

Face Detection – Recognition System Project

Introduction

The objective of this project is to develop a mode that predicts the name of an individual by analyzing their facial feature. This project for learning and deeper understanding about computer vision and how to deploy model on website.

Index

Introduction.....	1
Used Python Library:.....	3
Overview System	3
Dataset	4
Approach	5
Face Detection	5
Face Recognition	6
Trainning	7
SQL Server for taking attention.....	8
Application.....	9
Conclusion.....	9

Used Python Library:

- Keras/Tensorflow: Used to build model
- Matplotlib, Numpy, Pandas: These libraries are commonly used in data preprocessing or analysis.
- OpenCV: Used in image processing.
- Skicit-learn: Used to evaluate model's accuracy.

Overview System

The images are captured by a camera that is installed on my computer then transfer to machine learning model through Flask application programming interface (API). When the model receives an image, the facial landmarks from this can be extracted using the OpenCV library. This library supports the engine in proccessing the image and so detecting and recognizing the faces. Below is a simple picture that provides system overview and image processing.

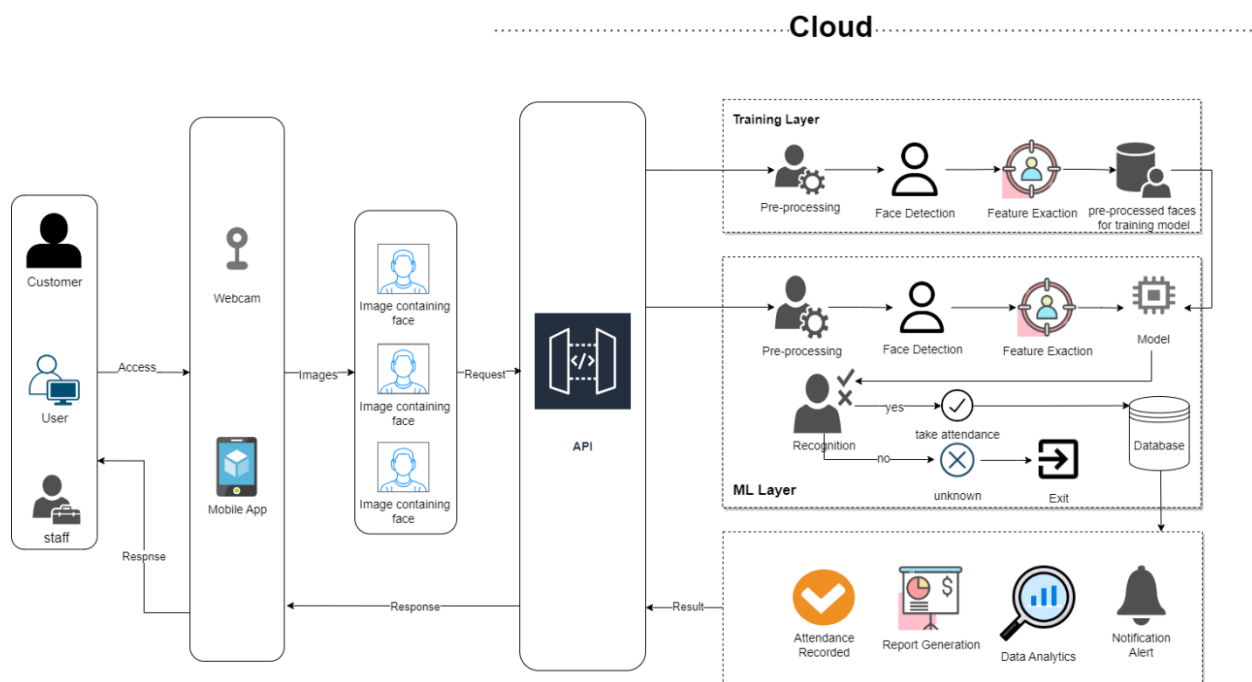
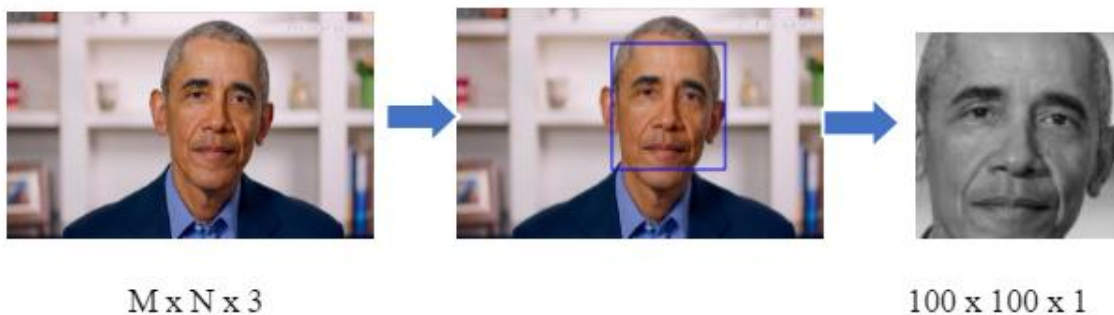


Figure 1: System Architecture

This like

Dataset

In an image, I just need facial region to extract facial feature so I crop the necessary region from image and store it on dataset folder. As shown in Figure 2,3,4, the dataset included the photographs of Donald Trump, Vladimir Vladimirovich Putin, Barack Obama. Each individual I took 50 images which were convert to gray and reshape 100x100x1.



By using Haar Cascade algorithms to detect face region, it is possible to automatically label each face in the dataset. This can significantly reduce the amount of manual effort required to label each face in the dataset. By converting the Region On Interest (ROI) images to grayscale, it can reduce the dimensionality of the data while keeping important feature. Additionally, resizing the images to smaller size (such as 100x100x1) can help to reduce the computational complexity of the model during both traning and inference phases processing.

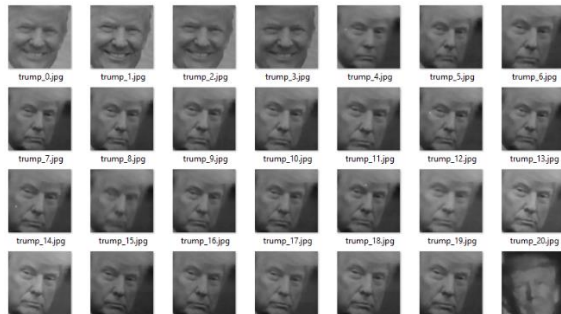


Figure 2: Trump dataset

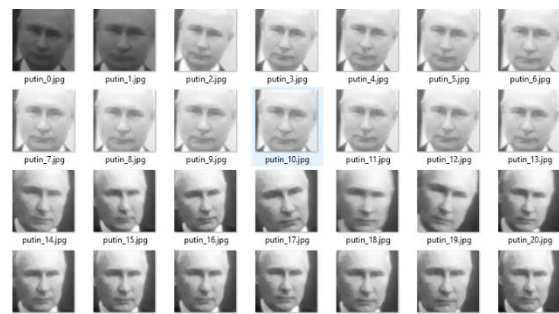


Figure 3: Putin dataset



Figure 4: Obama dataset

Besides, there is a class “unknown”, which will show label “unknown” if there is no one of above three class, has 50 images of different people from internet.

Approach

Face Detection

Haar Cascade is an algorithm that can detect faces in images, irrespective of their scale in image and location. This algorithm is not complex and can run in real-time, proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001.

Haar Features were not only used to detect faces, but also for eyes, lips, license number plates etc. The models are stored on GitHub, and we can access them with OpenCV methods.

When we obtain an image, we will slide a fixed size window across our image at multiple scales. At each of these phases, our window stops, computes some features, and then classifies the region as *Yes, this region does contain a face*, or *No, this region does not contain a face*. Below is a figure that shows how this algorithm works.

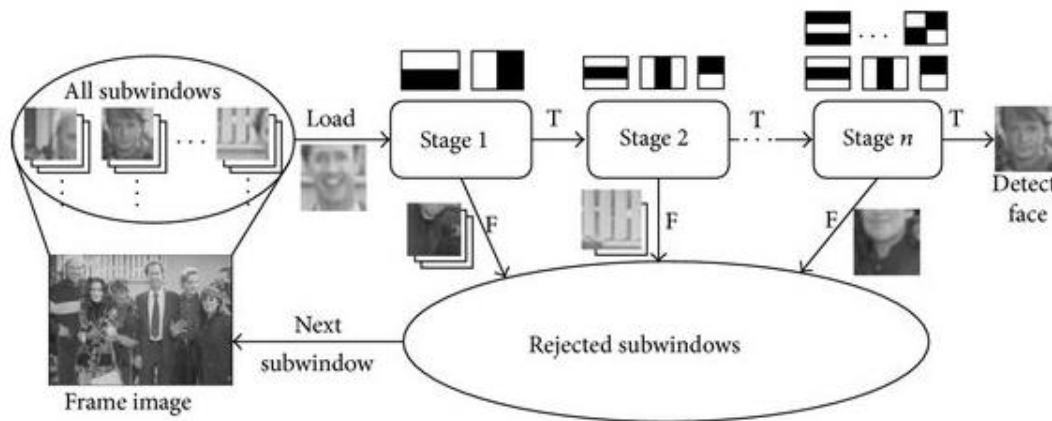


Figure 5: Haar Cascade algorithm processing

Face Recognition

In face recognition, I used 4 convolutional layers, 4 max-pooling layers, 1 fully-connected layer and 1 softmax output layer to extract facial features from input images.

Each convolutional layer is followed by a max-pooling layer, which downsamples the feature maps produced by convolutional layer to reduce the dimensionality of the data and extract the most important feature. The combination of convolution and max-pooling helps the network to learn hierarchical representations of the input images by detecting increasingly complex features at deeper layers of network.

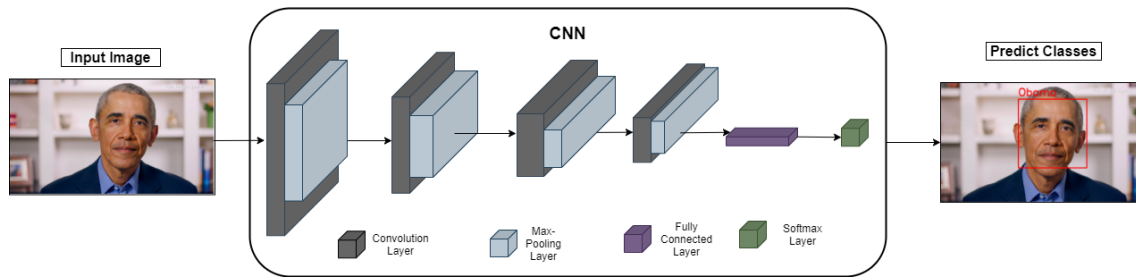


Figure 6: Architecture of CNN for face recognition

During the building process, I conducted training with different number of network layers. Finally, I decided to choose 4 of hidden layer because when training with number of hidden layers is less than 4, the accuracy of model decreases significantly.

Training

I trained this model with 20 epochs, batch size is 8. Due to classification model, I used loss function is categorical_crossentropy, optimizer is Adam to train model and figure 7 demonstrates accuracy and loss in training process.

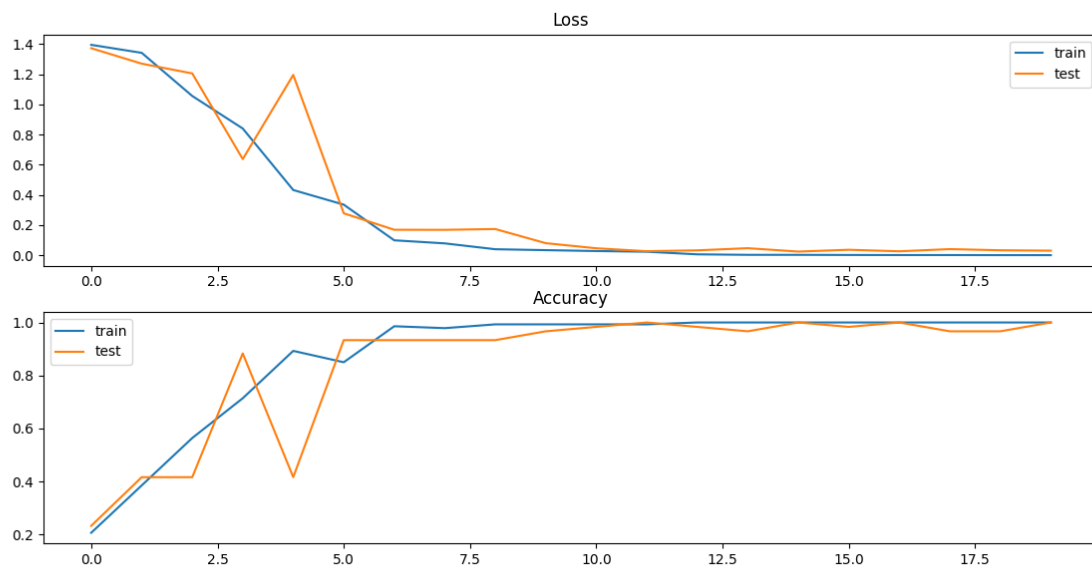


Figure 7: Accuracy and Loss

We can observe that model converges quickly because of small and simple dataset. This may be result overfitting, so we can overcome this issue by using a larger dataset with a lot of difference facial images. This will help model to learn more diverse feature.

SQL Server for taking attention

I wanted to expand my project so I used this model for attendance system. I used SQL Server to save time when specific face appears. I designed a simple ERD for this task.

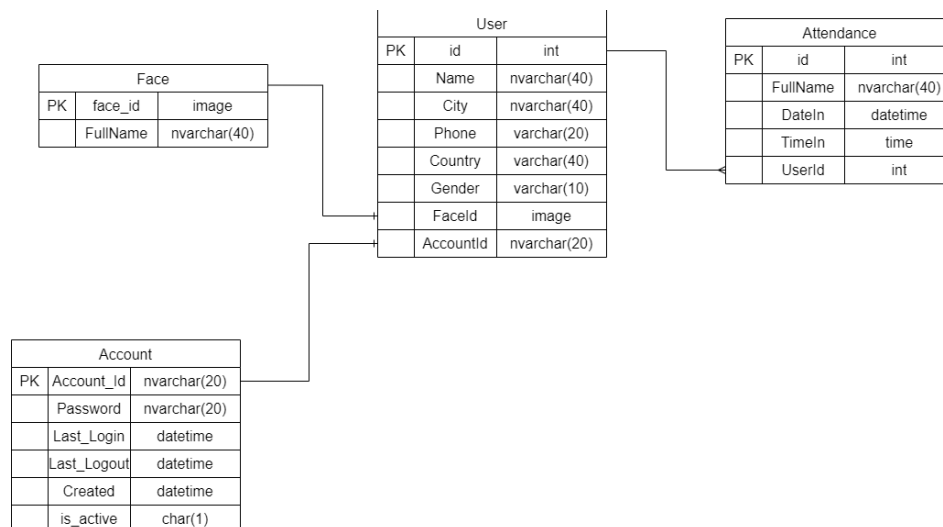


Figure 8: ERD

When there is one of three people who I mentioned appears, I will save time (hhmmss) into attitude TimeIn in table Attendance. So if I want to know who appeared on my camera, I just query table Attendance join with table User like this “*select * from Attendance join User on User.id = Attendance.UserId*” and I can get result like figure 9

Results		Messages								
	Id	FullName	DateIn	TimeIn	UserId	Id	FullName	City	Country	Phone
861	8...	Putin	2023-07-21 00:00:00.000	14:23:27.00000000	3	3	Putin	St. Petersburg	Russia	0123123123
862	8...	Obama	2023-07-21 00:00:00.000	14:23:28.00000000	2	2	Obama	Honolulu	USA	9876543201
863	8...	Trump	2023-07-21 00:00:00.000	14:23:28.00000000	1	1	Trump	New York	USA	0123456789

Figure 9: Experiment

Application

I want to deploy this system on web app so I used Flask library, which support develop API for web. The reason I chose Flask library to develop is simple and convenient for beginners like me. Besides, there are many strong library which we can use in this project such as RestAPI, Django, etc. But in this project I just use Flask to develop backend for attendance system. May be I will make a completed attendance system which will have frontend in the future.

I will effort to explore and implement new technologies paved the way for future projects and further enhanced my skills in software development.

Conclusion

The goal of this project was to learn and advance my skills in AI Engineer so I always think how to extend my project and implement my idea. Overall, the project was pretty successful in its showcasing who CNN can be implement in Flask to create a web application and how to connect SQL Server from IDE and save time on it