Media Engineering and Technology Faculty
German University in Cairo

Eyes on the City: Leveraging Multi-View Face Detection for Proactive Security in Smart Cities

Bachelor Thesis

Author:	Abdelrhman Atef Yakout
Supervisors:	Prof. Dr. Mohammed Abdel-Megeed Salem
	Prof. Dr. Mohamed Abdel-Ghany
Mentor:	Eng. Yomna Islam Youssef Alayary
Submission Date:	19 May, 2024

Media Engineering and Technology Faculty
German University in Cairo

Eyes on the City: Leveraging Multi-View Face Detection for Proactive Security in Smart Cities

Bachelor Thesis

Author:	Abdelrhman Atef Yakout
Supervisors:	Prof. Dr. Mohammed Abdel-Megeed Salem
	Prof. Dr. Mohamed Abdel-Ghany
Mentor:	Eng. Yomna Islam Youssef Alayary
Submission Date:	19 May, 2024

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgment has been made in the text to all other material used

Abdelrhman Atef Yakout
19 May, 2024

Acknowledgments

My deepest gratitude to Prof. Mohamed Salem for their most appreciated support and guidance. Thanks also to my mentor Eng. Yomna Islam, for their time in aiding me with the bachelor.

I would also like to thank my family for their love and belief in me. To my friends, I thank you for your constant support and motivation.

Abstract

With the huge number of video surveillance cameras to this day, it necessitates an efficient system for monitoring and security. This thesis explores the application of real-time face detection and recognition for identifying individuals in facilities and age, gender and race (AGR) detection for data analysis in video surveillance using deep learning techniques and convolutional neural networks(CNN).

Our research aims to create a robust model capable of operating under various conditions such as different lighting conditions, angles and occlusions. when experimenting with 3 different face detection models on surveillance footage, the pre-trained ResNet model. Additionally, trained a YOLOV9 model on WIDER FACE dataset with 30 epochs due to certain limitations(4.3.1), the model had an average precision of 92%. Finally, RetinaFace model was used in the final model.

The model created has 3 stages that each frame pass through: face detection, AGR detection and face recognition. After testing the model on different surveillance footage, the result shown had limited accuracy due to the struggle of the AGR and face recognition model on low resolution faces in the footage as all facial features has to be clearly visible. Otherwise these conditions the AGR and face recognition performed successfully with no issues. Additionally RetinaFace face detection model has an average precision of 96.1%.

This research investigates real-time face detection and recognition within smart cities, leveraging deep learning techniques and Convolutional Neural Networks (CNNs). The model goes beyond simple identification by **additionally** estimating a person's Age, Gender, and Race (AGR). This comprehensive approach using deep learning enhances security measures for smarter and safer environments.

With the numerous applications of facial recognition and detection technologies. It have become more popular in recent years. These technologies are very important to a wide range of sectors, which include monitoring access control systems and conducting theft investigations, improving the overall security of an environment.

How it works?

For Face detection RetinaFace model was used with an average precision of 96.1% on WIDER FACE dataset. For AGR detection, a model called DeepFace was used with an accuracy of 97%. Finally for recognition, a python library called face_recognition was used having an accuracy of 99.38% on LFW dataset.

Stage 1

In each frame all the faces are detected using RetinaFace model and their coordinates are saved for the next stages.

Stage 2

For all the face detected in the first stage, age, gender and race are computed using the DeepFace model.

Stage 3

For all the face detected in the first stage, the model encodes the face detected and compare it to the database to see if there is a match.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Due to their numerous applications facial recognition and detection technologies have become more and more popular in recent years. These technologies are very important to a wide range of sectors. Consider the purposes of security and surveillance which include monitoring access control systems and conducting theft investigations. These activities require the identification of individuals present in public spaces. Contributing to law enforcements criminal identification and investigation procedures. Enabling biometric authentication for secure access control systems in a variety of contexts including financial institutions personal devices and border control. Personalizing user experiences in marketing entertainment and social media domains as well.

Figure 1.1: Applications on video surveillance in smart cities.

Face detection and recognition technologies in addition to enhancing convenience and security hold a great promise for enhancing social welfare and public safety. These technologies are particularly useful in helping vulnerable populations such as children and the

elderly locate missing individuals. In addition to helping identify accident or disaster victims facial recognition can also speed up emergency response times and facilitate family reunions. Additionally the possibility of being recognized can discourage criminal activity making the environment safer for everybody. Moreover facial recognition technology can be a useful instrument for expediting emergency response activities. First responders can deliver critical aid more efficiently and possibly save lives by enabling faster and more accurate identification of individuals in need.

1.2 Problem Statement

Even with these advances there are still a number of issues with face recognition especially when there are multiple views involved. Pose variations make it challenging for conventional methods to reliably detect and recognize individuals because faces are captured from different angles. Illumination variations can also have a substantial impact on how facial features appear making recognition accuracy more difficult. The recognition process can also be made more difficult by partial or total occlusions brought on by hair accessories or other objects. Finally practical applications frequently demand real-time processing capabilities in order to produce effective and instantaneous results. Convolutional neural networks (CNN) a type of deep learning has been suggested as a solution to these problems. The field of face detection and recognition has undergone a revolution. These effective methods have many benefits. High-level feature extraction: in this process facial image data is automatically processed by CNNs to extract robust complex features—even in the face of difficult obstacles like occlusions and pose changes. increased recognition accuracy because deep learning models outperform conventional techniques in face recognition tasks on a regular basis. The real-time processing potential of CNN-based solutions has been made possible by improvements in hardware and optimized algorithms which qualify them for useful applications. With the use of CNN-powered deep learning and real-time processing this project seeks to create a multi-view face detection and recognition system. This system uses deep learning techniques to improve accuracy and robustness and enable real-time processing capabilities thereby addressing the aforementioned challenges.

1.3 Objectives

The objectives of this work are:

- Review of existing literature on face detection and recognition.
- Development of a multi-view face detection using convolutional neural networks and deep learning.

- Development of face recognition using deep learning and convolutional neural networks.
- Integrating both models together.
- Initial evaluation of the model's performance across diverse lighting conditions and angles.
- Enhancing initial results.

1.4 Thesis Outline

In this thesis we will explore all the steps that was taken from literature review to implementation to results. In 1 we discuss the motivation about why this thesis has an is important and its impact in our day to day lives. In 1.1 we discuss the challenges that is faced in this field. The second chapter 2.1 is the explanation of the concepts that was used to create our models. In 2.2 is also a summary of previous work done from various research papers regarding my topic. Chapter 3 is how the project was developed,

- **Concepts Overview:** A brief explanation of all technologies used throughout the paper.
- **Literature Review:** A compilation of all previous work done in the past regarding the implemented project or any of the technologies used.
- **Methodology:** Workflow overview on how the project was implemented and how the environment developed could be replicated.
- **Results:** The section is split into to parts the first part compares between datasets tried and the second part is the results received from the project.
- **Conclusion:** A brief discussion of what was achieved in this paper, and the future work that could be done to excel the results and further development.

Chapter 2

Background

2.1 Concepts overview

2.1.1 Deep Learning

Deep learning is considered as a subfield from machine learning, it studies algorithms that are modeled after the human brain as shown in figure 2.2. These algorithms are similar to the neurons in our brains they are known as artificial neural networks. Multiple layers make up deep learning neural networks with each layer gradually extracting more complex features from the unprocessed input data.

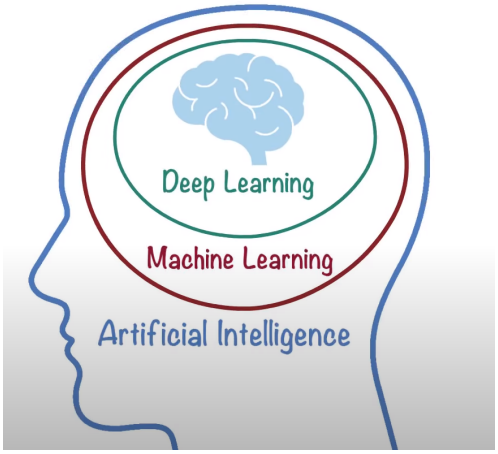

Figure 2.1: [1]Deep Learning as a Subset

Through the training of the model, the network is subjected to a vast amount of data that is labeled. These labels are used to modify the model weight in detecting. Through this iterative process the model is able to identify the patterns in the data.

2.1.2 Convolutional Neural Networks (CNN)

Particularly well-suited for computer vision tasks like face detection and recognition are CNNs a particular kind of deep learning architecture. Their proficiency lies in obtaining spatial characteristics from pictures which enables them to recognize objects and patterns with efficiency. The central component of a CNN is its convolutional layers. They apply filters also known as kernels that move over the picture to extract details such as textures edges and corners. Every filter picks up on identifying a particular kind of feature in the picture. The network can gradually learn more complex features by stacking multiple convolutional layers on top of each other starting with simpler ones. In order to decrease the dimensionality of the data and increase the computational efficiency of the network pooling layers down sample the output of convolutional layers. A feature maps most significant value is chosen from a predetermined region using pooling techniques like average pooling or max pooling. Finally fully connected layers connect output neurons to the flattened output of the convolutional and pooling layers in a manner akin to that of classical neural networks. Such tasks as classification (face detection) or recognition (person identification) fall under the purview of these last layers.

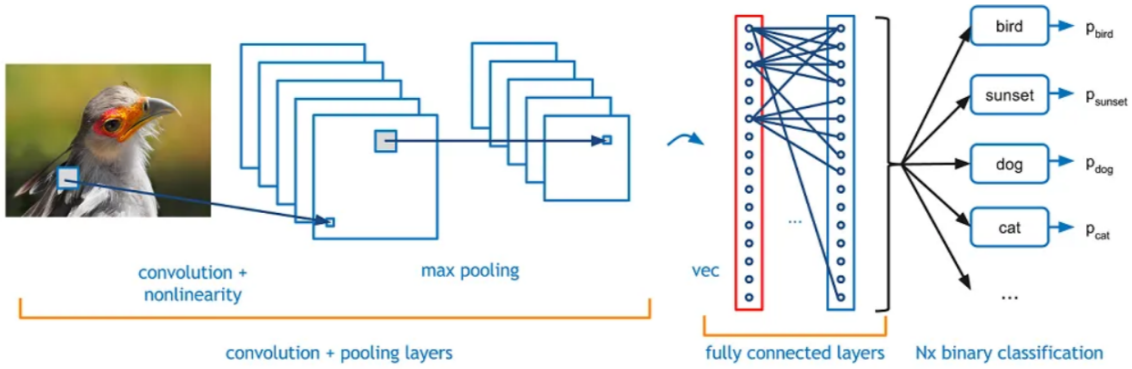

Figure 2.2: [2]Convolutional neural networks architecture

2.1.3 ResNet-50

A convolutional neural network (CNN) architecture called ResNet-50 was created expressly to overcome the difficulties involved in training extremely deep neural networks. For face detection and recognition tasks it is essential to achieve state-of-the-art performance. The vanishing gradient problem which occurs when gradients get extremely small during back-propagation prevents traditional deep neural networks from training deeper layers. To combat this residual blocks are introduced. By adding the original input and the output of a few convolutional layers these blocks enable the network to learn. By essentially creating a shortcut path this ensures that the gradients flow more effectively and allows ResNet-50 to train much deeper architectures (in this case around 50 layers). Usually it is broken down into four stages with multiple residual blocks in each. In order to extract progressively more complex features from the input image these stages increase

the number of filters (feature maps) one by one. When learning to identify facial features like eyes noses and mouths later stages may concentrate on edges and lines in the earlier stages.

2.1.4 YOLO

You Only Look Once (YOLO) is a powerful object detection algorithm due to its speed and real-time data processing capabilities. Unlike some object detection techniques that need separate stages for proposal generation and classification YOLO completes both tasks in a single neural network. Consequently it is highly computationally efficient and ideal for real-time applications. The input image is divided into an S x S grid by Yolo. It is the duty of every cell to anticipate objects that are inside its boundaries. After the initial prediction YOLO uses a method known as Non-Maxima Suppression (NMS) to eliminate unnecessary bounding boxes. Because of this only the most precise prediction will be left for every object.

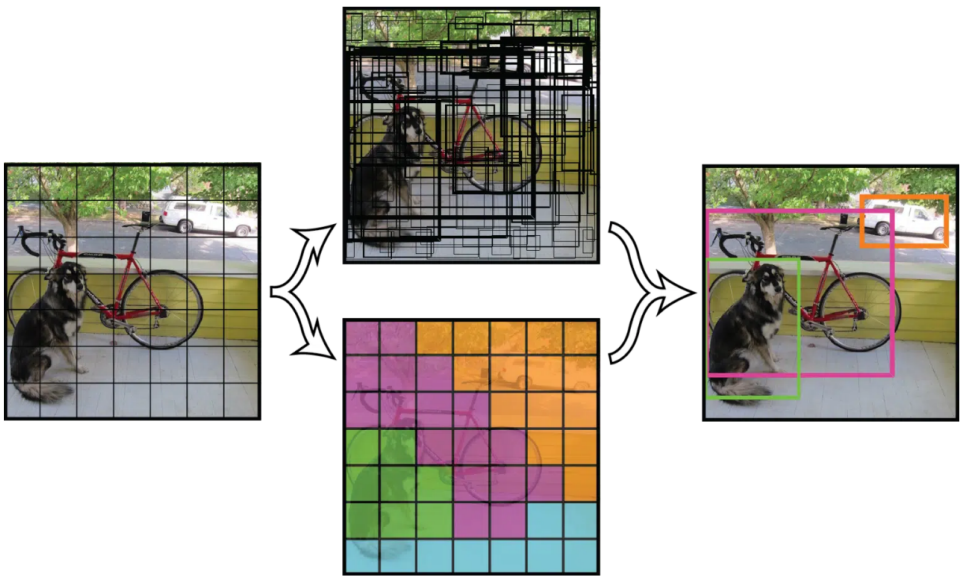

Figure 2.3: [3] YOLO example in object detection

2.2 Literature review

The whole research papers read can be categorized into three main areas: Face Detection, Face Recognition and Deep Learning Techniques.

2.2.1 Face Detection

A collection of articles explores the crucial work of precisely identifying faces in pictures and videos or face detection. An overview of the methods investigated is provided below:

Kumar et.al.[6] and Zhang et.al.[7] provide insightful information about the state of face detection research today. They give thorough summaries of current methods contrasting and comparing their advantages and disadvantages.

Maximum efficiency and accuracy are given equal priority in the Deng et al.[4] model (RetinaFace). Compared to some two-stage detection techniques it is faster because it uses a single-stage neural network architecture. In order to capture faces at different sizes within an image RetinaFace also uses a unique feature pyramid network. Having an average precision of 96.1% on WIDER FACE dataset. The TinaFace model by Zhu et al.[8] on the other hand places more emphasis on a lightweight model appropriate for devices with limited memory and processing power. Having an average precision of 96.3% on WIDER FACE dataset. Compared to RetinaFace some accuracy may be lost as a result. A Dual Shot Face Detector (DSFD) network is presented by Li et al.[9] that strikes a compromise between excellent accuracy and quickness. With a new Feature Enhancement Module (FEM) to produce richer feature representations and a Progressive Anchor Loss (PAL) to enhance learning across different face scales DSFD employs a two-stage methodology. Furthermore the appropriateness of DSFD for real-time face detection tasks is guaranteed by its lightweight components and optimized network architecture. Having an average precision of 95.0% on WIDER FACE dataset.

RetinaFace utilizes a pyramid feature, which is a multi-level pyramid. First features are extracted from an image and are then fed into the model. Encoding various details at different scales. Larger faces are recognized at the lower level of the pyramid, while smaller faces are recognized at higher levels of the pyramid as shown in figure 2.4.

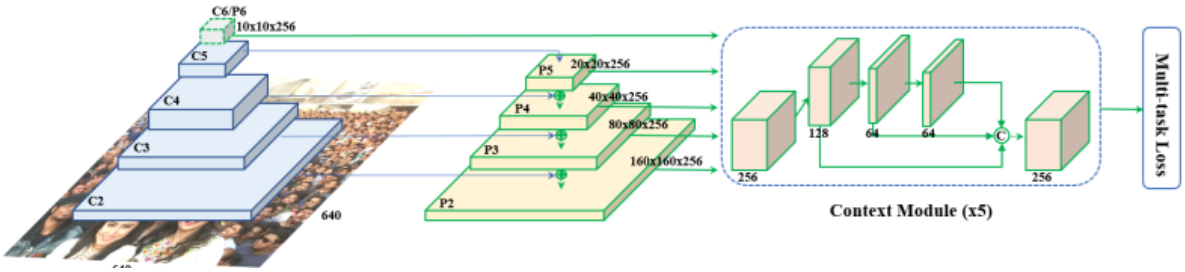

Figure 2.4: [4]RetinaFace pyramid feature

A multi-stage face detection framework called MTCNN is presented by Zhang et al.[10]. To attain great accuracy MTCNN makes use of three convolutional neural networks (CNNs) each of increasing complexity. Candidate bounding boxes are suggested in the first stage refined in the second and facial landmarks such as the mouth nose and eye key points predicted in the third. However FaceBoxes is a single-stage face detection algorithm that is optimized for CPU performance in real-time and is used by Zhang et al.[11] In order to achieve good accuracy and efficiency appropriate for CPU processing FaceBoxes uses a convolutional neural network architecture that has been specially designed for this purpose.

The Chi et al.[12] SRN model uses a two-pronged strategy to address false positives and enhance location accuracy in face detection. Selective Two-step Regression is used to fine-tune possible faces for accurate localization after Selective Two-step Classification removes the majority of negative detections early on. A Receptive Field Enhancement block designed to capture details in difficult poses further improves network performance. With this combination SRN is able to achieve cutting-edge face detection performance. Utilizing the well-known YOLO object detection framework Qi et al.[13] YOLO5Face modifies it especially for face detection. A face detector architecture that addresses the shortcomings of current techniques is proposed by Liu et al.[14] Three main challenges are addressed: removing false alarms enhancing scale-level data and assigning labels. In order to address these problems MogFace has added three new modules: the Hierarchical Context-Aware Module which lowers false positives the Adaptive Online Incremental Anchor Mining Strategy which improves label assignment and the Selective Scale Enhancement Strategy. Through the resolution of these issues MogFace attains cutting-edge results on multiple face detection benchmarks. By using contextual information Tang et al.[15] address the problem of detecting challenging faces (small blurry occluded). To make use of context it employs Pyramid Anchors in the training process a Context-sensitive Prediction Module ensures precise face location and classification and a Low-level Feature Pyramid Network (LFPN) integrates contextual and facial features. This method enables PyramidBox to recognize faces in difficult real-world situations.

Wang et.al.[16] and Kim et.al.[17] explore the application of deep learning architectures, particularly Fully Convolutional Networks (FCNs), for face detection. FCNs excel at learning spatial features from images, making them well-suited for tasks like identifying and locating faces within an image.

Newer papers in face detection research focuses on the development of models specifically designed for devices with limited resources which was the main focus on Xu et.al.[18] CenterFace model. This is particularly relevant for applications on mobile phones or embedded systems, where computational power and memory usage are critical factors.

2.2.2 Face Recognition

This subsection dives into the realm of face recognition, which focuses on identifying individuals from facial images or videos. Here, the papers explore various algorithms and techniques to achieve this task.

By providing thorough surveys of current face recognition systems Kortli et.al.[19] and Parkhi et.al.[20] offer fundamental knowledge. These questionnaires explore various strategies detailing their advantages and disadvantages. You will gain a thorough understanding of the state of the art in face recognition research with this comparative analysis.

GhostFaceNets an effective face recognition model architecture is examined by Alansari et.al.[21]. In order to achieve good recognition accuracy without requiring a lot of computational power this model places a high priority on lightweight design and uses inexpensive

operations like depthwise separable convolutions. using the ResNet-50 backbone model which is described in 2.1.3. GhostFaceNets are therefore appropriate for implementation on systems with constrained processing capacity. Having an accuracy of 99.8% on LFW dataset. In addition George et.al.[22] suggest the EdgeFace face recognition network architecture which is effective and lightweight like GhostFaceNets[21]. It does this by fusing the advantages of Transformer models and Convolutional Neural Networks (CNNs) through a hybrid network architecture that draws inspiration from EdgeNeXt. To further reduce computation in the linear layers without noticeably sacrificing performance a Low Rank Linear (LoRaLin) module is also introduced. Because of this combination EdgeFace can achieve high face recognition accuracy with low computational costs and compact storage which makes it appropriate for deployment on edge devices with limited resources. Having an accuracy of 99.8% on LFW dataset. A Dynamic Class Queue (DCQ) mechanism is also proposed by Li et.al.[23] to address imbalanced class distributions and computational limitations similar to GhostFaceNets[21] and EdgeFace[22]. During each training iteration DCQ reduces computational burden by dynamically selecting a subset of classes for recognition. Furthermore the model is able to handle tail classes with limited training examples because class weights are generated dynamically on the fly. DCQ is able to attain competitive results on extensive facial recognition datasets by using this method despite having restricted computational resources. A model that addresses the performance problems on smaller devices is developed in all three of these papers.

A collection of papers explores loss functions which are an important component in neural network training for face recognition. The goal of these loss functions is to maximize the networks capacity to distinguish between people according to their facial characteristics. Models trained on one data collection method may perform poorly on different evaluation methods this is known as process discrepancy and Kim et.al.[24] DiscFace model addresses this issue. DiscFace improves generalizability by making the model learn features that hold true in various scenarios. The quality-aware mechanism in QMagFace by Terhorst et.al.[25] is integrated into the recognition process. It generates a more robust performance in difficult conditions by estimating the quality of an image and adjusting the recognition confidence score accordingly. For the purpose of training representations of distinct identities existing methods frequently employ a fixed margin penalty. The ElasticFace model by Boutros et.al.[26] suggests a more adaptable strategy. During each training iteration a random margin value is employed which enables the model to acquire a decision boundary that can adjust to fluctuations in the data and could result in enhanced recognition precision. ElasticFace[26] and Deng et.al.[27] ArcFace share the goal of enhancing face recognition models discriminating power. In order to maintain high classification accuracy it uses a particular loss function that promotes a wider margin between the representations of various identities.

Schroff et.al.[28] presented a method for face recognition, it builds a single system rather than learning different models for every task. In a compact space distances signify facial similarity and FaceNet [28] maps face images to this space directly. This makes it possible to carry out common tasks within this new space such as recognition and clustering. This approach is effective because it obtains high accuracy with only a small amount of data (128 bytes) per face.

The complexity of real-world scenarios is addressed by recent research efforts. The problem of face recognition with masks for example is addressed by Mare et.al.[29] and is becoming more and more problematic as a result of the COVID-19 outbreak. The limitations of occluded facial regions for precise identification are examined in this paper. Furthermore Knoche et.al.[30] concentrates on enhancing face recognition models resistance to image changes. Real-world scenarios may involve capturing faces in varying lighting conditions poses or resolutions. This paper investigates ways to improve the models performance in spite of these changes.

An approach to the problems of face recognition in unrestricted environments is put forth by Liu et.al.[31]. Creating a controllable face synthesis model (CFSM) that can replicate the distribution of target datasets in various settings. The model picks up on a latent space that represents the target datasets stylistic variances. This makes it possible to precisely control the creation of artificial faces with particular traits. The study shows that incorporating this artificial data into a face recognition models training process greatly enhances performance in a range of environmental settings. Differentiating it from every other approach that has been discussed.

2.2.3 Deep Learning Techniques

This subsection examines deep learning approaches which form the basis of many face detection and recognition algorithms. Although the papers in the earlier sections concentrated on particular applications, in this section we explore deeper deep learning methodologies that could be useful for these kinds of tasks.

Deep Polynomial Neural Networks (DPNNs) are investigated by Chrysos et.al.[32] as a substitute architecture for deep learning applications. In face detection and recognition convolutional neural networks (CNNs) are the industry standard however DPNNs provide an alternative method. It can be insightful and even lead to new solutions in the future to understand alternative architectures.

Siamese Neural Networks are a kind of architecture that are skilled at recognizing similarities between data points. Koch et.al.[33] explores this technology. While learning representations for distinct identities is often required for face recognition, Siamese Networks are particularly good at comparing and analyzing image pairs and seeing if they are somewhat similar.

Xu et.al.[34] and Guo et.al.[35] discuss issues that arise frequently when implementing deep learning models in practical settings. The idea of domain adaptation is examined in [34], a method that enables models trained on one dataset to function well on another that is similar to it. Guo et.al.[35] focus on the effectiveness of resource distribution in deep learning models. This is essential when using face detection and recognition software on devices with constrained memory and processing power.

2.3 Datasets

WIDER FACE[36] is considered to be the most commonly used dataset for face detection, due to the representation of faces encountered that is similar to real-life scenarios. Providing varying poses, light conditions, faces sizes and occlusions. For face recognition many datasets are widely used for both training and evaluation of a model. Labelled faces in the wild[37] (LFW) is the most commonly used dataset as a benchmark dataset for evaluation of the model. However LFW limitation is the diversity for other ethnicities in the dataset.

MegaFace(MF2)[38], VGGFace2[39] and MS Celeb-1M[40] are all used for both training and evaluating face recognition models. Compared to LFW, MF2 and MS Celeb-1M datasets handle LFW limitation in diversity by having a vast dataset containing various ethnicities and age groups. While VGGFace2 on high resolution celebrity images for more detailed facial landmarks detection. Using these datasets for training rather than LFW gives the model generalizablity for real-life scenarios.

Name	Total images	Total faces	Split(train,val,test)
WIDER FACE	32,203	393,703	40%,10%,50%
LFW	13,200	5,749	used only for evaluation
MegaFace(MF2)	4.7 million	672,000	80%,10%,10%
VGGFace2	3.31 million	over 9,000	80%,10%,10%
MS Celeb-1M	10 million	100,000	80%,10%,10%

Table 2.1: Datasets used throughout literature review.

2.4 Evaluation metrics

When it comes to evaluating a model on how well it preforms there are multiple metrics that can used, such as:

Accuracy

Accuracy[41] is an evaluation metric that determines whats the percentage of the models predictions are correct. It gives an idea of how well a model is able to detected true positives and negatives.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Precision

Precision[41] evaluates how precise the model is at predicting positive labels. It calculates out of the total number of times the model predict positive, how many was correct.

$$Precision = TP/(TP + FP)$$

Recall

Recall[41] measures the percentage of actual positives a model correctly detect.

$$Recall = TP/(TP + FN)$$

$$TP = true\ positive,\ TN = true\ negative,\ FP = false\ positive,\ FN = false\ negative$$

2.5 Discussion

Based on the literature review above in 2.2, there is a gap in the research where they do not experiment on real-time video surveillance systems for both face detection and recognition. By incorporating real-time processing on video surveillance it can be turned into a proactive tool. Enabling immediate detection, could help in identifying the restricted individuals from entering a specified area or even identifying a person from a crowd, who is blacklisted from the environment.

Additionally, implementing an AGR (age, gender and race) detection could be very beneficial for video surveillance systems. Real-time processing of AGR data would allow us to analyze each individual entering the facility, providing a live count of the people inside the facility with how many are males and females and also their age groups and race. Having this data is not only beneficial for keeping track of the number of people in the facility but also for use as a statistical analysis.

Chapter 3

Methodology

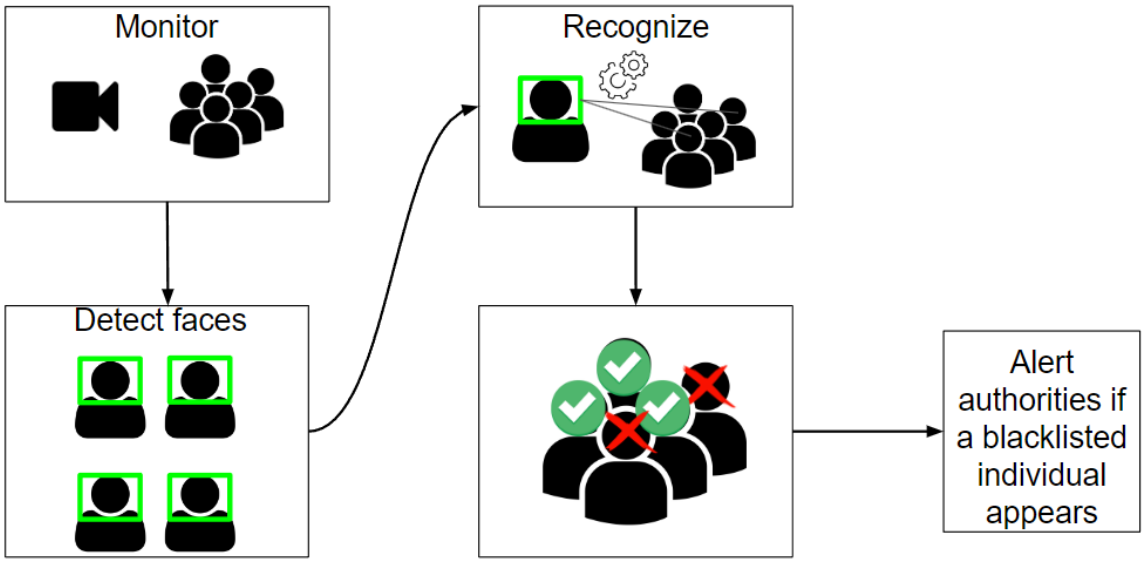

Figure 3.1: A visualization of how the model works

Figure 3.1 shows an overview of how our model would work. Where we would have a video surveillance camera monitoring a specific facility, always detecting and identifying individuals passing by it. By detecting faces and having a continuous count of people inside the facility and also being able to identify any known blacklisted individuals from the premises, an alert would be sent notifying the authorities of their presence and taking immediate action.

3.1 Image augmentation and annotations

The caliber and volume of training data are critical components in deep learning models performance. Building a strong dataset becomes essential in the context of multi-view face detection where capturing faces from various viewpoints is vital. Our two-pronged approach uses Labelme for image annotation and Albumentation for image augmentation to accomplish this.

3.1.1 Albumentation tool for image augmentation

A robust Python library called Albumentation was created especially for image augmentation. The process of artificially producing variations of preexisting images is known as image augmentation. The deep learning models benefit greatly from this process because it teaches them to avoid overfitting on the training set and to generalize well. We can create artificial versions of our original face images in this case by using albumentation which covers a greater range of rotations scales lighting and occlusions. Albumentation for instance can be used to modify contrast and brightness to replicate various lighting conditions or to slightly rotate faces to simulate various head positions. These changes expose the model to a wider range of face appearances improving its capacity to recognize faces accurately from different angles in real-world situations.

Figure 3.2: Albumentation tool

3.1.2 Labelme annotation tool

In contrast Labelme is essential to the process of annotating images. It is an open-source free graphical annotation tool that makes labeling images with bounding boxes easier. Labelme in this case enables us to precisely annotate the bounding boxes surrounding every face in the picture regardless of its angle. The ground truth for our model is provided by these annotation data. The augmented image and the bounding box annotations for it are both shown to the model throughout the training process. The model then gains the ability to correlate the features that were extracted from the picture with the existence and positioning of faces. Training our multi-view face detection model requires high-quality labeled datasets which Labelmes user-friendly interface and effective annotation capabilities make possible.

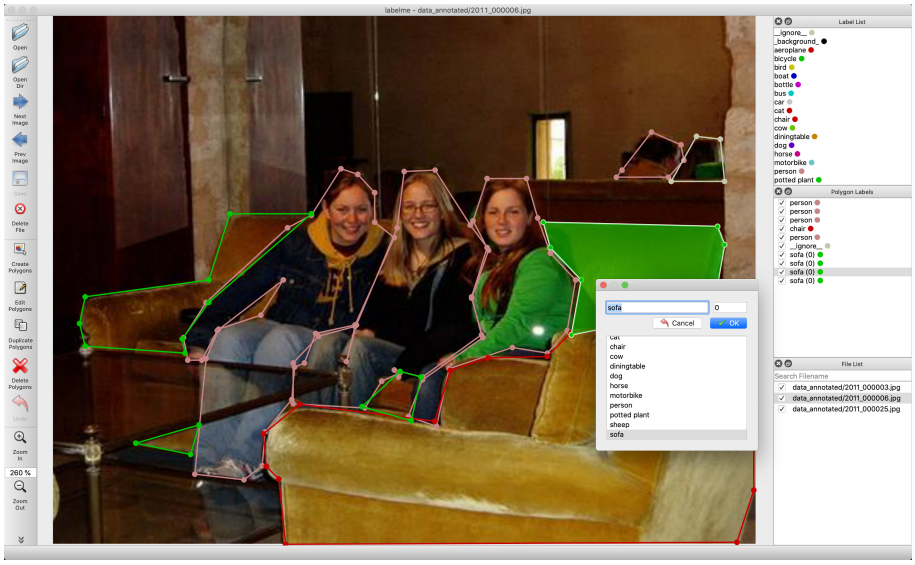

Figure 3.3: Labelme tool

We can create a large and varied training dataset that better prepares our model to handle the challenges of multi-view face detection by combining the strength of Labelme for image annotation and Albumentation for image augmentation. A greater variety of face variations are presented to the model through the augmented images and the carefully labeled annotations supply the ground truth required for efficient training. The development of a reliable and precise multi-view face detection system is made possible by this combined method.

3.2 Face detection

We examined three face detection techniques and contrasted their respective performances. YOLOv9, a trained ResNet model and the face detection model are well-known

models. contained in the library of face recognition. Analyzing each models robustness accuracy and efficiency in identifying faces across a range of image datasets was our goal. We sought to determine which model provided the best face detection abilities by examining the data.

3.2.1 YOLO based face detection

We leverage You Only Look Once version 9 (YOLOv9)s innovative features for our real-time multi-view facial detection. Our proposed system is one of the many real-world applications that benefit greatly from this state-of-the-art deep learning object detection models superb balance between accuracy and computational efficiency. That develops. We rely on YOLOv9 and its innovative features for our real-time multi-view face detection. Our suggested system is an excellent option for real-world applications that require real-time performance as this state-of-the-art deep learning object detection model remarkably balances accuracy and computational efficiency. A single integrated neural network is used by YOLOv9 to predict bounding boxes and class probabilities for objects in an image simultaneously in contrast to conventional two-stage detectors that need separate stages for proposal generation and bounding box regression. Achieving real-time processing requires a significant speed advantage which this one-stage detection approach offers.

We go deeper into YOLOv9s advantages and strategically use them to address the problem of multi-view faces. First we use a carefully selected training dataset to tackle the problem of pose variation. This dataset includes a wide range of facial poses including profile views frontal views tilted angles and even partially obscured faces. The YOLOv9 model can learn strong and generalizable features that enable it to correctly identify faces in images regardless of their orientation thanks to the extreme diversity of the data. Since faces are rarely displayed in a perfectly frontal fashion in real-world situations this is especially crucial.

Second we leverage the multi-scale prediction capabilities and anchor boxes of YOLOv9. Anchor boxes are pre-made boxes positioned thoughtfully throughout the image in different sizes and aspect ratios. In order to produce tight-fitting bounding boxes around the identified faces the model refines these anchor boxes based on the features extracted from the extracted image during the prediction stage. Moreover YOLOv9s architecture includes multi-scale prediction. As a result the model can predict at several scales at once and recognize faces of different sizes in the image with accuracy. This is especially useful for multi-view face detection where a faces apparent size can change dramatically based on how it is positioned in relation to the camera. Strengths such as the adaptability of anchor boxes the generalizability from a wide range of training sets and the efficiency of single-stage detection are leveraged by the multi-scale prediction based approach to achieve robust and real-time face detection across a spectrum of facial poses. The subsequent face recognition tasks in our system rely heavily on this reliable detection as their fundamental building block.

Figure 3.4: YOLO example in face detection

3.2.2 ResNet based face detection

Utilizing a ResNet model that has already been trained is key. One deep learning architecture known for its efficacy in image recognition tasks is called ResNet or Residual Neural Network. Its remaining learning blocks are its main advantage. By adding the input straight to the convolutional layers output these blocks allow the network to learn complex features avoiding the vanishing gradient problem a common training challenge for deep networks. This method makes it easier to learn feature hierarchies from images in an efficient manner.

We use a pre-trained ResNet for multi-view face detection. The purpose of this pre-training is to teach the ResNet generic image features such as edges textures and basic shapes using a large dataset of labeled images. These qualities provide a solid basis for the tasks that follow. This is where pre-training becomes useful. Because of the large dataset it was trained on the pre-trained ResNet already possesses strong feature extraction capabilities. Main stage goes to the pre-trained ResNet. The input picture is fed into the network possibly with multiple faces in different orientations. Features are gradually extracted from the image as it passes through the convolutional layers capturing progressively more intricate details. These characteristics serve as the fundamental components of face detection.

After being extracted the features are fed into more layers that are made expressly for face detection. In these layers classifiers that have been trained to detect faces in an image regardless of their orientation may be used. Bounding boxes enclosing the faces found in the image would be the stages output. Our advantage is substantial because we use a pre-trained ResNet. Our system is able to manage the complexity of multi-view face detection thanks to the networks pre-learned feature extraction capabilities. With the help of the pre-trained network faces in different orientations can be recognized and key features that are useful for face recognition and other later tasks can be extracted.

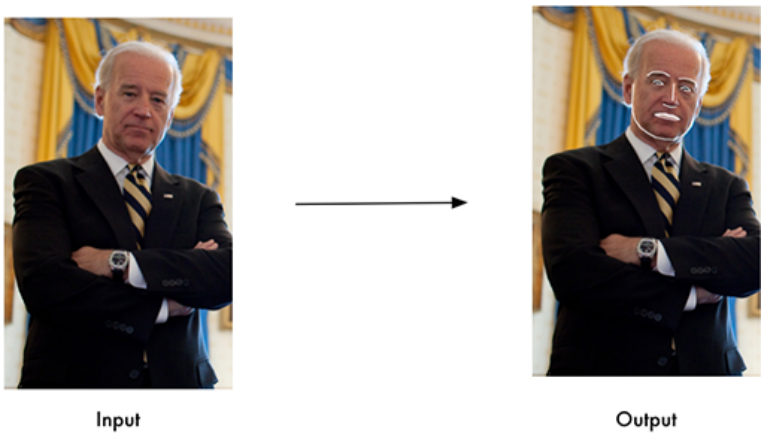

Figure 3.5: Feature extraction detection example

3.2.3 RetinaFace based face detection

RetinaFace[4] is a deep learning face detection model that is able to find faces in a frame with accuracy and speed, due to its focus on being lightweight model. Unlike most face detection models discussed in 2.2 that utilizes a multi-layered face detection, RetinaFace does this in one shot.

ResNet forms the backbone of the model. At every level of the pyramid it effectively creates these feature maps based on the image. Deformable convolutional networks (DCN) are a technique that is used to further improve the models comprehension of the context within the image. With the help of DCN the model can concentrate on particular regions within the features giving regions that probably contain faces more attention.

RetinaFace applies a cascade regression strategy after feature extraction and enrichment. This entails a number of adjustments to determine the precise position and dimensions (bounding box) of the face in the picture. Furthermore the model predicts the positions of important facial landmarks like the mouth nose and eyes in addition to bounding boxes. Applications for image editing and face recognition benefit greatly from this capability. RetinaFace uses a multi-task loss function to guarantee that these detections are accurate. This function allows the model to simultaneously improve its performance on both landmark localization and bounding box prediction by combining their errors.

After reviewing the different models for face detection, I came to a conclusion to use the RetinaFace model. That conclusion is due to the results of the different models above. The ResNet model could not detect faces unless image had high resolution, whole face is visible and face is close to the camera not faraway. The YOLO model was far better than the ResNet, however various faces were not detected due to the limitation mentioned in 4.3.1. Thus coming to the conclusion of using the RetinaFace model as the face detection model, as the model did not face the same issues as YOLO and ResNet.

3.3 Face Recognition

Various data must be processed effectively for the model to work as there is a growing need for real-time facial recognition applications. This covers jobs like recognizing faces in a live video stream and creating encodings for recognized faces.

3.3.1 Pre-processing

For accurate and timely results real-time facial recognition systems strongly depend on effective data pre-processing techniques. This section explores color space conversion and face encoding two critical pre-processing steps in the face recognition system discussed in this chapter.

Color Space Conversion: From BGR to RGB

OpenCV uses the BGR (Blue Green Red) color channel order by default to represent images in the video frames that are initially captured from the camera. Nonetheless a lot of deep learning frameworks and libraries used for facial recognition tasks use RGB (Red Green Blue) color formatting as does the face recognition library in this system. This seemingly insignificant detail requires a conversion step at the pipelines start for preprocessing.

Rearranging the color channels within the image data is necessary to convert it from BGR to RGB. The RGB format of the image representation corresponds to the expectations of the deep learning models used in the library for face recognition even though the underlying image information is still the same. Incorporating the image data into the following processing stages is made easier and compatibility is guaranteed.

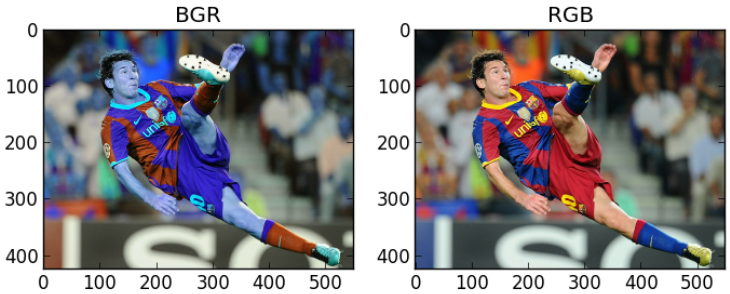

Figure 3.6: Conversion from BGR to RGB

Face Encoding: Capturing Facial Features for Recognition

Creating face encodings is a crucial step in the pre-processing stage of facial recognition. The fundamental features of a persons face can be represented numerically by a face

encoding. This compact representation acts as a faces unique identifier within the system it is usually a vector of numbers. A key component of our systems face encoding process is the face-recognition library which uses deep learning models that have already been trained for face recognition.

Large-scale labeled facial image datasets are used to train these deep learning models. The model gains the ability to recognize and extract a set of important features that set one face apart from another during the training phase. These characteristics can be the separation between the eyes the contour of the jawline the prominence of the cheekbones or any other special traits. Averaging 99. 38 percent precision on Labeled faces in the wild dataset (LFW) was also achieved. A deep learning model can be used to create face encodings for previously undiscovered faces once it has been trained. This is accomplished by using the built in face encoding function.

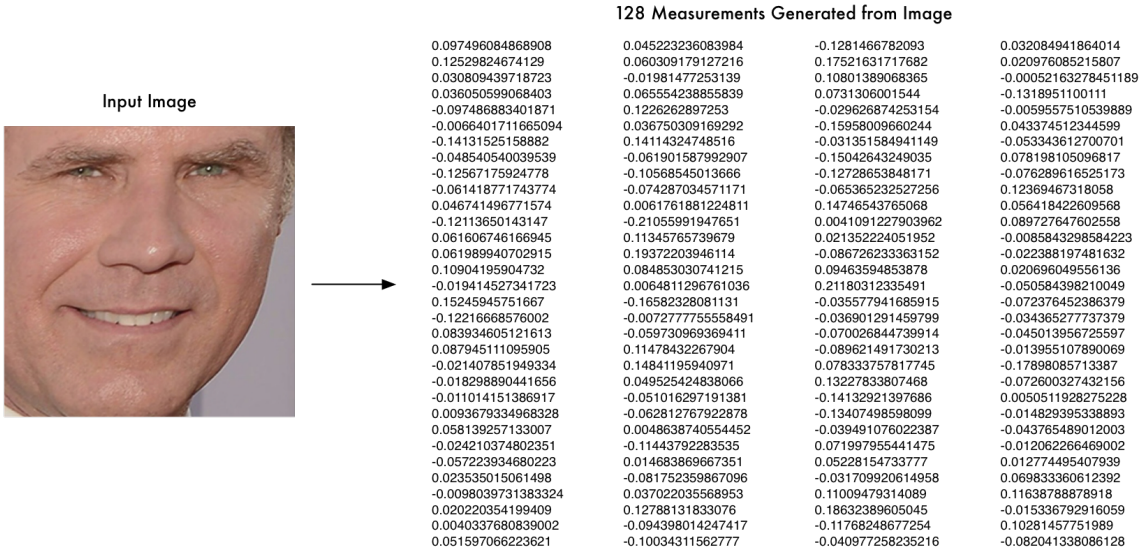

Figure 3.7: Face encoding example on input image

It recognizes faces in images and extracts facial features using the pre-trained deep learning model in the library. After that a condensed numerical representation of the extracted features is created by encoding them in the face. The generated face encoding for a given person remains largely consistent across images of that same person as long as their lighting and facial expressions are reasonably similar. The feature mentioned above allows the system to compare the face encodings of people who are known to be in the database with the face encodings of faces that are detected in the video stream. By using methods to compute the similarity between encodings the system determines whether the detected face corresponds to a known individual.

The recognition process and the raw picture data are essentially connected by face encoding. By reducing the complex information present in an image to a more manageable and comparable format it enables the system to perform real-time face recognition effectively.

3.3.2 Real-time processing

Real-time processing is where the magic of facial recognition happens. The main processes for converting static photos of well-known faces into a system that can recognize people in a live video stream are broken down in this section. We will investigate face detection name recognition and video capture combined to provide real-time facial recognition.

Getting a steady stream of video frames from a camera is the foundation of real-time processing. Our code makes use of cv2 in OpenCV in order to access the desired camera. Serving as a bridge this function connects to the camera to allow frames to be retrieved at a predetermined rate. A snapshot of the scene taken by the camera at a specific moment is represented by each retrieved frame. The live video stream that the system is analyzing for faces is made up of this series of continuously recorded and processed frames.

Our system extracts a face encoding using the face recognition function for each detected face location obtained from the face detection model 3.2 (bounding box). This function makes use of the pre-trained deep learning model in the face recognition library much like the method used during pre-processing for known faces. The bounding box encloses a portion of the frame that is analyzed by the model this region most likely contains a face. Then it extracts a condensed numerical representation that captures the key facial features of the person that was detected—the face encoding.

The comparison of detected faces face encodings with the encodings of known people kept in the systems database forms the basis of real-time face recognition. By repeatedly going through each extracted face encoding our code achieves this. Every encoding is compared against all known face encodings kept in the face-encodings list using the compare faces function. Whether a match is found between the current encoding and any of the known encodings is indicated by the list of boolean values that this function returns.

The algorithm gives the detected faces names after identifying possible matches. The strategy used by our code gives the highest priority to locating the recognized face that is the closest to the current face encoding or the most similar encoding. This is accomplished by finding the index of the closest match in the list of known encodings and computing the face distances. The recognized name for the detected face is then assigned to the corresponding name from the list at that index. It will be labeled unknown if no match is discovered.

At last the user is shown the processed frame which may contain faces the user has identified with names above them. Our code makes use of OpenCVs feature to show the frame on the screen. Through the use of bounding boxes drawn around detected faces and names displayed for those who are recognized the user is able to view the live video stream. The system produces a smooth real-time facial recognition experience by continuously taking pictures processing them and displaying them. Through a series of coordinated actions real-time processing essentially creates a dynamic system that can recognize faces in a live video stream from a static database of recognized faces. Real-time face recognition is made possible by an intricate interplay of functions from capturing video frames to extracting face encodings and comparing them against a database.

3.4 DeepFace AGR based detection

DeepFace is a Python library designed for face analysis tasks, developed by the OpenAI research team. It leverages deep learning techniques and pre-trained neural networks to perform facial analysis, including age estimation, gender prediction, emotion recognition, and facial recognition.The first step involves loading an image into the program for analysis. The input image is read using the OpenCV library, which is a popular computer vision library in Python.The analyze function is the key component responsible for performing facial analysis on the provided image. Such as age, gender and race (AGR).

Figure 3.8: [5]DeepFace AGR detection example

There are certain limitations associated with deepface that must be acknowledged. The accuracy of age, gender and race prediction heavily depends on the quality of the input image and the performance of the pre-trained models used by DeepFace. The success of the analysis is dependant upon the presence of detectable faces in the image. The pre-trained models may be biased towards certain age or gender distributions in the training data, potentially leading to inaccuracies in some cases.

Finally, for the final draft of our model, there are 3 stages that each frame pass through. First the frame is passed through the RetinaFace for identifying where each face is and as discussed here **??** is why we used this model over the others. After identifying and saving the coordinates of all the known face locations we loop over each face, where each loop passes a cropped image of the face through the deepface model served as a preliminary step for feature extraction for the final stage, then we save the individuals characteristics (age, gender and race). In the end the face goes to the final stage for recognizing any face from our dataset. Only if a face is recognized do we show is the frame his/her name, otherwise we do not write unknown to avoid over crowding of information in a face.

Chapter 4

Results

4.1 Development and Testing Environment

To execute and run our code we used Google Colab a free Jupyter notebook environment. Python was our main programming language due to it being a high level language that offers various libraries and frameworks. Some of these libraries and frameworks are the following:

Pytroch: an open-source deep learning framework developed with Python highly regarded for its dynamic computation graphs and approachable coding style. Because of this developing and experimenting with deep learning models is made simple for researchers and developers. PyTorch is a potent tool for tasks like image recognition natural language processing and creating brand-new data because of its capacity to use GPUs for accelerated training.

OpenCV: a useful open-source library used for real-time computer vision tasks that excel in image processing and manipulation.

Matplotlib: a useful for tool that aids in creating static visualisation of data. Creating graphs and customizable plots.

Numpy: A core library for Python scientific computing. Data science and other scientific domains rely heavily on it because it provides an extensive range of tools for handling numerical data.

4.2 Dataset

For my face detection model I used WIDER FACE[36] dataset for both training and evaluating the model. In addition, LFW[37] dataset was utilized for face recognition model used for only evaluating the model. Furthermore, I incorporated a custom dataset consisting of images of me and other figures with masks on to further increase people wearing

masks detection. This dataset was augmented using albumentation tool (3.1.1) and annotated using labelme tool (3.1.2). This custom dataset enhances models performance on recognizing me in different lighting conditions and occlusions thanks to albumentation tool for image augmentation to simulate my face in various conditions.

To test our model performance of real video surveillance, three videos varying in different conditions. First video surveillance in eye level footage consisting of 460 frames showing the facial features clearly of all the faces. Second video surveillance is positioned as a birds eye view in the morning and consists of 6832 frames. Last surveillance video is same as second video, however at night and consisting of 10871 frames.

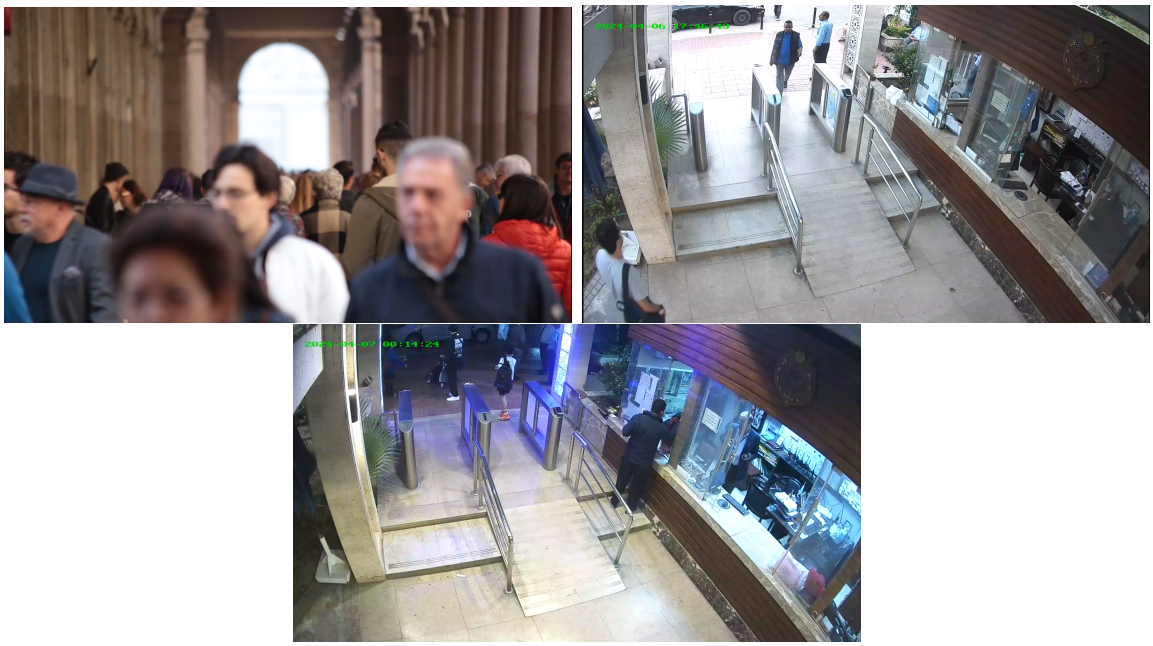

Figure 4.1: Video surveillance videos

4.3 Face detection

4.3.1 YOLOV9 based face detection

Using the new version of you only look once (YOLO) released in February 2024 version 9, I trained the YOLOV9 model on both WIDER FACE dataset and my own custom dataset together.

X-axis here represents the number of epochs used to train the model (30 epochs).

Y-axis here represents the magnitude of the loss or metric being measured.

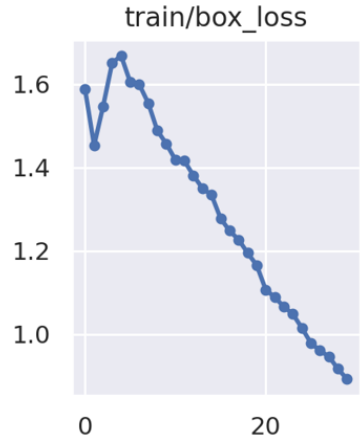

Figure 4.2: Box loss graph during training

The bounding box loss measures how far off the model's predicted bounding boxes are from the ground truth. Lower bounding box loss indicates that the model is better at predicting accurate bounding boxes for faces.

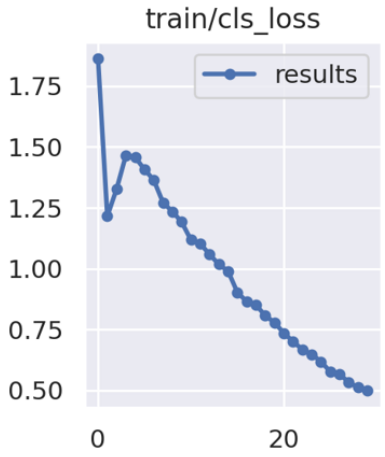

Figure 4.3: Classification loss graph during training

Here classification loss is measured to see how good the model is at, whether the bounding box created contains a face or not. Lower classification loss means the model is better at distinguishing faces from backgrounds.

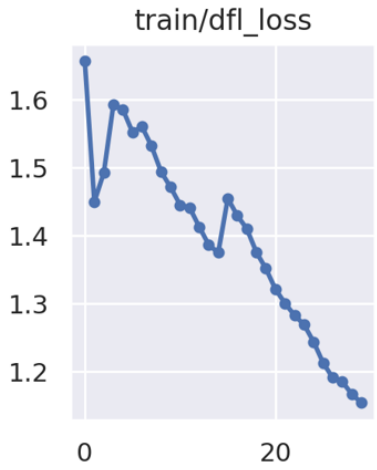

Figure 4.4: Degrees of freedom loss graph during training

This loss term is particular to YOLO. It stands for "Degrees of Freedom Loss" (dfl-loss) and corresponds to the number of anchor boxes in the model. It penalises the model for making bounding box predictions with low confidence levels. Lower dfl generally means the model is better at making anchor boxes on faces.

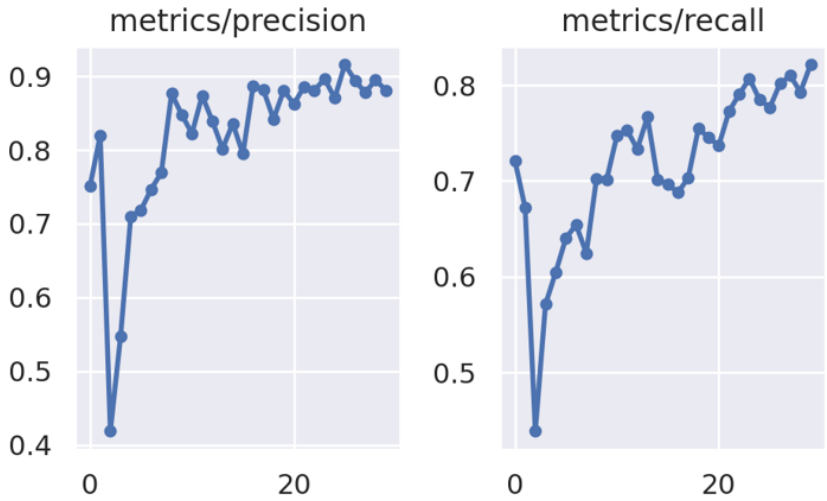

Figure 4.5: Precision and recall graphs during training

Measuring both precision and recall is considered essential in evaluating a model using the formula from 2.4. Having both high precision and recall indicated the high accuracy of the model.

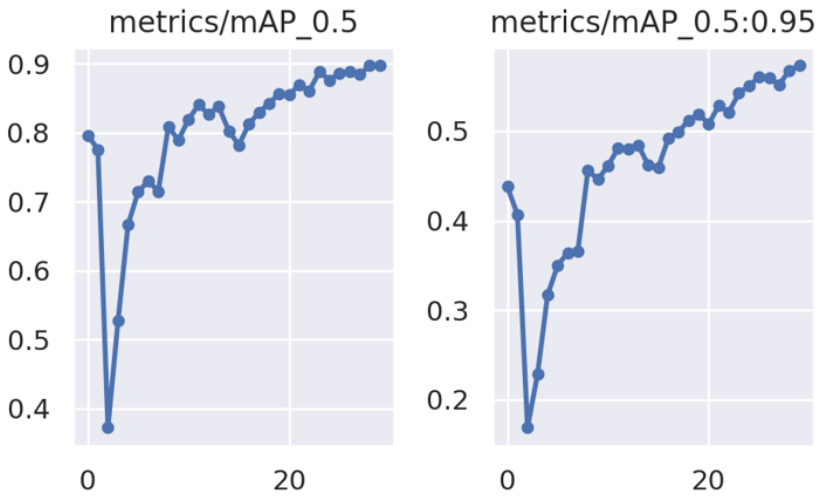

Figure 4.6: Average precision graph during training across different thresholds

Mean average precision (mAP) summarizes the precision-recall curves across various IOU (intersections over union) thresholds. IOU is how well the predicted bounding box matches the true bounding box.

Figure 4.7: Sample batch of testing images

This is the result on a sample of the testing images showing the detection for both masked and unmasked individuals.

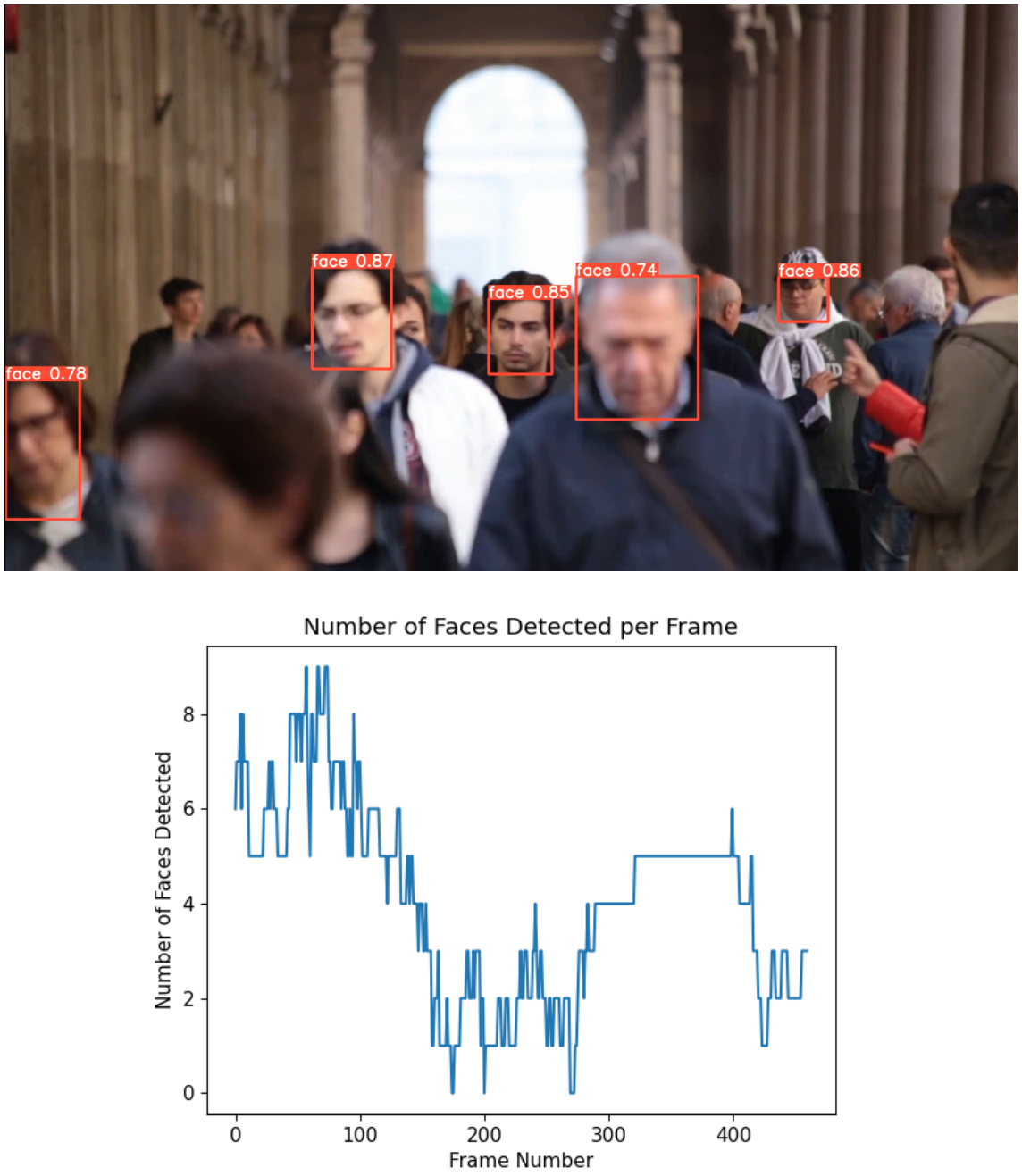

Figure 4.8: YOLOV9 test on video surveillance with a confidence of 0.4

In the Figures 4.8 the results show a significant amount of faces detected, however also missing few faces due to occlusions or face is very small. Giving the model an average precision of 75% in this video 4.2.

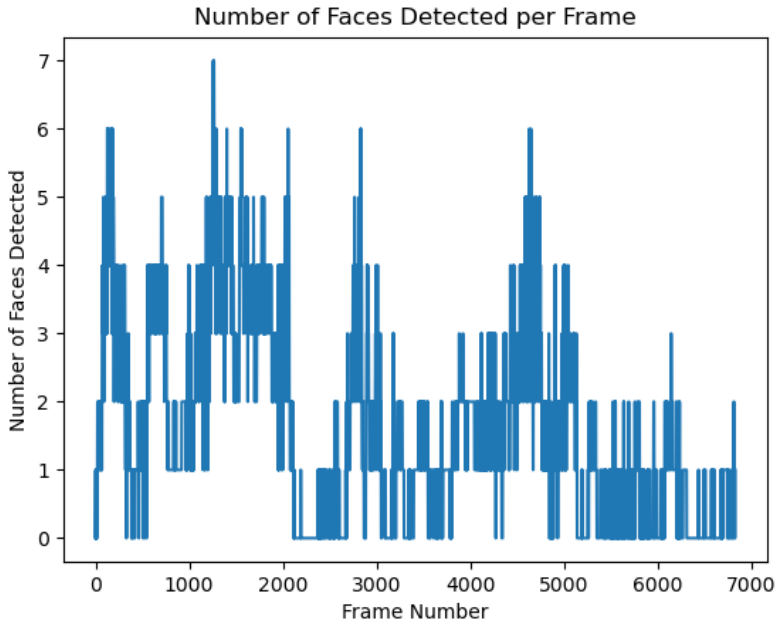

Figure 4.9: YOLOV9 test on video surveillance using confidence of 0.4

This is the result of the trained model on a video surveillance. The model were able to detect multiple faces same as before in first video, however some faces were not able to be detected giving an average precision of 62.8%.

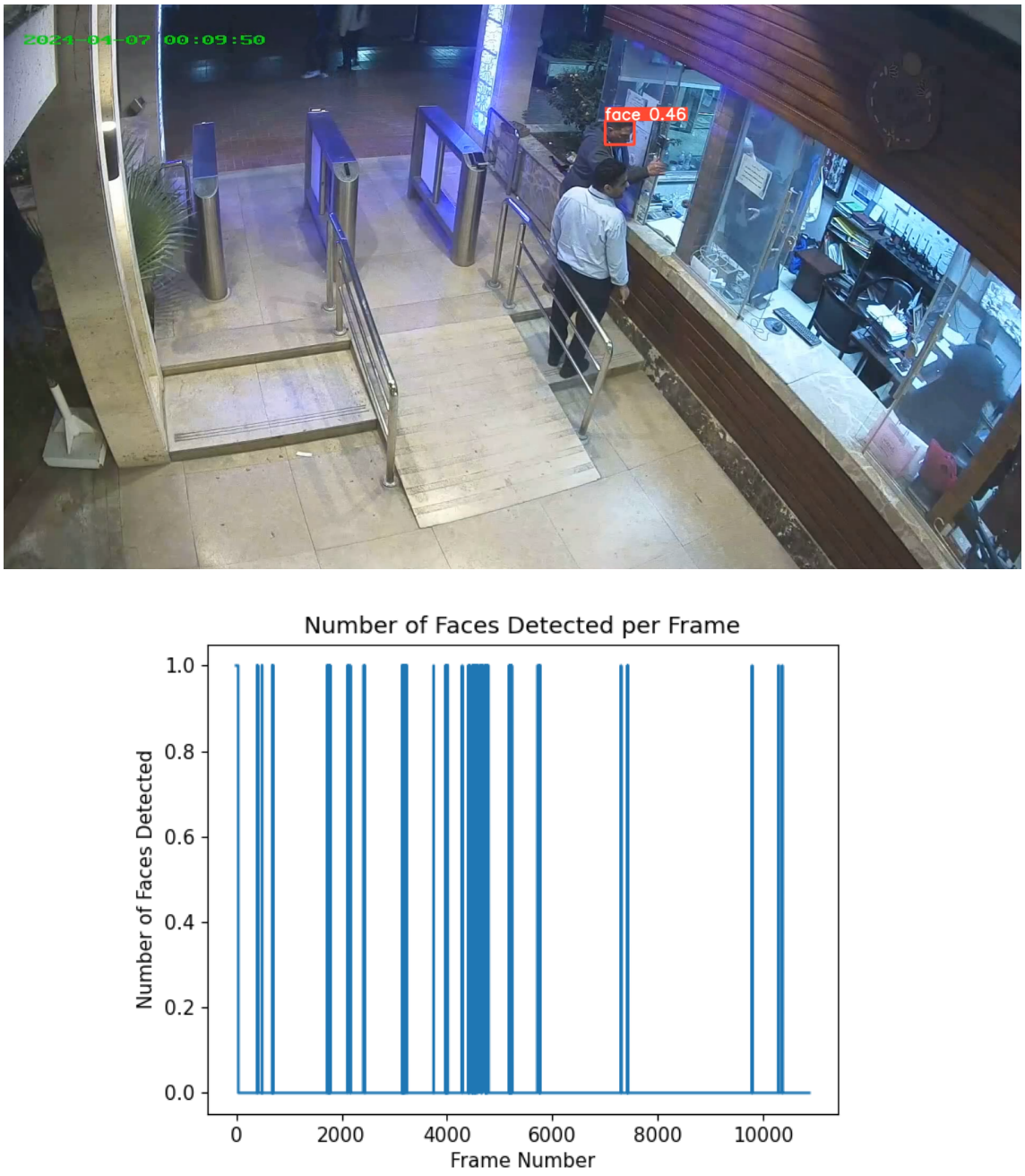

Figure 4.10: YOLOV9 test on video surveillance using confidence of 0.4

This is the result of the trained model on a video surveillance 4.2. Due to the videos poor lighting conditions and few faces present the model performed poorly not able to detect more than 1 face in a single frame. Having an average precision of 12.5%.

Limitation

This model could be improved on a bigger and more diverse dataset and train the model on more epochs than 30 to have higher average precision and detect the faces that it

could not detect in the video surveillance video above. However, I could not train the model more than this due to the limitation of my laptop processing power and speed and the time limit of both google colab and kaggle.

4.3.2 ResNet based face detection

The ResNet[42] pre-trained feature based face detection discussed in 3.2.2 had decent results on the the first video only because of most facial features being visible, however few faces was missed due to its limitations, there were issues when faced with different conditions that are similar to a video surveillance camera conditions such as in second and third (4.2) video. Having low resolution input image could result in no detection due to some feature not being entirely visible. Additionally, when a face is sideways the model can not detect the face. Furthermore, this model will not work in our favor as most faces are faraway or are small in size and the model is not able to identify them at all.

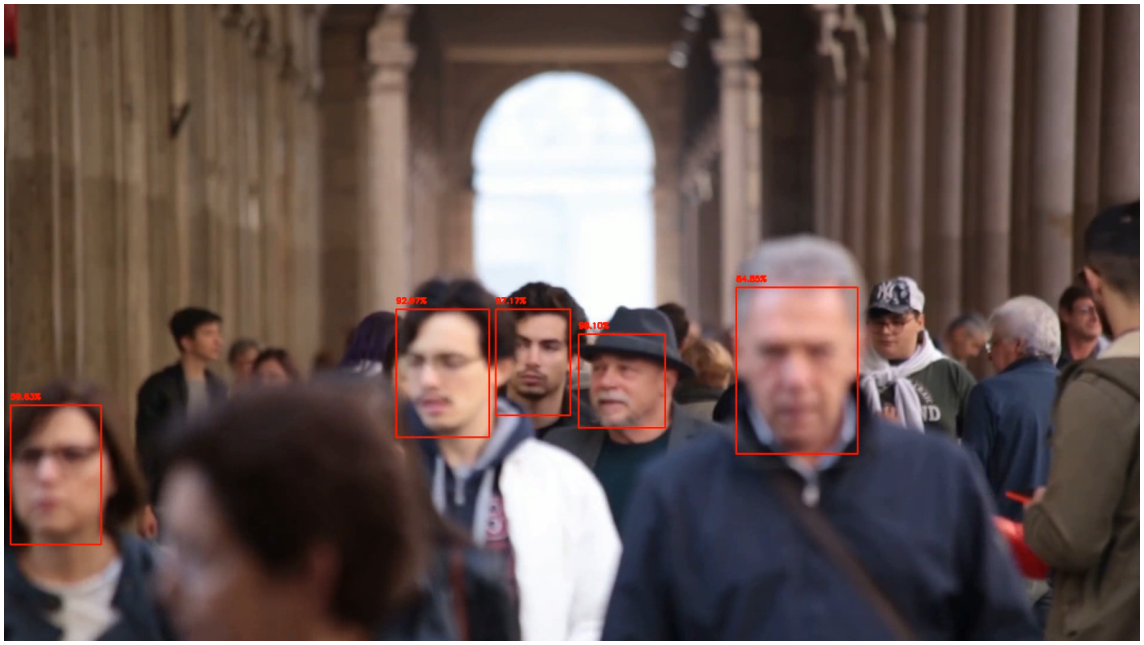

Figure 4.11: ResNet Result on first video surveillance with 0.4 confidence

Figure 4.11 show correct results, however it is missing multiple faces due to low resolution, occlusions and small face sizes as well, giving it an average precision of 34.7%.

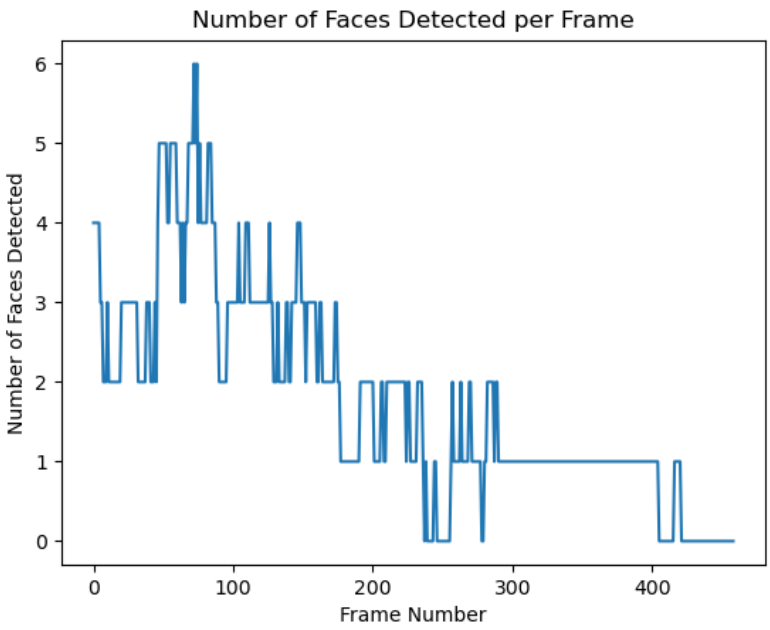

Figure 4.12: Result graph of ResNet

Figure 4.12 show the total number of faces detected in each frame.

However, for the other 2 videos the ResNet model was not able to detect any faces as all face sizes were small and their facial features were not clearly visible. As clearly shown in figures 4.13 the model were not able to detect any faces in third video surveillance and false readings in the second. Giving it an average precision of 0% in both of these videos.

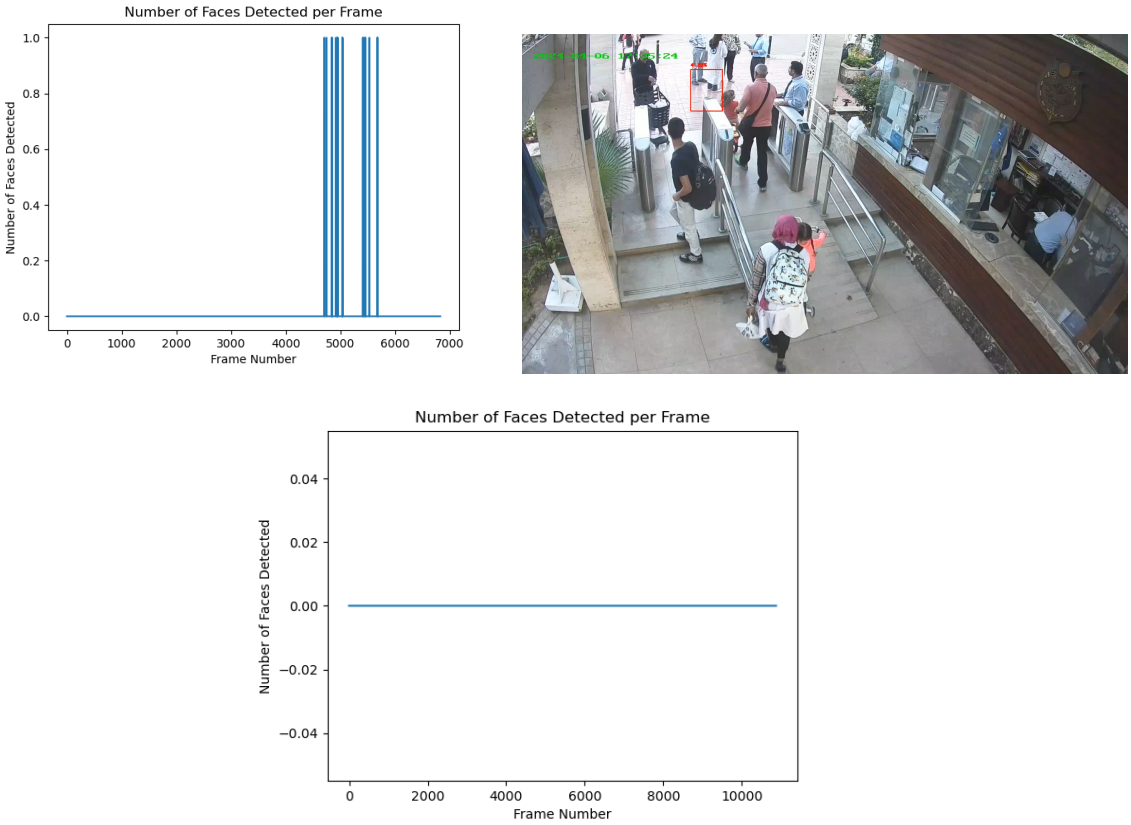

Figure 4.13: Result graph of ResNet

4.3.3 RetinaFace based face detection

The RetinaFace model explained in 2.2.1 and 3.2.3, excels in detecting faces in various environments. Using same video sample in 4.10 the model was able to detect more faces than YOLO model and also able to detect various faces, where only half the face were visible due to various angles and occlusions. Additionally, the model was also able to detect faces given an low resolution or even blurred image.

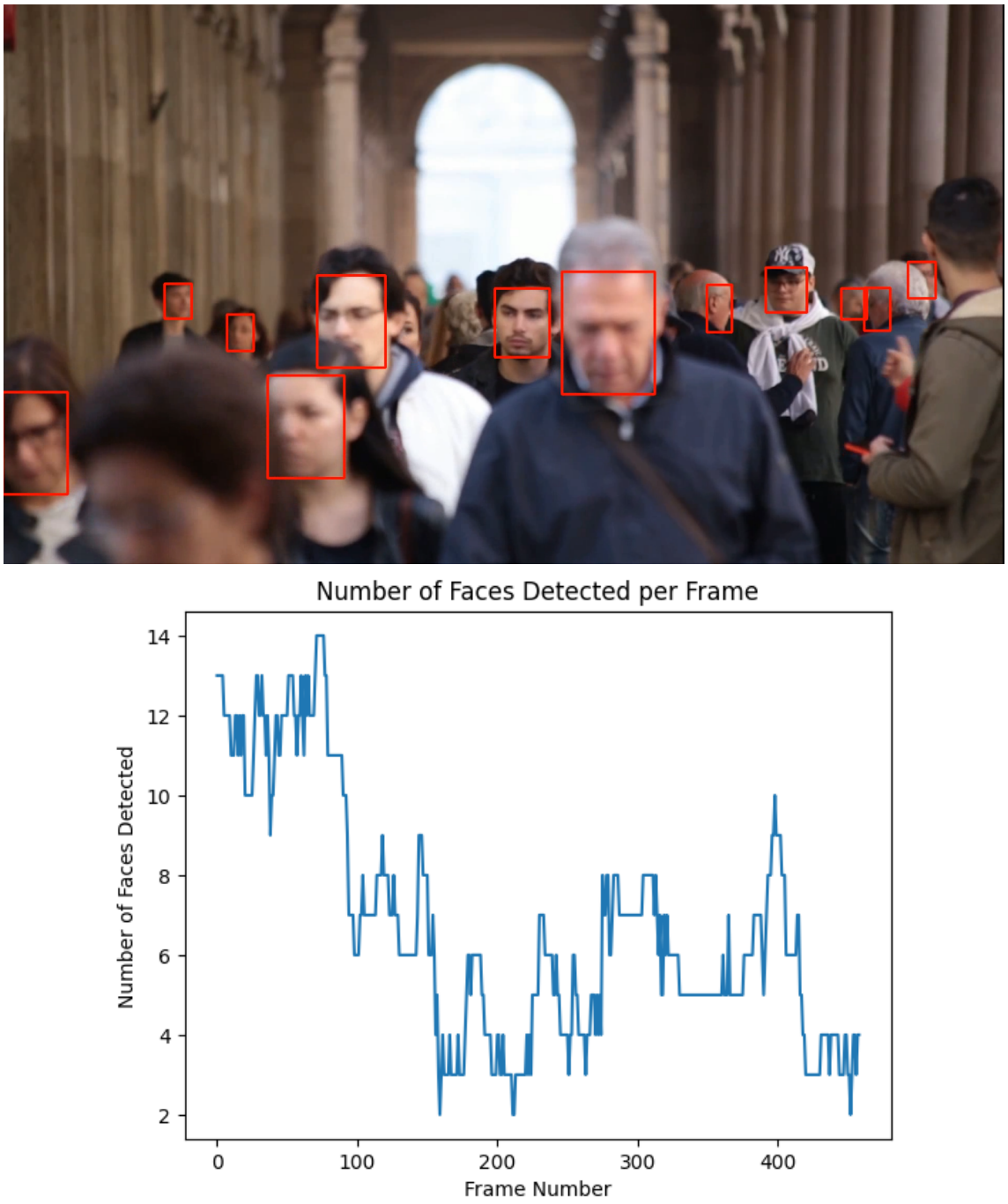

Figure 4.14: RetinaFace result on first video surveillance

Figures 4.14 show the result of the RetinaFace model on the first video, excelling in detection almost all faces, having an average precision of 99.3%.

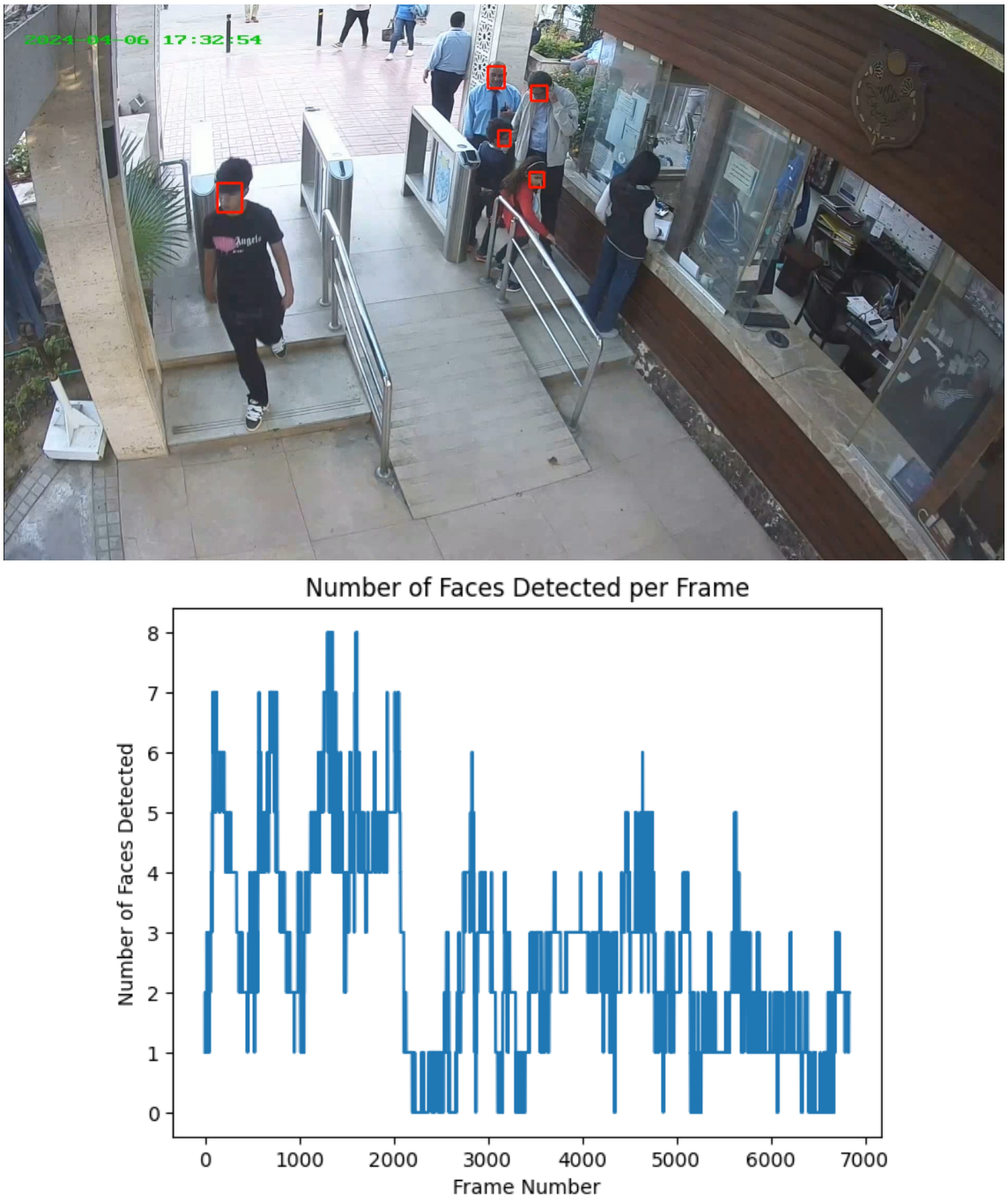

Figure 4.15: RetinaFace test on second video surveillance

Figures 4.3.3 show the result of the RetinaFace model on the first video, able to detect almost all faces missing only a few, having an average precision of 98.1%.

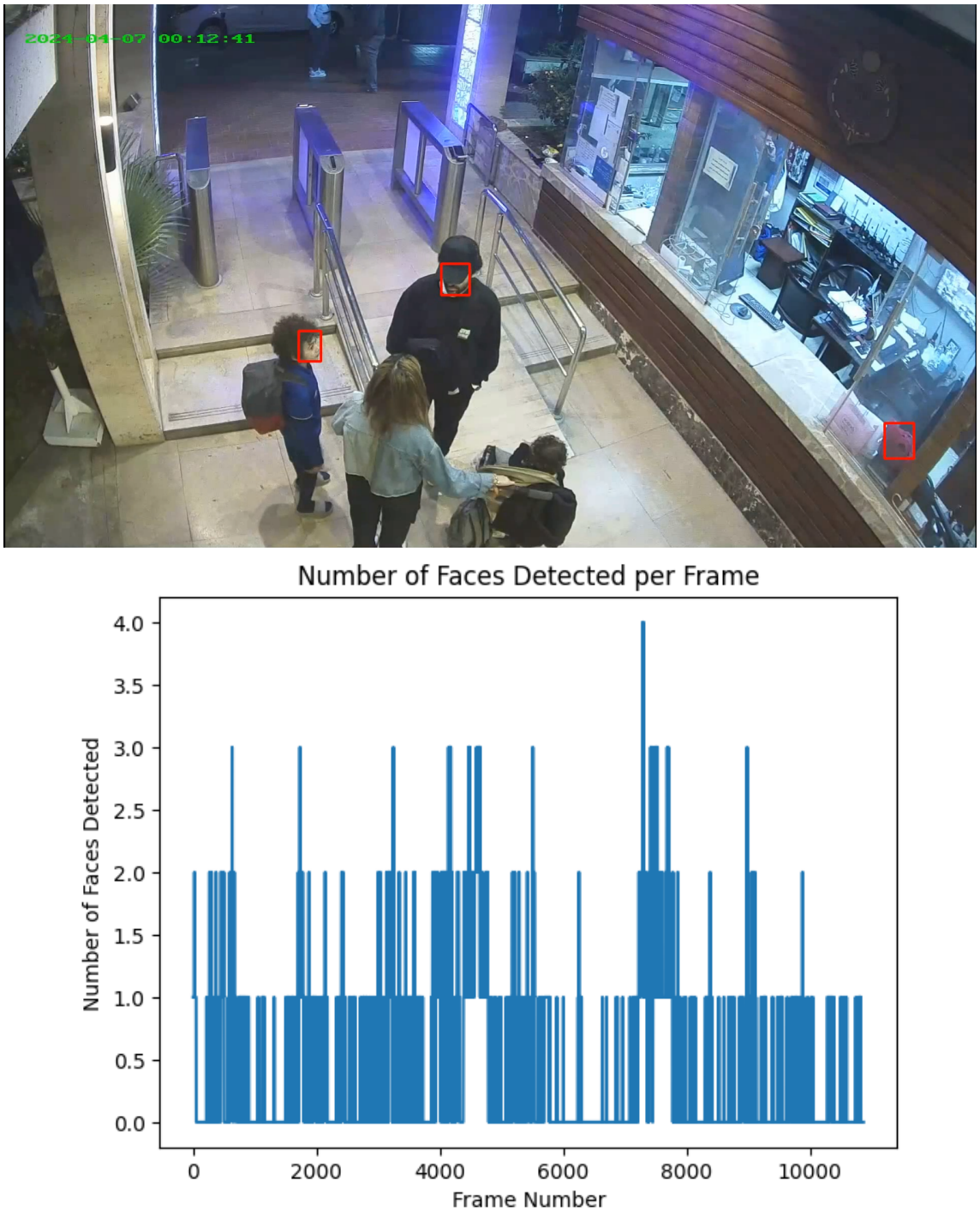

Figure 4.16: RetinaFace test on third video surveillance

Figures 4.3.3 show the result of the RetinaFace model on the third video, having few misses and false negatives giving it an average precision of 93%.

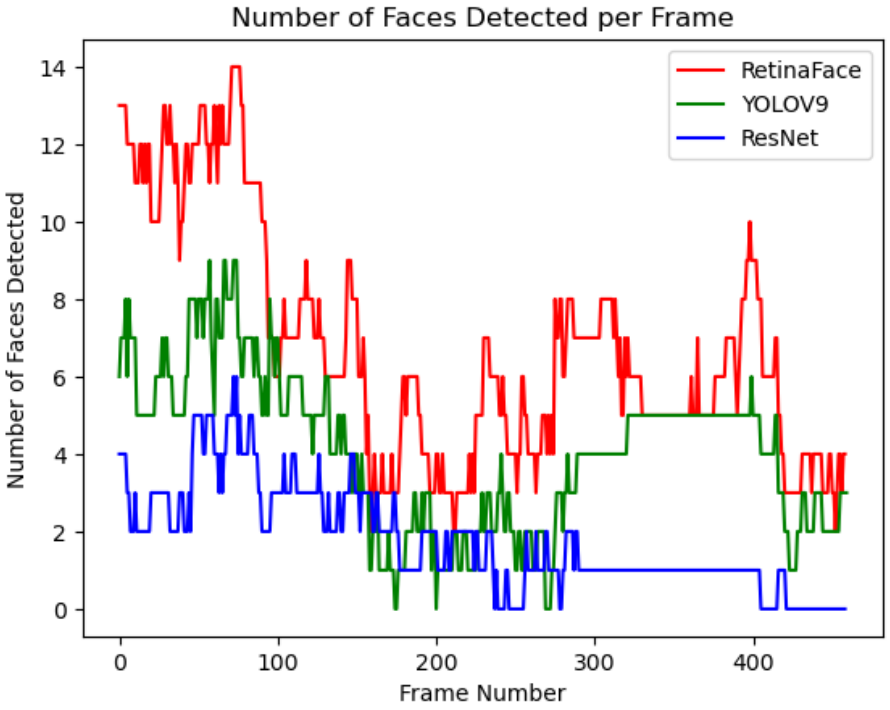

Figure 4.17: Comparison of the 3 face detection model on first video surveillance.

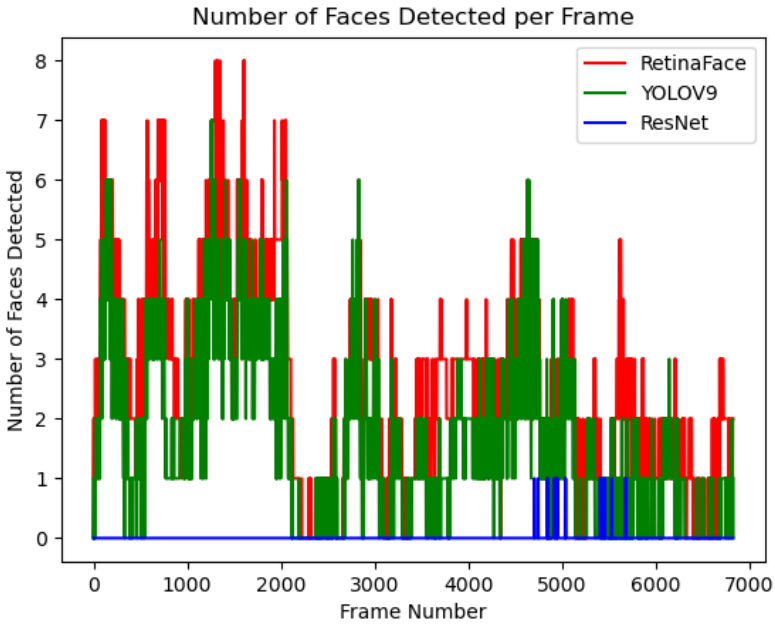

Figure 4.18: Comparison of the 3 face detection model on second video surveillance.

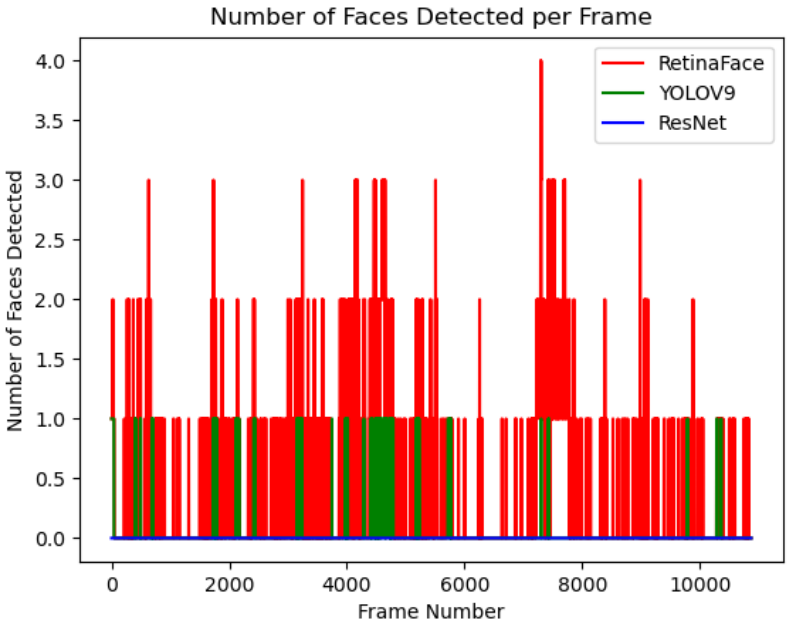

Figure 4.19: Comparison of the 3 face detection model on third video surveillance.

As clearly shown in figures 4.17,4.18 and 4.19 RetinaFace performance is far superior to other models that is why is was chosen over the others in our final model.

Video Surveillance	Total Frames	RetinaFace	YOLOV9	ResNet
Video 1	460	99.3%	75%	34.7%
Video 2	6832	98.1%	62.8%	0%
Video 3	10871	93%	12.5%	0%

Table 4.1: Average precision of the 3 models on video surveillance.

The table above 4.1 supports our decision, because of RetinaFace average precision across our video surveillance dataset.

4.4 Face recognition

Using face recognition model discussed in 3.3, I experimented with model under different conditions in order to know how it performs and what is its limitations.

Firstly, using the LFW dataset to check for models accuracy, after gathering various different photos from the internet of popular figures in the dataset. Such as Joe Biden,

Jennifer Lawrence and more. Providing a result of an accuracy equal to 99%. However these results are not on the whole dataset and that is due to having a number of figures who are unknown and could not find any alternative images of online.

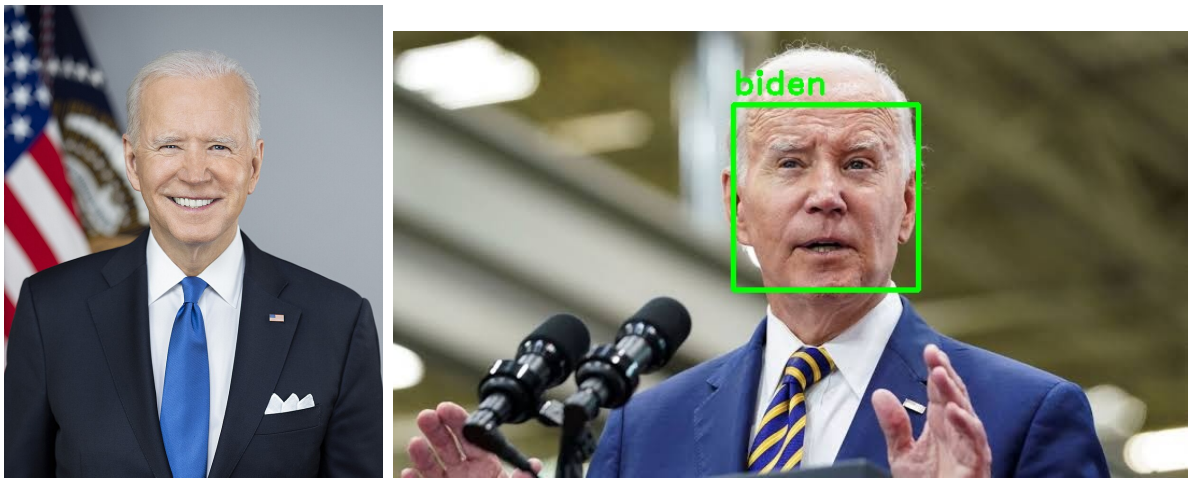

Figure 4.20: on the left is an image of Biden from dataset and second image the result.

Figure 4.21: Results of on different celebrities.

Figures 4.20 and 4.21 shows the face recognition model results on celebrities with different poses and expression than the ones in our database. However, The model faced multiple issues with some images. This face recognition model is trained on adults and does not work well with children. Additionally, accuracy can vary between different ethnic groups. Moreover, using the model on a video surveillance sample video, the model built in function to detect faces could identify a single face to recognize, that is due to the faces size small and resolution low.

Figure 4.22: Eye level video surveillance

Figure 4.22 show the result of a eye level video surveillance. The reason for choosing eye level rather the traditional birds eye view is because in the eye level we can see the whole facial features, while the other view can not completely view all facial landmarks when encoding the face detected.

4.5 DeepFace AGR detection

After giving the deepface model the coordinates of where a face is located, we use the analyze function to specify the individuals age. gender and race.

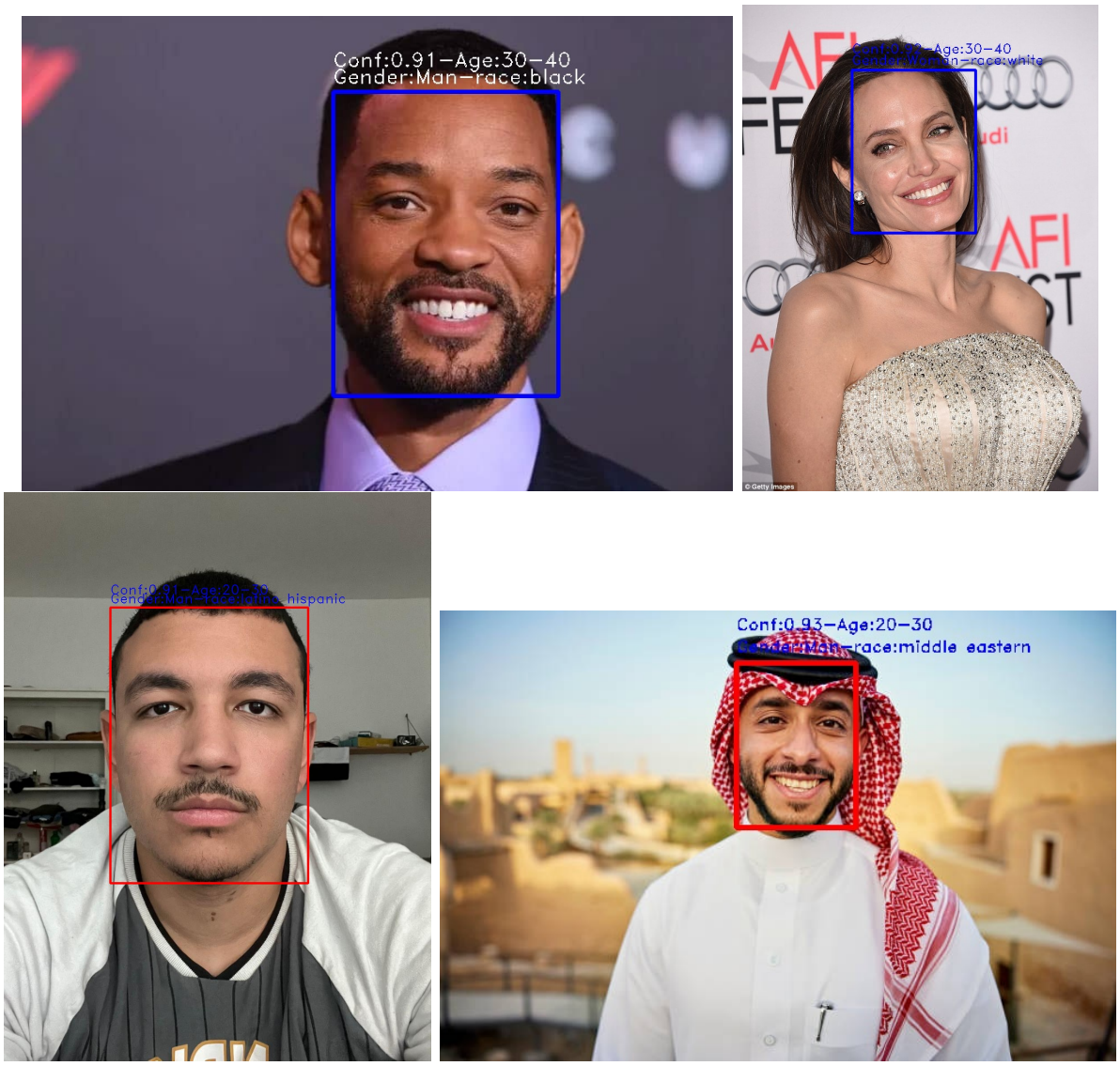

Figure 4.23: AGR detection results

Figures in 4.23 above showcase models result on different persons from around the world and accurately calculating their AGR.

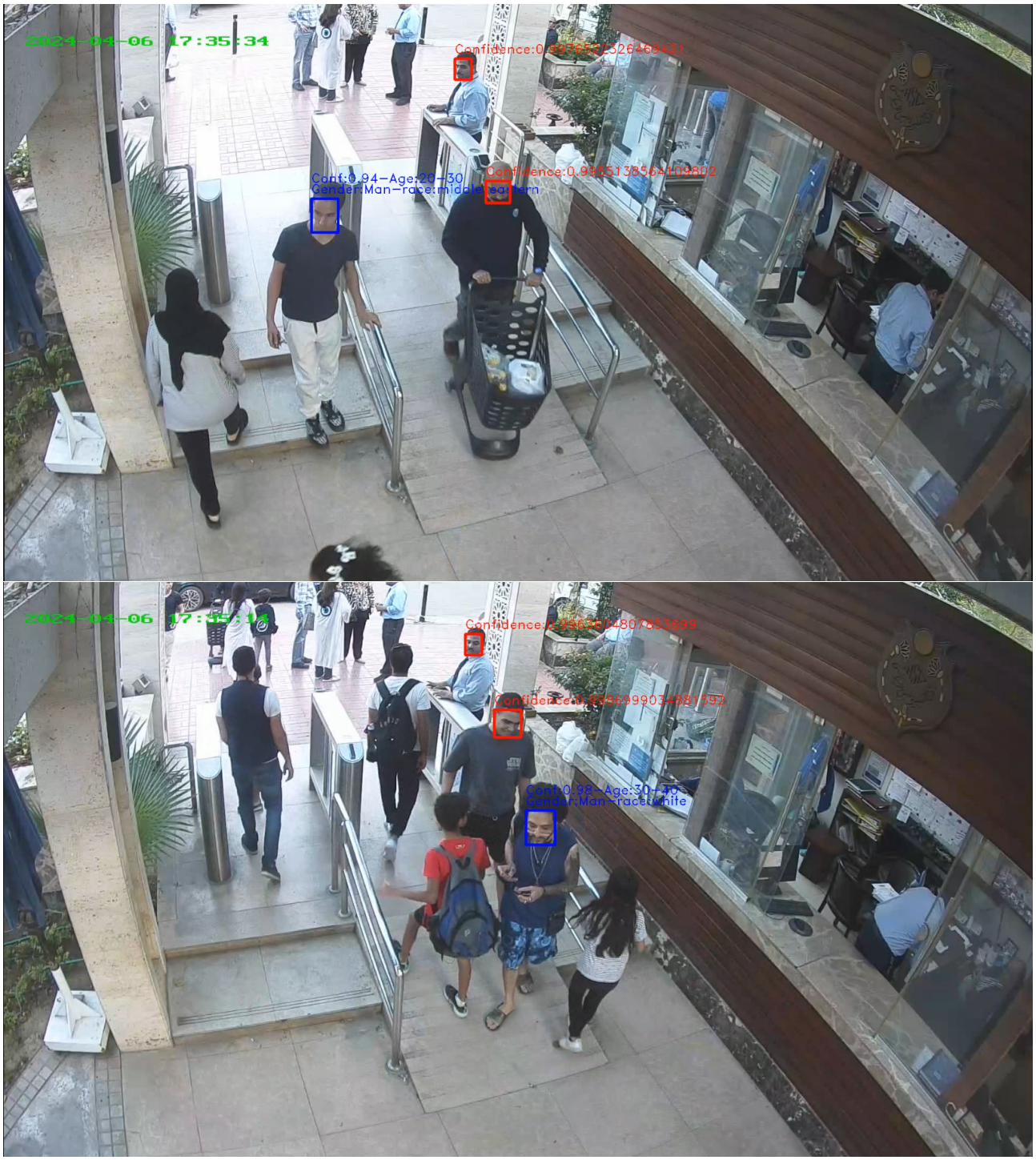

Figure 4.24: Results from our models final version

Figures 4.24 showcase the results of our final model on video surveillance footage. By clearly being able to detect faces and computing their AGR as well in every Frame and waiting to find a match if any known individual in our database appears.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we discussed the development of a model that is able to detect individuals faces in video surveillance video, identifying these individuals from our database and knowing age, gender and race using deep learning techniques and computer vision.

In the literature review 2.2, we reviewed various deep learning techniques such as CNNs(2.1.2), ResNet(3.2.2) and YOLO(2.1.4) that were used by different research papers. While some papers focused on increasing the precision and recall of the model, others focused on prioritising speed and making their models lightweight for more compact devices. For face detection, WIDER FACE[36] was the main dataset that was used for training, validating and testing. For face recognition, LFW[37] dataset was only used to evaluate the models accuracy, while MegaFace2[38], VGGFace2[39] and MS Celeb-1M[40] datasets were used to train, validate and also test the model. In the literature review there was a gap in the research where the models created were not tested on real-time video surveillance. Additionally, with AGR analysis would be beneficial contribution to my thesis.

After experimenting and testing with 3 different model for face detection, I opted to use RetinaFace(3.2.3) model. There are many reasons supporting this decision. ResNet feature based model had issues with small face sizes and occlusions, making it inefficient for any video surveillance footage. YOLO was far better ResNet, it was able to detect small faces in surveillance footage as shown in the results here 4.10, However, due to the limitation issues I faced (4.3.1) there were some faces the could not be identified and confidence had to be low to get the results shown. RetinaFace was far superior in detecting faces, it was able to detect all faces in surveillance footage with high threshold. For face recognition, the model has limitations as all facial feature has to be shown to ensure accuracy, so small face sizes and low resolution might give us unreliable results. DeepFace(3.4) was use for the AGR detection within the face, same as face recognition all facial features has to be shown and it may be biased towards certain age or gender, which may to lead to some inaccuracies.

Our model has 3 stages, with each frame we detect the faces present in this frame first, then after saving the coordinates of these faces, we loop over each face. Each face pass through the second stage and analyse the face for age, gender and race serving as a preliminary stage for feature extraction. Finally, after encoding the face features, we compare the results to the database and see best match.

By creating this model, real-time recognition allows security staff to focus suspicious activity by quickly identifying known individuals in a crowd, as early identification and elimination of human error leads to a faster investigation process. Additionally, The presence of such a system can discourage criminal activity because of the increase in risk of detection. By adding automated routine tasks such as monitoring a facility during a night shift, security personnel can focus on different tasks, as if an individual in detected alert will be sent to the security. Also the number of false alarms will be reduced due to the absence of human error. By looking at the demographics of individuals who are shown on camera, users can explore more about the individuals behavior. These data can be used to identify the areas that need stronger security measures. Customized advertising or promotions could also be made with the help of this demographic data.

5.2 Future Work

To further enhance our model in the future we must address its limitations, more techniques should be investigated in order to improve face recognition in surveillance footage for recognizing individuals with small faces and low resolution. Also to address DeepFace bias in age and gender detection, this could involve using a more balanced dataset or further train DeepFace model to adjust this bias.

Although our model already has useful real-time applications there is still room for improvement and further value creation. We can incorporate the recognition of human emotions into it in future developments. Peoples emotional states could be deduced from their facial expressions by adding an emotion recognition layer to the model which would enable it to do more than just recognize faces. By incorporating an emotion recognition layer into the model, the model would be able to read peoples facial expressions and determine their emotional state in addition to identifying them. For security professionals this information can be extremely helpful, because it can reveal a persons possible intentions or reactions to a given circumstance. Adding object detection and tracking can greatly improve the models situational awareness. This would enable the model to detect and monitor suspicious items like weapons or unattended packages in addition to persons of interest. Moreover, it might provide the user with an analysis of how long a person spends on the property by calculating the duration of the persons presence in front of the camera.

Bibliography

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[2] towardsdatascience.com, "neural networks." https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529, 2019. CNN.

[3] tezeract.ai, "Yolo algorithm." https://tezeract.ai/yolo-algorithm-real-time-object-detection/. YOLO.

[4] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[5] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 23–27, IEEE, 2020.

[6] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review*, vol. 52, pp. 927–948, 2019.

[7] F. Zhang, X. Fan, G. Ai, J. Song, Y. Qin, and J. Wu, "Accurate face detection for high performance," *arXiv preprint arXiv:1905.01585*, 2019.

[8] Y. Zhu, H. Cai, S. Zhang, C. Wang, and Y. Xiong, "Tinaface: Strong but simple baseline for face detection," *arXiv preprint arXiv:2011.13183*, 2020.

[9] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang, "Dsfd: dual shot face detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5060–5069, 2019.

[10] B. Zhang, J. Li, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Xia, W. Pei, and R. Ji, "Asfd: Automatic and scalable face detector," *arXiv preprint arXiv:2003.11228*, 2020.

[11] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "Faceboxes: A cpu real-time face detector with high accuracy," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–9, IEEE, 2017.

[12] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 8231–8238, 2019.

[13] D. Qi, W. Tan, Q. Yao, and J. Liu, "Yolo5face: why reinventing a face detector," in *European Conference on Computer Vision*, pp. 228–244, Springer, 2022.

[14] Y. Liu, F. Wang, J. Deng, Z. Zhou, B. Sun, and H. Li, "Mogface: Towards a deeper appreciation on face detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4093–4102, 2022.

[15] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 797–813, 2018.

[16] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li, "Detecting faces using region-based fully convolutional networks," *arXiv preprint arXiv:1709.05256*, 2017.

[17] M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality adaptive margin for face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18750–18759, 2022.

[18] Y. Xu, W. Yan, G. Yang, J. Luo, T. Li, and J. He, "Centerface: joint face detection and alignment using face as point," *Scientific Programming*, vol. 2020, pp. 1–8, 2020.

[19] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, p. 342, 2020.

[20] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*, British Machine Vision Association, 2015.

[21] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, "Ghostfacenets: Lightweight face recognition model from cheap operations," *IEEE Access*, 2023.

[22] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "Edgeface: Efficient face recognition model for edge devices," *arXiv preprint arXiv:2307.01838*, 2023.

[23] B. Li, T. Xi, G. Zhang, H. Feng, J. Han, J. Liu, E. Ding, and W. Liu, "Dynamic class queue for large scale face recognition in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3763–3772, June 2021.

[24] I. Kim, S. Han, S.-J. Park, J.-W. Baek, J. Shin, J.-J. Han, and C. Choi, "Discface: Minimum discrepancy learning for deep face recognition," in *Proceedings of the Asian conference on computer vision*, 2020.

[25] P. Terhörst, M. Ihlefeld, M. Huber, N. Damer, F. Kirchbuchner, K. Raja, and A. Kuijper, "Qmagface: Simple and accurate quality-aware face recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3484–3494, 2023.

[26] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Elasticface: Elastic margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1578–1587, 2022.

[27] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[29] T. Mare, G. Duta, M.-I. Georgescu, A. Sandru, B. Alexe, M. Popescu, and R. T. Ionescu, "A realistic approach to generate masked faces applied on two novel masked face recognition data sets," *arXiv preprint arXiv:2109.01745*, 2021.

[30] M. Knoche, M. Elkadeem, S. Hörmann, and G. Rigoll, "Octuplet loss: Make face recognition robust to image resolution," in *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8, IEEE, 2023.

[31] F. Liu, M. Kim, A. Jain, and X. Liu, "Controllable and guided face synthesis for unconstrained face recognition," in *European Conference on Computer Vision*, pp. 701–719, Springer, 2022.

[32] G. G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, and S. Zafeiriou, "Deep polynomial neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 8, pp. 4021–4034, 2021.

[33] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[34] Y. Xu, J. Yang, H. Cao, K. Wu, M. Wu, and Z. Chen, "Source-free video domain adaptation by learning temporal consistency for action recognition," in *European Conference on Computer Vision*, pp. 147–164, Springer, 2022.

[35] J. Guo, J. Deng, A. Lattas, and S. Zafeiriou, "Sample and computation redistribution for efficient face detection," *arXiv preprint arXiv:2105.04714*, 2021.

[36] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] www.cs.umass.edu, "Lfw." https://vis-www.cs.umass.edu/lfw/, 2019. LFW.

[38] A. Nech and I. Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[39] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 67–74, IEEE, 2018.

[40] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 87–102, Springer, 2016.

[41] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[42] K. Rathod, "Feature based resnet face detection." https://github.com/keyurr2/face-detection, 2019. ResNet.