS
5	202000861	محمود هاني سعيد	3	CS
6	202000895	مصطفى جمال عبد الحكيم	3	CS
7	202001010	نورهان طارق عبدالهادي	3	CS

TABLE OF CONTENT

Introduction & Overview

Project Description

Facial recognition is a way of recognizing a human face through technology.

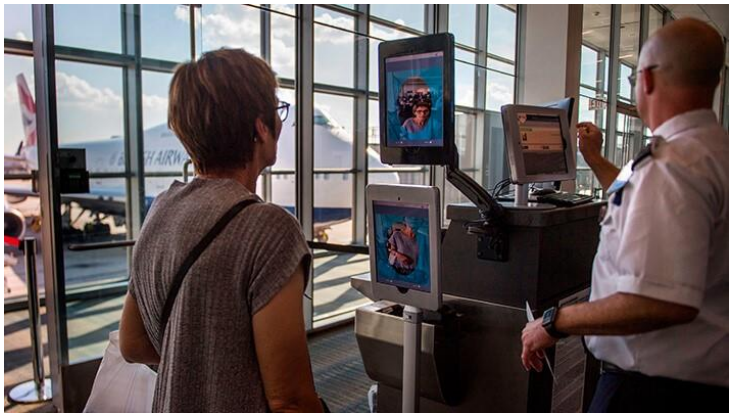
A facial recognition system uses **biometrics** to map facial features from a photograph or video. It compares the information with a database of known faces to find a match. Facial recognition can help verify a person's identity, but it also raises privacy issues.

It is a technology that can recognize or authenticate a person from a digital image or a video frame. Facial recognition systems operate in a variety of ways, but in general, they compare chosen facial traits from a given image with faces in a database.

It's also known as a Biometric Artificial Intelligence-based application that analyses patterns based on a person's facial features and form to uniquely identify them, the project solution is to recognize the face in the image and classify it between 5 labels.

Similar Applications

1- Airports and border control



Facial recognition has become a familiar sight at many airports around the world.

Increasing numbers of travelers hold biometric passports, which allow them to skip the ordinarily long lines and instead walk through an automated ePassport control to reach the gate faster.

Facial recognition not only reduces waiting times but also allows airports to improve security.

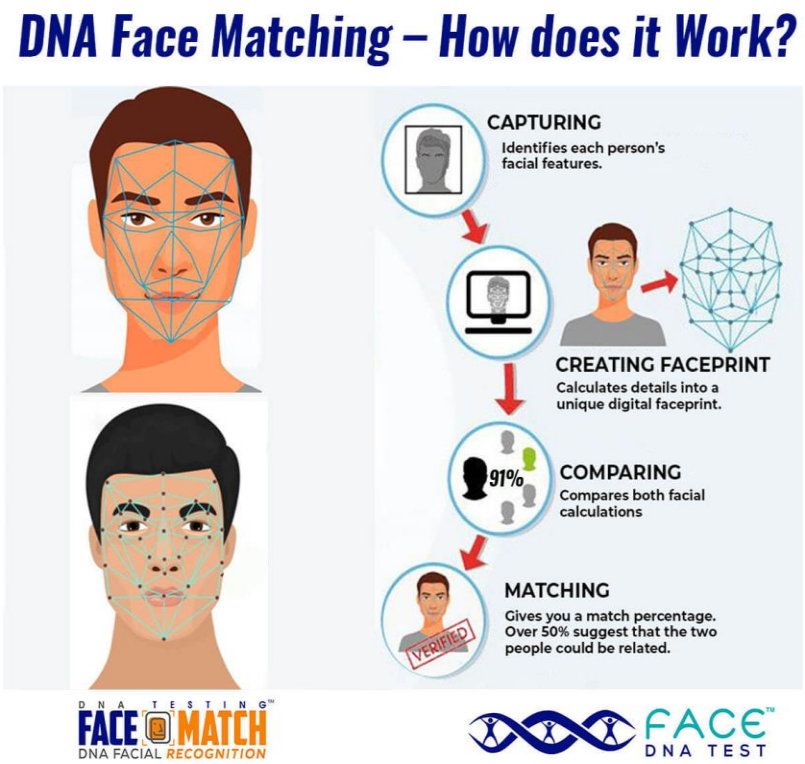
The US Department of Homeland Security predicts that facial recognition will be used on 97% of travelers by 2023.

2- Face DNA Test

DNA Face Matching is a technique that allows experts to compare one person's face to another and suggest a possibility of relationship.

This is commonly done with a father & Child, but with recent advancements, DNA Face Matching now available to Grandparents, Aunts, Uncles, Siblings, and potential Cousins.

Our customers tell us DNA Face Matching is an awesome way to verify if the results you've received from an **Ancestry DNA test** are truly accurate. Cool right! Once you receive your ancestry test report from any company, including ours, perform a DNA Face Matching analysis. Its lots of fun and results are fast.



DNA Face Matching uses a similar theory as present day Facial Recognition Technology. Facial Recognition is generally completed in three steps: Detection, Faceprint Creation, and Verification or Identification.

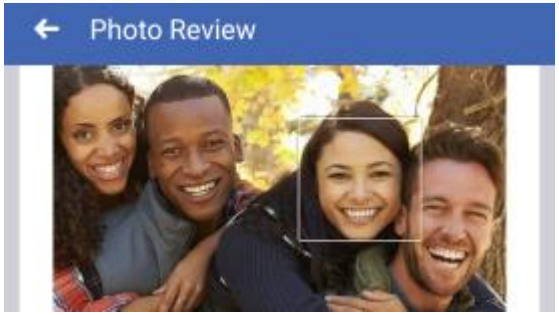
3- Railer

Face Recognition Mobile Attendance & Shift Management - Mobile Application

- Employees check in and out quickly using face recognition on the mobile
- roster and shift management
- time attendance management

4- Facebook

Facebook which uses face recognition technology to automatically recognize when Facebook members appear in photos.



This makes it easier for people to find photos they are in and can suggest when particular people should be tagged in photos

Academic Publications

Most current face recognition algorithms perform relatively well in a strictly constrained situation, where well aligned, well illuminated, and frontal pose face images are considered for performance evaluation to ensure the higher performance.

Under controlled image acquiring constraints, it is possible to capture high quality images and attain an impressive accuracy with a very low error rate.

However, recognition accuracies substantially decrease when the captured images do not have enough quality either due to subject's alignment problem to the camera, various facial expressions, gaze deviations or facial hair. Several researchers proposed different facial recognition.

Review of Literatures

Paper 1

An Improved Artificial Neural Network Design for Face Recognition - Utilizing Harmony Search Algorithm

Face recognition has become an interesting field for researchers where it is used in many applications. One of the most common methods of soft computing is named the artificial neural network (ANN) has been suggested to achieve the face recognition process. Nonetheless, the performance of ANN depends on the number of neurons in the hidden layers and the value of the learning rate.

These variables are usually defined based on the trial-and-error method which is time-consuming. Furthermore, in many cases, it is very difficult to find the optimum value for these variables.

Therefore, this paper introduces an improved ANN design for face recognition using a meta-heuristic optimization algorithm. The ANN represents a distributed processing system consists of neurons which are simply connected elements.

The results revealed that the proposed system (HSA-ANN) achieved lower MSE compare with the ANN. Furthermore, the HSA-ANN gives a better face recognition rate than traditional ANN.

Resource: **OP Conference Series Materials Science and Engineering**

Paper 2

Face recognition with illumination, scale, and rotation invariance - Using multiblock LTP-GLCM descriptor and adaptive ANN

Face recognition has recently gained significant attention as one of the most useful image analysis applications. By leveraging their unique but incredible identification skills, these systems are capable of recognizing users. Face recognition systems have been extensively studied. The system, however, has several drawbacks.

Existing face recognition methods may result in a longer histogram, which slows down for a large-scale database. To address the challenges with face recognition, we have proposed a hybrid descriptor using Multiblock Local Ternary Pattern (LTP)—Gray Level Co-occurrence Matrix (GLCM). In this study, we have employed the LTP, GLCM, and Speeded Up Robust Features (SURF) methods to extract the illumination, rotation, and scale-invariant features of the face database images.

These features are then trained using Artificial Neural Network. The layer neurons are optimally selected by Crow Search Optimization (CSO) method yielded an accuracy of 95%. The proposed approach was implemented in the MATLAB software, and the experimental data were analyzed to show that the developed texture descriptor has a higher recognition rate than existing methods.

Resource: International Journal of Systems Assurance Engineering and Management

Paper 3

Smart Door Lock Using Face Recognition

Most doors are controlled by persons using keys, security cards, passwords, or patterns to open the door. This paper aims to help users improve the door security of sensitive locations by using face detection and Recognition. The face is a complex multidimensional structure and needs good computing techniques for detection and Recognition.

Face detection is the process of detecting the region of the face in an image. The look is seen using the viola jones method, and face recognition is implemented using the Principal Component Analysis (PCA). Face Recognition based on PCA is generally referred to as the use of Eigenfaces. If a face is recognized, it is known, else it is unknown. The door will open automatically for the known person due to the command of the microcontroller.

On the other hand, the alarm will ring for the unknown person. Since PCA reduces the dimensions of face images without losing essential features, facial images for many persons can be stored in the database. Although many training images are used, computational efficiency cannot be decreased significantly. Therefore, face recognition using PCA can be more beneficial for door security systems than other face recognition schemes.

Resource: International Journal of Computer Applications Technology and Research

Paper 4

Multi-view Face Recognition by Neural Network

Multiple views face recognition has become significant in various requisitions, such as observation, human workstation connection, and recreation.

A reduction-based feature extraction and neural network inspired by biological neurons for learning and recognizing the multiple views faces of the person has been presented in this paper.

Neural Network (NN) is significant in the places where formulating an algorithmic solution is difficult and we need to retrieve the structure from existing and predefined data. Multi-view face recognition is required here because its more feasible and reliable than single view face recognition

Resource: International Journal of Knowledge Based Computer Systems

Paper 5

3D Face Recognition Neural Network for Digital Human Resource Management

Digital human resource management can improve the organizational and operational efficiency of enterprises. To improve the efficiency of enterprise digital management and solve the problems of low security level and insufficient stability of 2D face recognition, we introduce 3D face recognition into the digital human resource management system. We propose a face recognition method based on a multi stream convolutional neural network and local binary pattern and build a digital face recognition management system.

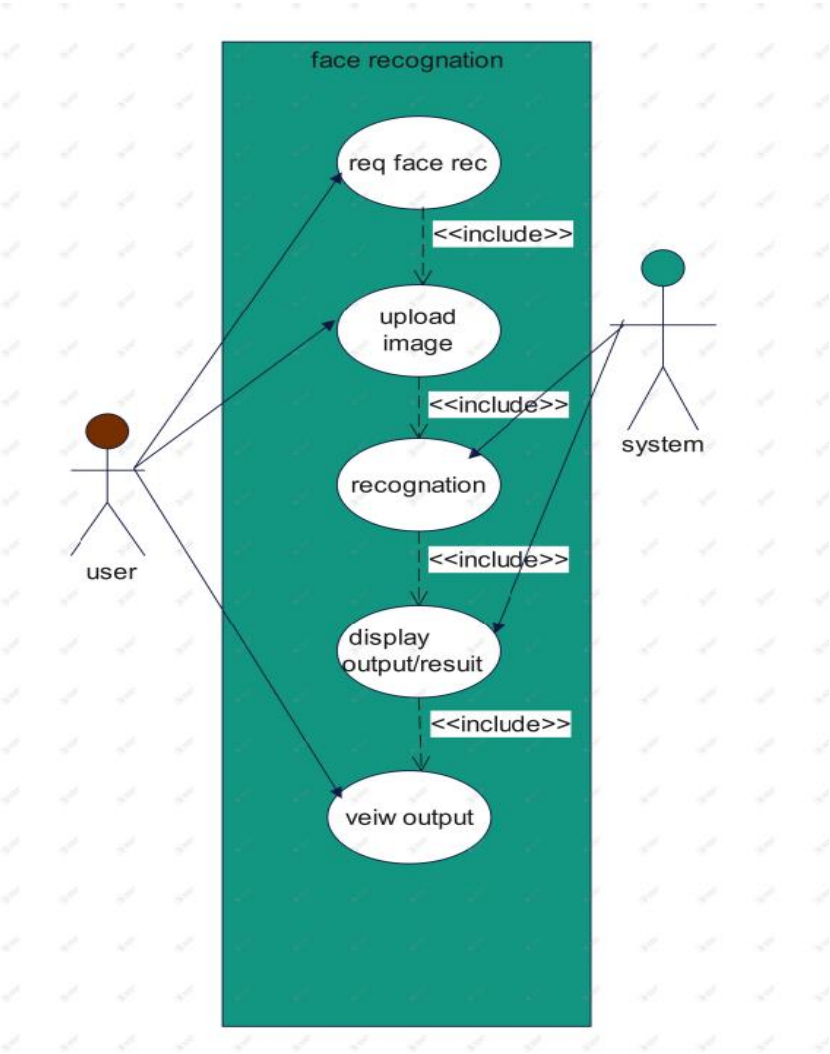
We first build the system computer vision scene. Then a local binary mode facial expression feature extraction scheme is designed according to the depth camera image extraction method. Considering that face 3D features are easy to be missed, we build a multi stream convolutional neural network to learn facial 3D features.

Finally, we validate the effectiveness of the method in selecting a public face dataset. Experiments prove that our method can reach 98% face recognition accuracy, which is significantly better than other methods.

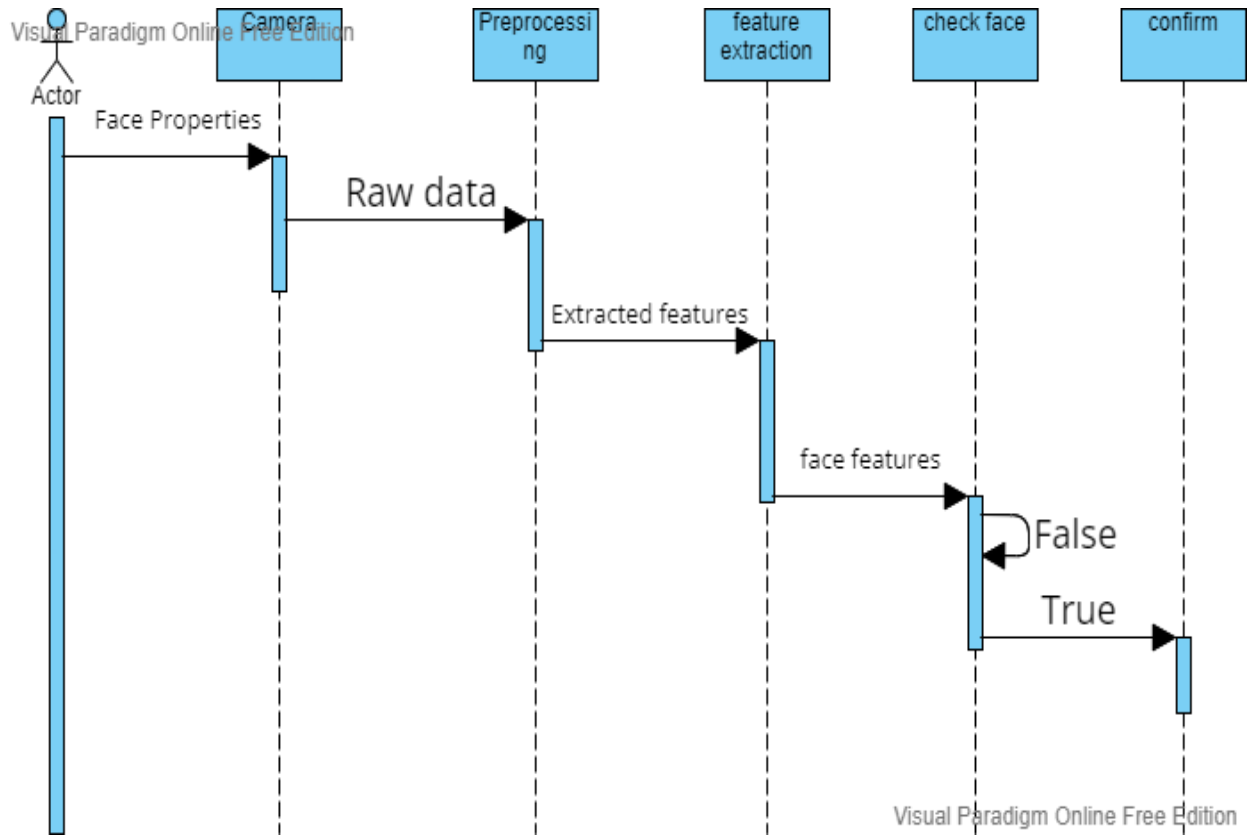
Resource: Scientific Programming

Proposed Solution & Dataset

Use Case Diagram



Sequence Diagram



Datasets Used

❖ Face Recognition on Olivetti Dataset

Face images taken between April 1992 and April 1994.

There are ten different images of each of 40 distinct people.

There are 400 face images in the dataset.

Face images were taken at different times, varying lighting, facial express and facial detail.

Link: https://www.kaggle.com/code/serkanpeldek/face-recognition-on-olivetti-dataset/notebook

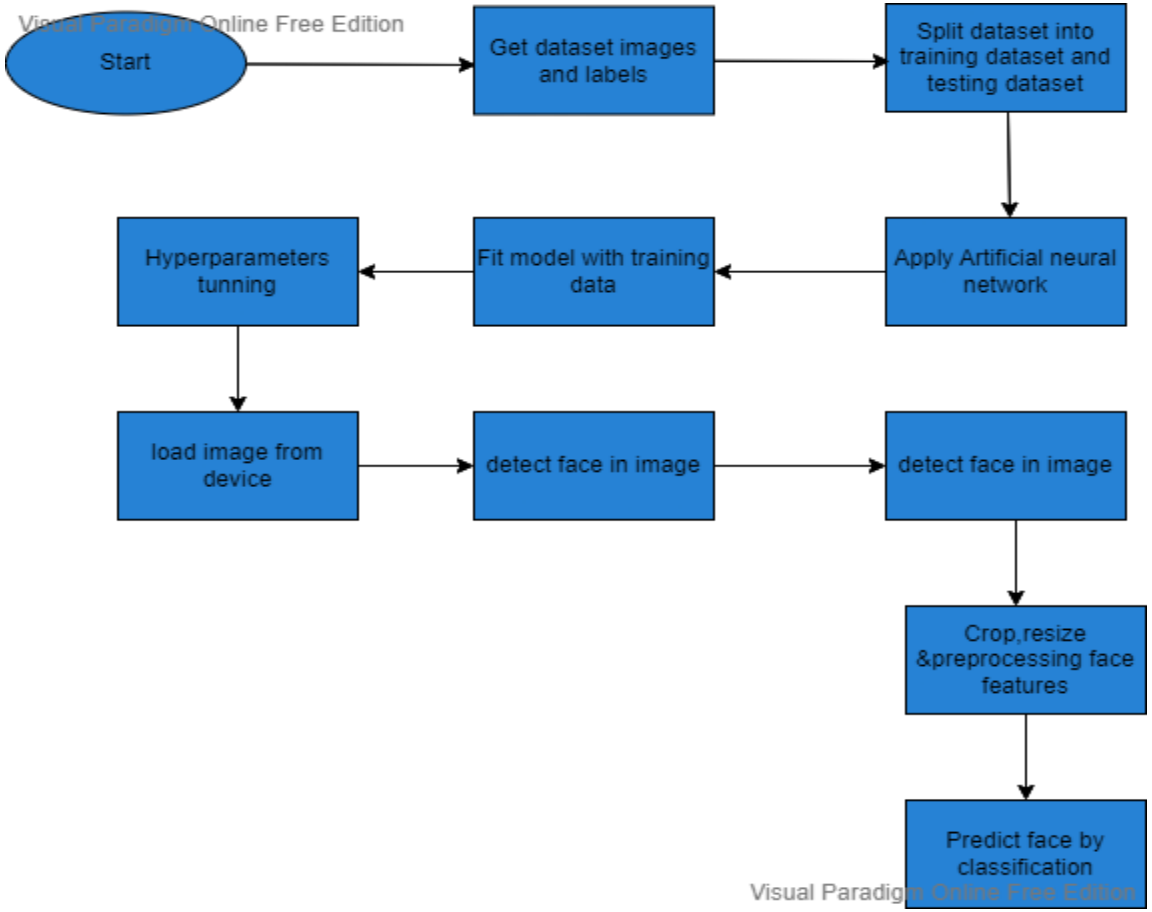
❖ Labeled Faces in the Wild

A database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web.
Each face has been labeled with the name of the person pictured.
1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector.

Link: http://vis-www.cs.umass.edu/lfw/

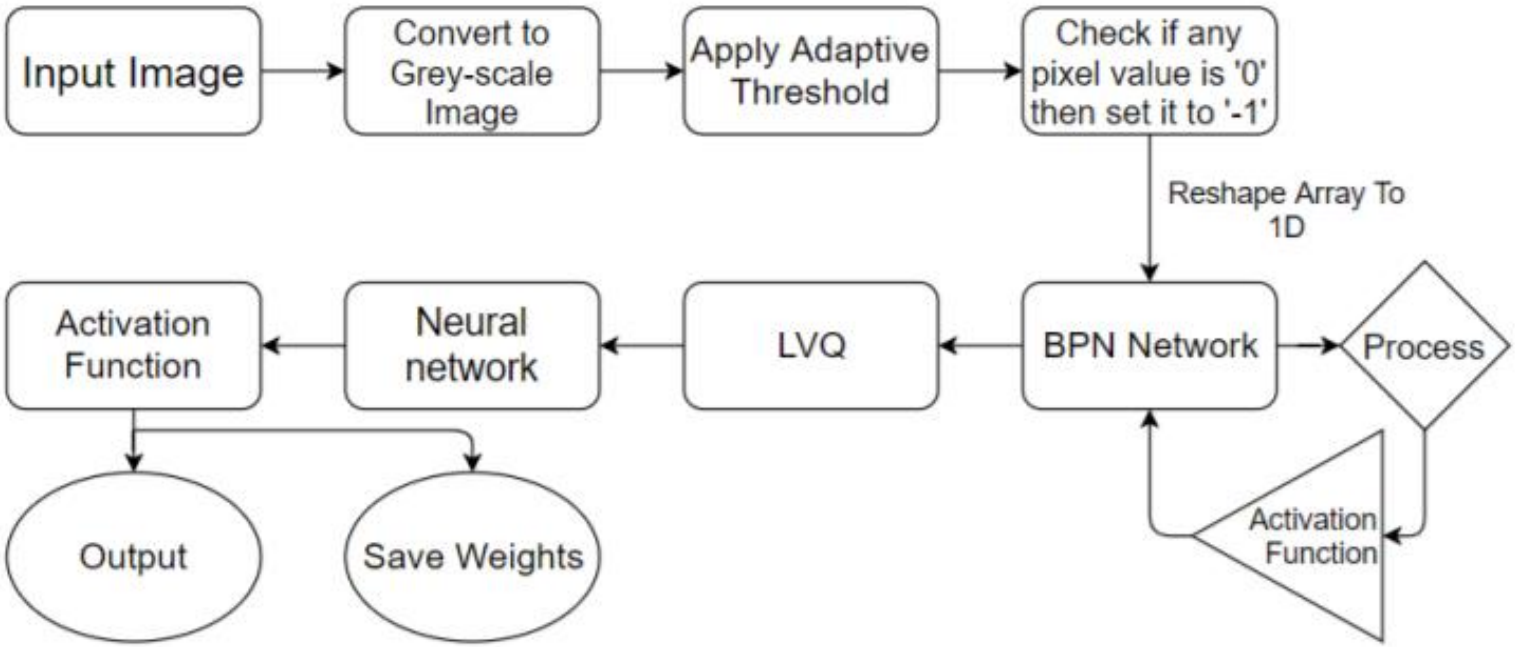
Applied Algorithm

Block Diagram

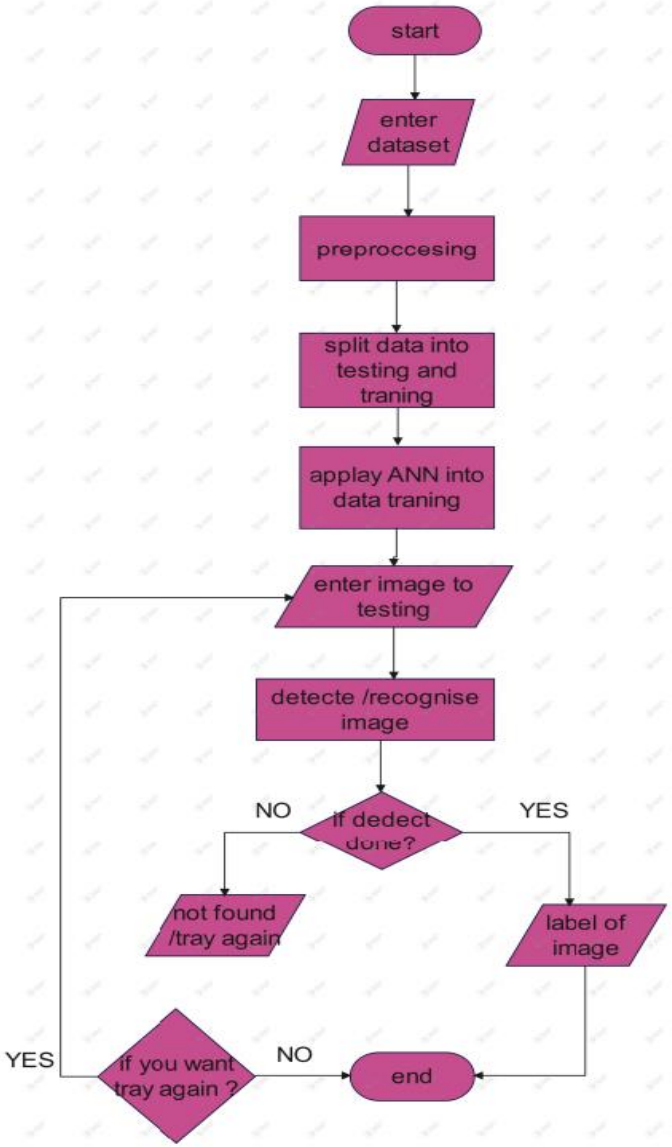


Artificial Neural Network (ANN)

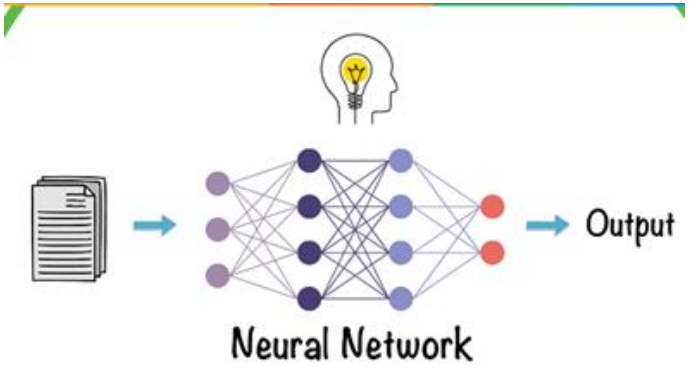
Overview



Network For Training


start
enter dataset
preproccesing
split data into testing and traning
applay ANN into data traning
enter image to testing
detecte /recognise image
if dedect done?
NO
YES
not found /tray again
label of image
if you want tray again ?
YES
NO
end

ANN Overview

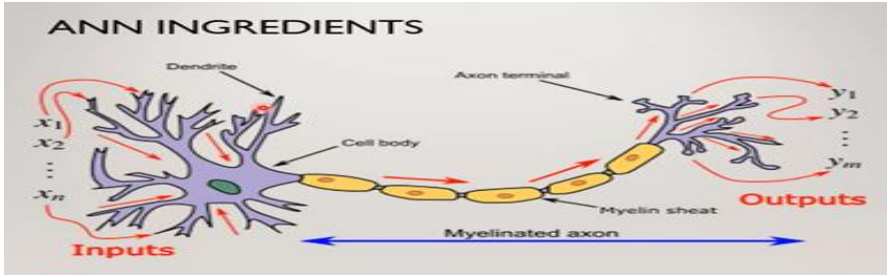


Neural network is a very powerful and robust classification technique which can be used for predicting not only for the known data, but also for the unknown data. It works well for both linear and nonlinear separable dataset. NN has been used in many areas such as interpreting visual scenes, speech recognition, face recognition, fingerprint recognition, iris recognition etc.

An ANN is composed of a network of artificial neurons also known as "nodes".

These nodes are connected to each other, and the strength of their connections to one another is assigned a value based on their strength: inhibition (maximum being -1.0) or excitation (maximum being +1.0). If the value of the connection is high, then it indicates that there is a strong connection. Within each node's design, a transfer function is built in. There are three types of layers in an ANN, input layers, hidden layers, and output layers.

ANN Logic



There is a dendrite from which the input enter cell body (neurons) in which mathematical operations occur depending on the input , and axon terminal in which there are ouytputs.

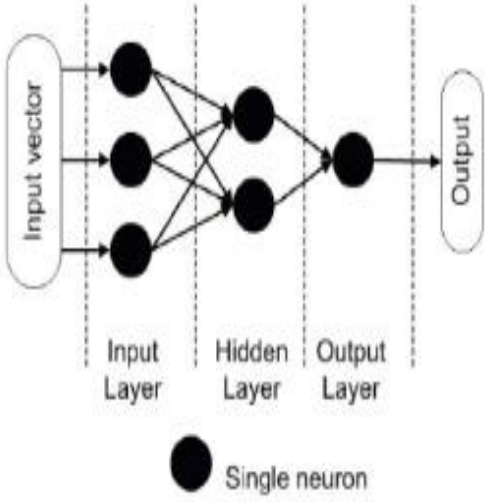
ANN are modeled after the human brain. That is, just like how the neurons in our nervous system are able to learn from the past data, similarly, the ANN is able to learn from the data and provide responses in the form of predictions or classifications.

Topologies of Neural Network

A Feed-Forward Network (forward propagation) FFN:

When we divide the image to pixels each pixel fed as input to each neuron of the first layer neurons of one layer are connected to neurons of the next layer through channels.

Each of these channels is assigned a numerical value known as weight, the inputs are multiplied to the corresponding weights and their sum is sent as input to the neurons in the hidden layer.

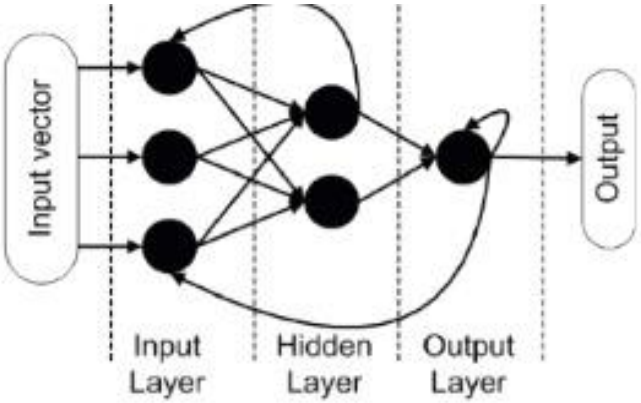


Each of these neurons is associated with a numerical value called the bias which is then added to the result of the activation function determines if the neuron will get activated or not to the input.

Some value is then passed through a threshold function called the activation function, activated neuron transmits data to the neurons of the next layer over the channels in this manner the data is propagated through the network this is called forward propagation. This a non-recurrent network which contains inputs, outputs, and hidden layers.

Recurrent Neural Network (back propagation) RNN:

In the output layer the neuron with the highest value fires and determines the output.



The values are basically, a probability so the shape with the highest value is selected to be the goal output predicted by the neural network of course, just by a look at it, we know our neural network has made a wrong prediction.

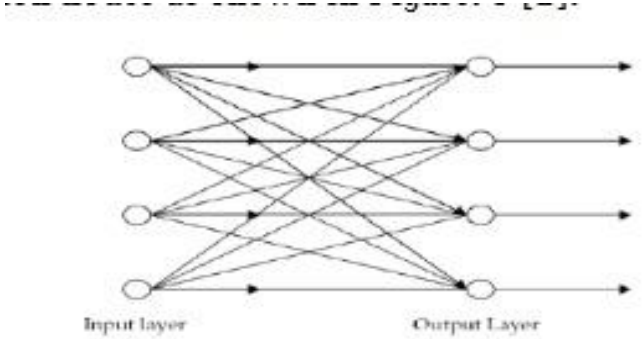
But how does the network figure this out note that our network is yet to be trained during this training process along with the input. Our network also has the output fed to it; the predicted output is compared against the actual output to realize the error in predicting.

The magnitude of the error indicates how wrong we are, and the sign suggests if our predicted values are higher or lower than expected.
The arrows here give an indication of the direction and magnitude of change to reduce the error. This information is then transferred backward through our network this is known as backpropagation now, repeating this cycle to learn the correct classification.

Layers

Single Layer Feed Forward Network:

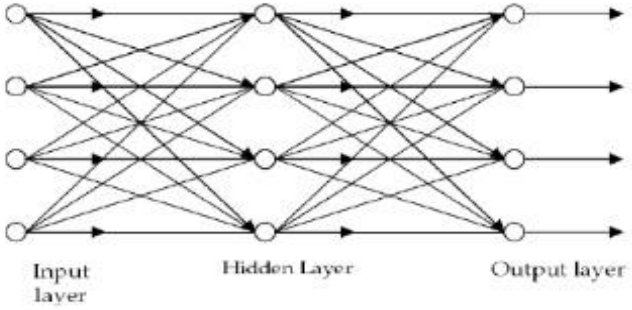
A neural network in which the input layer of source nodes is connected to an output layer of neurons, but not vice-versa is known as single feed-forward or acyclic network.
In single layer network "single layer" refers to the output layer of computation nodes.



Multilayer Feed Forward Network:

This type consists of one or more hidden layers, called hidden neurons or hidden units.

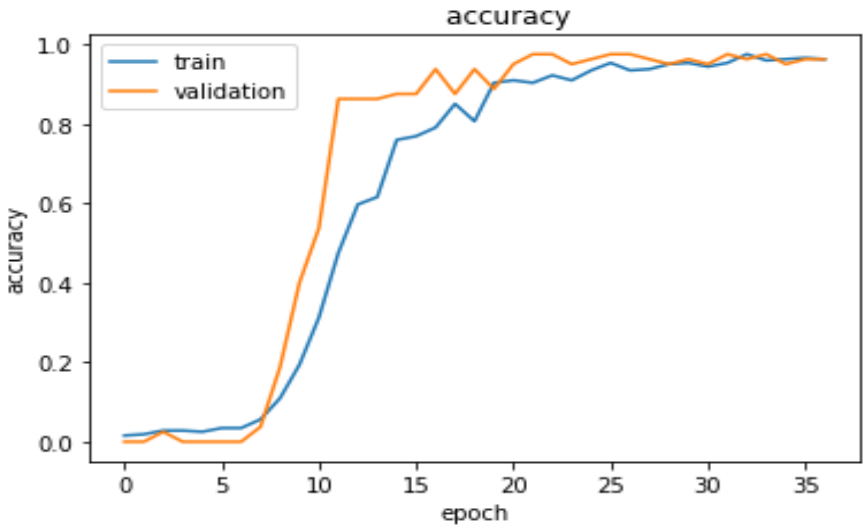
The function of hidden neurons is to interact between the external input and network. The source nodes in input layer of network supply the input signal to neurons in the second layer of 1st hidden layer.



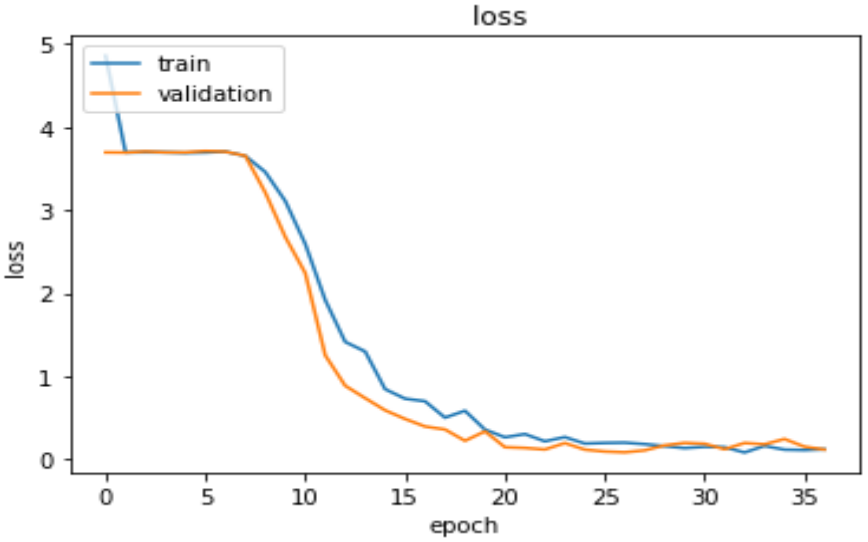
The output signals of 2nd layer are used as inputs to the third layer and so on. The set of output signals of the neurons in the output layer of network constitutes the overall response of network to the activation pattern supplied by source nodes in the input first layer [2].

Experiments & Results

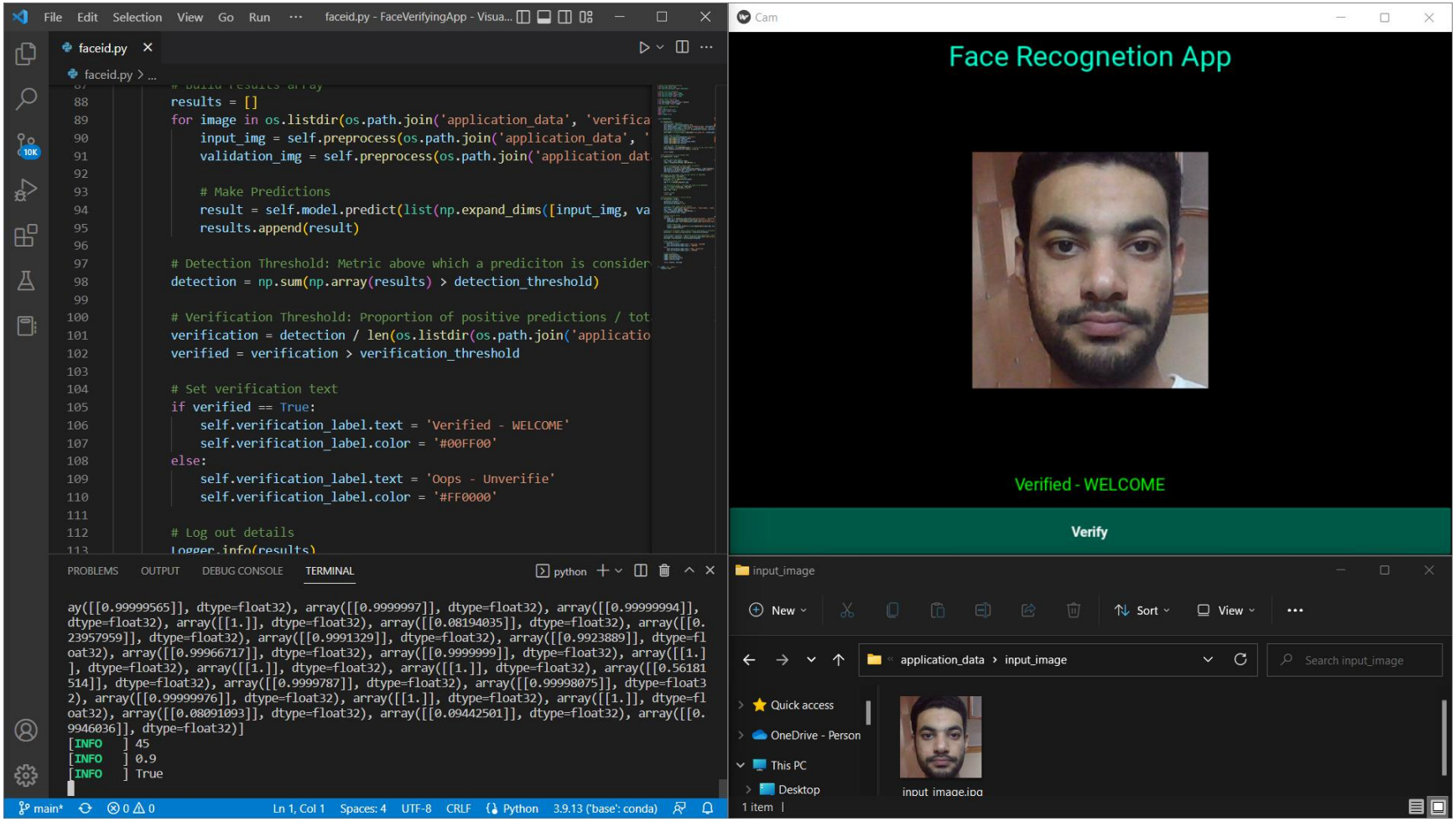
- **Accuracy Curve**



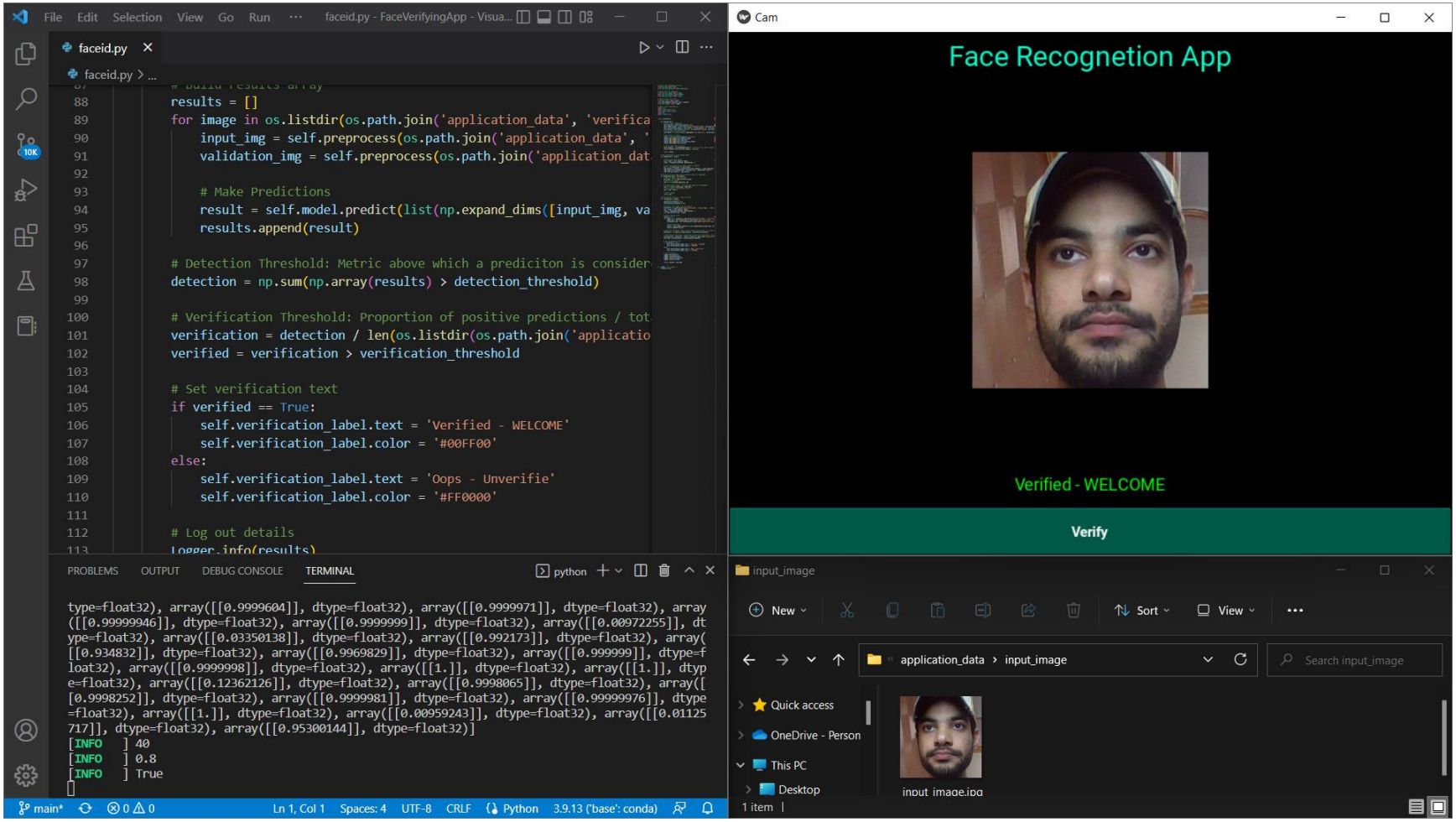
- **Loss Curve**



Test Case 1


faceid.py - FaceVerifyingApp - Visua...
faceid.py
results = []
for image in os.listdir(os.path.join('application_data', 'verifica
input_img = self.preprocess(os.path.join('application_data', '
validation_img = self.preprocess(os.path.join('application_dat
# Make Predictions
result = self.model.predict(list(np.expand_dims([input_img, va
results.append(result)
# Detection Threshold: Metric above which a prediciton is consider
detection = np.sum(np.array(results) > detection_threshold)
# Verification Threshold: Proportion of positive predictions / tot
verification = detection / len(os.listdir(os.path.join('applicatio
verified = verification > verification_threshold
# Set verification text
if verified == True:
self.verification_label.text = 'Verified - WELCOME'
self.verification_label.color = '#00FF00'
else:
self.verification_label.text = 'Oops - Unverifie'
self.verification_label.color = '#FF0000'
# Log out details
Logger.info(results)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
python
[INFO] 45
[INFO] 0.9
[INFO] True
main*
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.9.13 ('base': conda)
Cam
Face Recognetion App
Verified - WELCOME
Verify
input_image
New
Sort
View
application_data › input_image
Search input_image
Quick access
OneDrive - Person
This PC
Desktop
input_image.jpg
1 item

Test Case 2


faceid.py - FaceVerifyingApp - Visua...
faceid.py
faceid.py > ...
results = []
for image in os.listdir(os.path.join('application_data', 'verifica
input_img = self.preprocess(os.path.join('application_data', '
validation_img = self.preprocess(os.path.join('application_dat
# Make Predictions
result = self.model.predict(list(np.expand_dims([input_img, va
results.append(result)
# Detection Threshold: Metric above which a prediciton is consider
detection = np.sum(np.array(results) > detection_threshold)
# Verification Threshold: Proportion of positive predictions / tot
verification = detection / len(os.listdir(os.path.join('applicatio
verified = verification > verification_threshold
# Set verification text
if verified == True:
self.verification_label.text = 'Verified - WELCOME'
self.verification_label.color = '#00FF00'
else:
self.verification_label.text = 'Oops - Unverifie'
self.verification_label.color = '#FF0000'
# Log out details
Logger.info(results)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
python
type=float32), array([[0.9999604]], dtype=float32), array([[0.9999971]], dtype=float32), array
([[0.99999946]], dtype=float32), array([[0.9999999]], dtype=float32), array([[0.00972255]], dt
ype=float32), array([[0.03350138]], dtype=float32), array([[0.992173]], dtype=float32), array(
[[0.934832]], dtype=float32), array([[0.9969829]], dtype=float32), array([[0.999999]], dtype=f
loat32), array([[0.9999998]], dtype=float32), array([[1.]], dtype=float32), array([[1.]], dtyp
e=float32), array([[0.12362126]], dtype=float32), array([[0.9998065]], dtype=float32), array([
[0.9998252]], dtype=float32), array([[0.9999981]], dtype=float32), array([[0.99999976]], dtype
=float32), array([[1.]], dtype=float32), array([[0.00959243]], dtype=float32), array([[0.01125
717]], dtype=float32), array([[0.95300144]], dtype=float32)]
[INFO] 40
[INFO] 0.8
[INFO] True
main*
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.9.13 ('base': conda)
Cam
Face Recognetion App
Verified - WELCOME
Verify
input_image
New
Sort
View
application_data › input_image
Search input_image
Quick access
OneDrive - Person
This PC
Desktop
input_image.jpg
1 item

Platform & Libraries

Desktop Python GUI:

- KIVY Library

Model Libraries:

- TensorFlow
- CV2
- OS
- Matplotlib