MULTI-CAMERA FACE DETECTION AND RECOGNITION IN AN UNCONSTRAINED ENVIRONMENT

LEE KIAN HUANG

UNIVERSITI TUNKU ABDUL RAHMAN

MULTI-CAMERA FACE DETECTION AND RECOGNITION IN AN UNCONSTRAINED ENVIRONMENT

LEE KIAN HUANG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Biomedical Engineering with Honours

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Lee Kian Huang

ID No. : 18UEB04050

Date : 12th May 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"MULTI-CAMERA FACE DETECTION AND RECOGNITION IN AN UNCONSTRAINED ENVIRONMENT"** was prepared by **LEE KIAN HUANG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Biomedical Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	
Supervisor	:	Ir Ts Dr Tham Mau Luen
Date	:	12 May 2023

Signature	:	
Co-Supervisor	:	Dr Kwan Ban Hoe
Date	:	12 May 2023

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my research supervisor Ir. Ts. Dr. Tham Mau Luen and co-supervisor Dr. Kwan Ban Hoe, for their invaluable guidance, support, and encouragement throughout my research. Their expertise and insights were essential in shaping the direction of my work and ensuring its quality.

I am also grateful to my moderator Ir. Ts. Dr. Hum Yan Chai, for his insightful feedback and suggestions, which greatly improved my thesis.

In addition, I would also like to acknowledge the support of my family and friends, whose encouragement and understanding sustained me during the challenging times of my research. Their unwavering support and love were invaluable.

ABSTRACT

Multi-camera face detection and recognition is an Artificial Intelligence (AI) based technology that leverages multiple cameras placed at different locations to detect and recognize human faces in real-world conditions accurately. While face detection and recognition technologies have exhibited high accuracy rates in controlled conditions, recognizing individuals in open environments remains challenging due to factors such as changes in illumination, movement, and occlusion. In this project, the multi-camera face detection and recognition is developed in unconstrained environment setting. The multi-camera solution can overcome these challenges by capturing images of individuals from different angles and lighting conditions, thus providing a more comprehensive view of the monitored area. The pipeline in this project consists of three main parts – face detection, face recognition, single and multi-camera tracking. A series of models training is done with the open-source dataset to build a robust pipeline, and finally, the pipeline adopted trained YOLOv5n for the face detection model with mean Average Precision (mAP) of 0.495. The system also adopted the SphereFace SFNet20 model with an accuracy of 82.05% and a higher inference rate as compared to SFNet64 for face recognition. These models are then fed into DeepSORT for multi-camera tracking. Our dataset has been applied to the pipeline and shown ideal outcomes with objectives achieved. The solution feasibility is demonstrated via prototype implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

ACC	Accuracy
AI	Artificial Intelligence
AUC	Area under ROC Curve
CCTV	Closed Circuit Television
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSPNet	Cross Stage Partial Network
DCNN	Deep Convolutional Neural Network
DeepSORT	Simple Online Realtime Tracking with deep association metric
DNN	Deep Neural Network
EER	Equal Error Rate
FPS	Frame Per Second
eGPU	External Graphics Processing Unit
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
ID	Identification
iGPU	Integrated Graphics Processing Unit
IoU	Intersection of Union
IR	Intermediate Representation
mAP	Mean Average Precision
ML	Machine Learning
MTMCT	Multi-Target Multi-Camera tracking
OpenCV	Open Source Computer Vision Library
OpenVINO	Open Visual Inference and Neural network Optimization
PANet	Path Aggregation Network
ReLu	Rectified Linear Unit
Re-ID	Person Re-identification
SphereFace-R	SphereFace Revived
SSD	Single Shot Detector
UFDD	Unconstrained Face Detection Dataset
WDA	Weighted Distance Aggregation

VPU	Vision Processing Unit
YOLO	You Only Look Once
YOLOv5	You Only Look Once version 5
YOLOv8	You Only Look Once version 8

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Face detection has been broadly used in computer vision applications. It is an artificial intelligence (AI) based technology that is able to identify and recognize human faces in digital images. The face detection and recognition techniques exhibit a good biometrics application especially in surveillance system as it is frictionless. Individuals can be recognized from distance in the video footage without any physical contact such as fingerprint or iris recognition. This technique has been widely studied and shown a high accuracy rate under controlled conditions (Rambach, Huber, Balthasar and Zoubir, 2015). However, video-based face recognition in an open environment is still challenging especially in the field of changes in illumination, noise due to movement of the subjects and facial occlusion (Ristani and Tomasi, 2018).

Multi-camera video-based face recognition systems have emerged as a potential solution to the challenges faced in open environment face recognition. By leveraging multiple cameras, these systems can capture images of individuals from different angles and lighting conditions, increasing the accuracy of face detection and recognition. Additionally, these systems can provide a more comprehensive view of the monitored area, enhancing the security and surveillance capabilities of the system. Despite the challenges, the development of multi-camera face recognition systems is a promising area of research with significant potential applications in various fields. In this project, a comprehensive system to detect faces in an unconstrained environment is structured with multi-camera video-based face recognition.

The multi-camera face recognition is a way of tracking individuals by combining the trajectory information from multiple cameras with the recognized faces (Israel and Bolton, 2020). It enables the position of individual to be detected at real times with the video streams from multi-cameras. It brings advantage of the limitation of the vision field in single camera and provide a more detailed analysis of the target through multiple cameras. The applications of this multi-camera face recognition include surveillance system, anomaly

detection, sports analysis as well as crowd behaviour analysis (Ristani and Tomasi, 2018).

The implementation of multi-camera face detection and recognition has to consider several factors such as the number of cameras used, whether the field of view is overlapping and camera placement. A limited number of cameras used will form a limited tracking process, but the higher number of cameras would cause the complexity of the system and overlapping of field of view issues. Besides, the location and placement of cameras are critical to the effectiveness of a multi-camera face detection and recognition system. Cameras should be placed in a way that maximizes the coverage of the area to be monitored while minimizing obstructions and other factors that can reduce the quality of the captured images. All these are important to take into consideration, as they will affect the occlusion periods, illumination in different fields of view, and data storage which lead to delay in detection and recognition (Rambach, Huber, Balthasar and Zoubir, 2015).

The basic working principle of multi-camera face detection and recognition is that it involves the use of multiple cameras placed at different locations to capture face images from different viewpoints. The images captured by these cameras are then processed using computer vision algorithms to detect and recognize faces.

First, the person's face is captured by a specific camera, and it is stored in the database with a specific ID. The face re-identification (Re-ID) system will retrieve from the database a list of shots captured from different cameras at various times and rank it in descending order of similarity to the specific person (Ristani and Tomasi, 2018). These features will be trained with the convolutional neural network (CNN). The multiple tracking is done with the function of DeepSORT and Weighted Distance Aggregation (WDA) which tracks velocity and motion as well as the appearance of the person detected, and last data clustering is performed to reduce the complexity of the data. It is a complex process that involves sophisticated computer vision and machine learning techniques to accurately and reliably detect and recognize faces in unconstrained environments.

In the context of an unconstrained environment face recognition, it indicates that the subjects are mobile, and captured in real-world conditions that

are uncontrolled and unpredictable. The video footages consist of noises which is due to motion causing it blurring, occlusion, lighting, facial expression, and postures (Bialkowski, Denman, Sridharan, Fookes and Lucey, 2012). This will lead to inaccuracy of face recognition and failed to track the subject through multiple cameras (Saffar, Rekabdar, Louis and Nicolescu, 2015). Thus, a comprehensive and robust system designed for unconstrained environments needs to be implemented to variations in illumination and pose, which can cause significant changes in the appearance of a face. They must also be able to handle occlusions and able to identify individuals even if they are wearing different expressions or have slight changes in appearance. With that, it allows the multiple camera face detection and recognition able to identify the subject in the unconstrained environment.

1.2 Importance of the Study

The implementation of multi-camera face detection and recognition allows the back end to determine the position of the individual at a specific time frame that is within the vision field of the cameras. It matches all the local tracklets from all the cameras and produces a full trajectory for each subject across the whole multi-camera network (He, Wei, Hong, Shi and Gong, 2020).

The multi-camera face recognition provides a wide range of commercial and law enforcements application. It provides a more comprehensive view of monitored areas, allowing for better tracking of subjects and reducing the likelihood of errors that may occur in single-camera systems. Moreover, the technology allows for detection and recognition of individuals from multiple angles, increasing the system's ability to identify potential threats.

It can be used in the airport surveillance system, the implementation of the system in an unconstrained environment allows the surveillance process run in a more effective way as the system is able to recognize the individual with a higher accuracy with the reduction of occlusion, lighting condition, low resolution, and blurring factors.

It also allows the back-end security to identify the suspects who are committed in criminal activities by face recognition techniques over multiple cameras. As the coverage is widened with the implementation of multiple cameras, it able to improve the overall tracking activities from the suspects.

Overall, the development of multi-camera face detection and recognition in an unconstrained environment offers significant potential for enhancing security and surveillance measures in various fields. The technology is able to aid in investigations and ensuring the safety and security of individuals in high-risk areas.

1.3 Problem Statement

Despite that the face detection and recognition has been matured over the decades, there are still some challenges need to be addressed to improve the accuracy and efficiency of the systems on the video sequences in several aspects (Rambach, Huber, Balthasar and Zoubir, 2015). This is because video face recognition has limitations in recognising the faces when the resolution of the video is low, illumination, occlusions, noise due to movement of the subjects and lighting factors. In such environments, the lighting conditions, angles, and distances between cameras and subjects vary, making it difficult for the system to accurately detect and recognize faces (Ristani & Tomasi, 2018). Therefore, a more comprehensive system needs to be implemented to overcome the factors, to allow the face recognition system able to recognize faces in an open environment without any constraints.

In recent years, face detection and recognition have been widely used as it does not require any physical contact, the subjects can be detected and recognized by the camera setups (Wolf, Hassner and Maoz, 2011). When it comes to single camera, the field of view is limited, thus the subjects can only be recognized in specific coverage. This has restricted the surveillance system in law enforcement applications and crowd behaviour analysis. With the implementation of multiple cameras, it is able to improve the coverage, expand to a bigger area and recognize the subject in a wider range of fields within the time. However, in a multi-camera setup, multiple cameras are needed to cover a larger field of view, which introduces challenges such as occlusions and changing viewpoints.

Tracking and recognizing the subjects in a crowded scene is a complex problem as in a manual way, it requires a lot of time and manpower (Zervos, 2013). This problem can be solved in a single framework with the implementation of face recognition over multiple cameras. The subjects can be

tracked in the system seamlessly across multiple cameras. The process of tracking a face across different cameras is complicated by changes in lighting conditions, viewpoints, and occlusions. Thus, accurate tracking of faces is essential for recognizing individuals across different cameras and creating a complete trajectory of the person's movement.

1.4 Aim and Objectives

This project aims to build a multi-camera face detection and recognition system in an unconstrained environment that can be used in commercial and law enforcement applications. The unconstrained environment in this context refers to video footage that is shown in a low resolution, varying lighting conditions, arbitrary poses, noise due to subject motion, illumination, and occlusions. A series of back-end training is needed to allow the system able to recognize and tackle the subject under an unconstrained environment. The goal is to improve the accuracy and robustness of face recognition systems across different cameras and viewpoints while minimizing false positives and false negatives.

The objectives of the project are as follows:

1. To train the dataset so that the system is able to detect and recognize the face of the individuals in an unconstrained environment.
2. To develop the face detection and recognition of the system in an open area environment where the individuals may appear in low resolution and noises exist as well as illumination and occlusion happened.
3. To develop a multi-camera face tracking system that can accurately track faces across multiple cameras in an unconstrained environment.
4. To implement the system in a real-life situation under a multi-camera view.

These objectives are achieved through the use of advanced computer vision and machine learning techniques, such as deep learning-based face detection and recognition algorithms and feature extraction methods that are robust to variations in illumination, pose, and occlusion. With that, it allows the tracking activities to be done easily in multiple cameras in an unconstrained environment. The system is able to track the individuals seamlessly across multiple cameras disregarding the constrained factors that exist previously. The

multi-camera view able to improve the field of view and coverage, and able to track the subjects easily from one camera to another.

1.5 Scope of the Study

This project covers the implementation of a face detection technique using You Only Look Once version 5 (YOLOv5). Then, the detected face is recognized in the SphereFace model. It is a deep learning face recognition model that allows the system to recognize the faces of individuals. It is done by retrieving the snapshots of subjects from the database captured from different cameras at various times (Ristani and Tomasi, 2018). It is then ranked by descending order of similarity to the query. The given query to the snapshots from the database that are co-identical will be ranked higher.

A convolutional neural network (CNN) is structured to process the pixel data by reducing the images into a form that can be processed easily without losing the important features (Ben, Bouguezzi and Souani, 2021). This approach is to make sure a good prediction is done. It allows the model to understand the features of the image.

In the face tracking part, the face of the subject is detected and recognized from one single camera, a multiple object tracking algorithm – DeepSORT will be implemented which is able to track the subjects based on velocity and motion as well as their appearance. The occlusion will also be solved under the algorithm with Kalman Filter (Israel and Bolton, 2020).

In multi-camera tracking, a track distance calculation is formulated to determine the weighted distance aggregation based on the time constraints and feature distance (Kohl, Specker, Schumann and Beverer, 2020). Last, the associated data will be analysed with the unsupervised learning – clustering method. This is to reduce the complexity of the data (Israel and Bolton, 2020). The system is trained so that the subjects are able to recognize and track in an unconstrained environment.

1.6 Limitation of the Study

The limitation includes only non-overlapping cameras implemented in the system. It is assumed that the field of view of the overall system is independent of each camera, thus it is not possible for the specific subject to be

detected in the views of more than one camera at the same time (Kohl, Specker, Schumann and Beverer, 2020). It also indicates there is no homography-matching distance in such a scenario.

Besides, the proposed system may require significant computational resources as it is high computational complexity. It involves processing a large amount of data generated by multiple cameras. The system may require significant computational resources such as high-performance processors and GPUs, making it costly to deploy and maintain (Ristani and Tomasi, 2018).

Moreover, the limitation of the system is the potential privacy concerns that it raises, especially in public areas where individuals may not be aware that they are being monitored. The use of face recognition technology has raised ethical concerns and has been criticized for its potential misuse, such as unauthorized surveillance or data breach (Deng, Guo, & Zafeiriou, 2019). Therefore, it is essential to ensure that the system adheres to privacy regulations and guidelines to prevent any misuse of personal data.

Scalability is another limitation that may affect the proposed system's deployment in large-scale environments. As the number of cameras and processing units increases, the complexity of the system may also increase, leading to the need for more significant resources and infrastructure. In addition, the system's performance may deteriorate with the increase in the number of cameras and processing units, leading to reduced accuracy and reliability (Zhang et al., 2016).

1.7 Contribution of the Study

The main contribution of the study on multi-camera face detection and recognition in an unconstrained environment is the development of a comprehensive system that is able to detect faces in an open environment using multiple cameras and perform accurate recognition of the detected faces. The system is integrated with face detection from YOLOv5 and face recognition from the SphereFace model. Both of the models are trained with a dataset that comes from an unconstrained environment setting.

In addition, the study also explores different strategies for face tracking, including the use of Kalman filtering and multi-object tracking, to address the challenge of tracking faces across multiple cameras and maintaining face

identities. The study eventually used the weighted distance aggregation method which includes the Kalman filtering and multi-object tracking as well as data association by clustering.

Overall, the study's contribution lies in developing a practical and effective multi-camera face detection and recognition system that can operate in an unconstrained environment. This has significant implications for a wide range of applications such as surveillance, security, and access control.

1.8 Outline of the Report

The report first covers the introduction of the intended implementation of the system – multi-camera face detection and recognition in an unconstrained environment with its aim and objectives stated as well as the scope and limitation of the study. In Chapter 2, the Literature Review of the particular topic has been discussed from scratch such as the current technology, how to construct the system, models training, datasets as well as the front-end issues.

The research and methodology of the study are reviewed in the following chapter, Chapter 3. It discusses the method to implement the system and the criteria involved as well as the timeline to conduct the research. The evaluated study is then reported under Chapter 4 – Results and Discussion. Both qualitative and quantitative results are shown in this chapter, and extensive discussion has been done to analyze the results. The overall study will be summarized and recommendations for future improvements in sorted out in Chapter 5 – Conclusion and Recommendation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the current face detection and recognition technology and its technology is introduced. The flow of details of the multi-camera setup is also reviewed including the convolutional neural network models in face detection and face recognition, single camera tracking along with multi-camera tracking.

The literature on multi-camera tracking is reviewed from scratch. Firstly, the Multi-Target Multi-Camera tracking (MTMCT) concept on how it works with a multi-camera setup is studied. Then, the Convolutional Neural Network, on how it processes the input images through several layers to enable the system to understand the features of the image is reviewed. Face detection by the YOLOv5 model and face recognition by the SphereFace model which is to reidentify the detected faces from camera frames are also reviewed. It will output the bounding boxes which specify the locations of the detected targets and assign IDs based on the appearance features.

The tracking from a single camera is also studied from frames in a video sequence. It first comes with detection from each frame in bounding boxes and tracks with associated data to compute the final trajectories in a single camera. Some of the metrics are introduced for the single camera tracking for a better prediction. The optimization of the performance in a single camera is necessary as multi-camera tracking depends on every single camera that is linked.

Last, the multi-camera tracking is reviewed with weighted distance aggregation (WDA) of multiple distances with consideration of certain constraints. The single camera will be linked in the multi-camera setup, and the associated tracks data are generated from the hierarchical clustering. The privacy concerns and dataset used will also be discussed.

2.2 Overview of Current Face Detection and Recognition Technology

Face detection and recognition technology have made significant advancements in recent years, it is a popular topic in biometrics research. A person’s face plays

a crucial role in conveying their identity and feelings. Compared to machines, humans are not as good at distinguishing between distinct faces. Thus, an automatic face detection system is important in recognizing faces for purposes such as security, expression analysis, head-pose estimation, and human-computer interaction (Kumar, Kaur and Kumar, 2019).

Face detection is a computer technology to locate and identify human faces in digital images or videos (Kumar, Kaur and Kumar, 2019). Currently, deep learning-based algorithms, particularly convolutional neural networks (CNNs), have demonstrated impressive results in face detection applications. The most popular CNN-based face detection algorithms include the Viola-Jones algorithm, the Histogram of Oriented Gradients (HOG) algorithm, and the Deep Convolutional Neural Networks (DCNN) algorithm (Zafeiriou, Zhang and Zhang, 2015).

As for face recognition technology, it is a type of biometric technology that compares a person's facial features to those in a database to identify and verify their identity. In particular, Deep Neural Networks (DNNs) have displayed an astounding performance in facial recognition tasks out of other deep learning-based models. Deep neural network-based face recognition algorithms such as SphereFace, VGG-Face, DeepFace, and FaceNet are the most widely used in this technology (Zhang et al., 2021).

The integration of both face detection and recognition provides vast and varied applications, especially in security and surveillance systems, access control systems, social media, and e-commerce platforms. In addition, the technology has also found applications in the healthcare industry, especially in monitoring and diagnosing mental health conditions (Hussain et al., 2022).

Despite such face detection and recognition technology has several merits, the raised concerns should also be pointed out such as its ethical and privacy concerns. For example, the technology has the potential to violate people's privacy, especially in public areas. The technology can also be utilized for discriminatory activities such as monitoring and racial profiling (Smith and Miller, 2022).

The future of face detection and recognition technology is promising. It can be applied in 3D face recognition, facial expression recognition, and the integration of artificial intelligence (AI) and machine learning (ML) techniques

to improve the accuracy and performance of the technology. It has potential to improve efficiency, safety, and public experience in various fields and applications. The trend of addressing ethical and privacy concerns needs to be mitigated with strong enforcement from the relevant parties.

2.3 Overview of Deep Learning Method – Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a classification algorithm for deep learning, it inputs the image and processes the pixel data by assigning the meaning in several aspects of the image so that it can differentiate from one another. It can be trained to understand the image better and capture the spatial and temporal dependencies of the image with several filters (Saha, 2018). CNNs have achieved state-of-the-art performance in many image recognition tasks, such as object detection, image segmentation, and facial recognition. For example, the popular ImageNet dataset, which contains over 14 million images, has been used to train CNNs to recognize thousands of object categories with high accuracy (Krizhevsky et al., 2012). The purpose of implementing CNN into the system is to lower the complexity of the process by eliminating the image parameters without losing important features for accurate prediction. It consists of multiple layers which include a convolutional layer, a pooling layer, and fully connected layer.

The convolutional layer focuses on the learnable kernels, it determines the neuron's output which is linked to the input with the computation of the scalar product for the weights and the region linked to the input volume. The input data will convolve in each filter across the input spatial dimensions when it reaches a convolutional layer, creating a 2D activation map (Albawi, Mohammed and Al-Zawi, 2017). Then, the scalar product is computed for each value in a particular kernel, thus the network will understand the kernels and activate it when the particular feature of an input at a given spatial position is detected (O'Shea and Nash, 2015). The layers can be further optimized by hyperparameters – depth, stride, and padding. With that, the inputs with high-level features are extracted.

As for the pooling layer, it is to decrease the convolved feature's spatial size. It works by down sampling the input along its dimensionality and further

reducing the number of parameters in the activation (O'Shea and Nash, 2015). The purpose of doing so is to reduce the computational capability needed to manage the data and extract the dominant features. There are two categories of pooling – max pooling which yields the maximum value from the kernels and average pooling which yields the average value from the kernels. Max pooling is also responsible to filter and eliminate the noises from activations, thus it is also known as the noise suppressant (Saha, 2018).

The fully connected layer, it includes neurons which connected to neurons from neighbouring layers without connecting to other layers within them. It is arranged in the traditional form of an artificial neuron network; thus it performs the same duties as an artificial neuron network and attempts to get scores from the activations for classification purposes (O'Shea and Nash, 2015). Rectified linear unit (ReLu) is commonly applied between these layers to improve performance. ReLu is a piecewise activation function which outputs the input (previous layer) right away if it is positive, otherwise, it will result in zero. The goal is to employ features from the input image into different classes based on the training dataset (Coskun, Ucar, Yildirim and Demir, 2017).

The CNN architecture can be modified based on the performance of the system and design requirements (Pranav and Manikandan, 2020). In the application of CNN for face image processing, the CNN involves a large dataset of the pictures as input for the model to learn. The images are stitched, and features are extracted by the CNN model.

Despite its effectiveness in image recognition tasks, CNN also possessed several limitations that should be taken into consideration. One of the main limitations of CNN is its susceptibility to overfitting, especially when training data is limited (Zhang et al., 2016). Overfitting occurs when a model learns to recognize specific patterns in the training data but is unable to generalize to new data. This will lead to poor performance on real-life actual applications. Another limitation is the lack of interpretability of the learned features since CNN often learns complex, non-linear representations that are difficult to interpret (Simonyan et al., 2013). Furthermore, CNN requires large amounts of training data and computational resources, making them expensive and time-consuming to train.

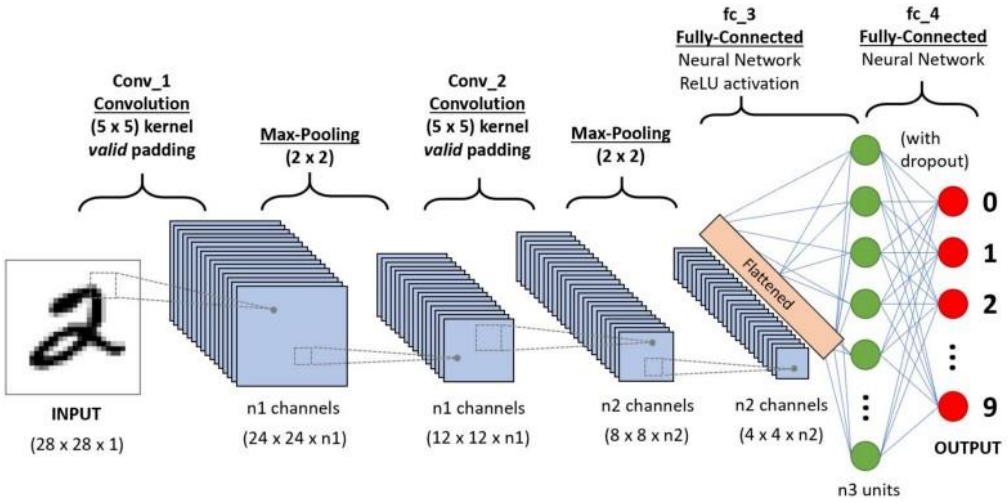

Figure 2.1: Typical Convolutional Neural Network (CNN) Architecture (Saha, 2018).

2.4 MTMCT – Multi-Target Multi-Camera Tracking

Multi-Target Multi-Camera tracking (MTMCT) is a field of computer vision and surveillance systems, with wide-ranging applications in fields such as security, traffic management, and retail analytics. The goal of MTMCT is to track multiple targets (such as people, vehicles, or objects) across multiple cameras, with the aim of accurately and efficiently monitoring their movements and interactions. It is a useful computer vision as it overcomes the limitation of the field of view in a single camera and allows the back end to conduct an analysis of the target.

MTMCT system works in two modules. They are single camera tracking and inter-camera tracking. Single camera tracking is to track the individual trajectories within a single camera. While inter-camera tracking is to re-identify the individual trajectories across multiple cameras with person re-identification (Re-ID) (Hsu, Cai, Wang, Hwang and Kim, 2021). Tracking by detection technique has been used in single camera tracking which is composed of object detection in frame and trajectory generation detections across the time (Hsu et al., 2021).

There are numerous unsolved problems about MTMCT which include the object occlusions and background noises that cause incomplete tracking results in a single camera, disparities in visual quality and ambient surroundings that make cross camera matching difficult, and last, an unknown number of

cameras where each target appears and the presence of numerous targets across multiple cameras that make it difficult to infer the global trajectory of each target (He et al., 2020).

Apart from that, there are problems arise that the field of view of cameras is not overlapping, they are placed far apart due to cost constraints. It will result in prolonged periods of occlusions, leading to significant changes in viewpoint and illumination across the vision field (Ristani and Tomasi, 2018). Another challenge in MTMCT is to improve the scalability of MTMCT algorithms so that they can handle large-scale surveillance systems with hundreds or thousands of cameras.

In the MTMCT system, the features of the individuals in the main camera are extracted after the system has identified them through detection and tracking. Then, the individuals' features are also taken from the sub-camera using the feature vector and searched in the feature space to identify the feature that matches the individual ID the most closely. This is done to construct the tracking across multiple cameras. Weighted Distance Aggregation (WDA) modular design is used to provide the convenience of exchanging components, including a person detection module, feature extraction module, and single-camera tracker (Kohl et al., 2020). Additionally, WDA incorporates a fundamental track comparison algorithm that computes five different feature distances between tracks.

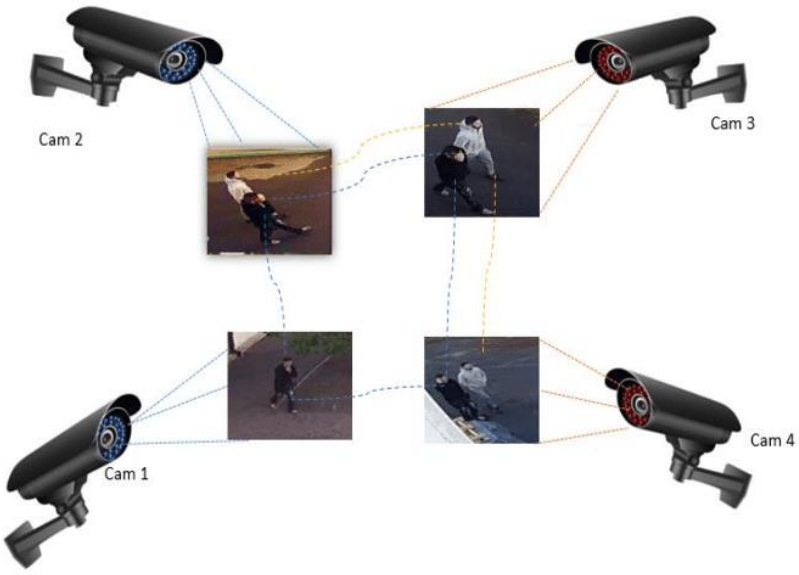

Figure 2.2: Overview of Multi Target for Multi-Camera Tracking (Kalake, Wan and Hou, 2021).

2.5 Human Face Detection

The system relies on the accuracy of human face detection, only it can assign identity to the individual and track through multiple cameras with a person re-identification (Re-ID) feature. The outputs take the form of rectangular bounding boxes which indicate the positions of the detected individuals. The bounding boxes are also known as the region of interest, it allows the system to localize and recognize the faces in images. The process comprises three different tasks which are image classification, object localization, and object detection (Li, Ma, Sajid, Wu and Wang, 2020). It first comes with image classification which determines the image class of an object. Then, object localization is used to identify the objects from an image and specify the location with a bounding box. Object detection is to detect the objects in a bounding box and identify the classes for the detected objects in these boxes.

Single-Stage detector (SSD) is a standard pre-trained neural network is used as a feature extractor and with the CNN function, it employs the anchor boxes to make predictions on multi-scale feature maps. Meanwhile, You Only Look Once (YOLO) is similar to SSD, it conveys the object detection as a regression problem and sends the input image at once to CNN for end-to-end training. Both SSD and YOLO will output the object detection. YOLO is often preferred for real-time applications due to its inherent high inference speed and accuracy. The recent installments in the YOLO family include YOLOv5, YOLOv6, YOLOv7, and YOLOv8. While YOLOv5 was released in 2020, it still stands out due to its maturity over the newer versions, which are still in the improvement phase (Iftikhar et al., 2023). Hence, YOLOv5 is a good candidate for the face detection model and will be used in this project, as YOLO has a higher recognition speed and accuracy than SSD (Yuan, Du, Liu, Yue, Li and Zhang, 2022).

2.5.1 YOLOv5

You Only Look Once version 5 (YOLOv5) is a well-known object detection algorithm as it has a high speed and accuracy. It is released in 2020 by Glenn Jocher by using the Pytorch framework. It adopted the optimization strategy from the convolutional neural network to outcome the bounding box anchors and data augmentation (Li, Tian, Liu, Liu and Shi, 2022). It uses a single neural

network to detect objects in an image, and it can achieve state-of-the-art results on various benchmark datasets.

The network architecture of YOLOv5 is shown in Figure 2.2. It consists of three parts, which include Backbone: CSPDarknet, Neck: PANe, and Head: Yolo Layer. The CSP backbone allows the network to process larger images more efficiently. The data will first input for important and informative feature extraction in the backbone – CSPDarknet. Then it will be fed to the Neck – PANet for feature fusion. It is to create feature pyramids that support the model to generalize on the object scaling so that it can recognize the same object with various sizes and scales. The feature pyramid is a useful tool to allow the model to operate perfectly on unseen data. Lastly, it will output the detection results in location, size, and score (Xu, Lin, Lu, Cao and Liu, 2021).

One of the advantages of YOLOv5 is its ability to detect objects of different sizes and aspect ratios with high accuracy. It also provides convenient for use and customization. The model can be easily trained on new datasets with minimal changes to the architecture (Xu et al., 2021).

There are five main versions of YOLOv5 architectures, which are YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. It categorizes differently based on the number of layers and parameters. They are also different in inference speed and memory requirements. Thus, resulting in different trade-offs between accuracy and speed. Selecting the versions depends on several factors such as dataset size, speed requirements, accuracy requirements as well as hardware resources (Al-Smadi et al., 2023).

The performance of a face detection model is highly dependent on the quality of the training dataset. Hence, the training dataset should resemble the deployment environment as closely as possible. There have been multiple face detection benchmark datasets proposed over the past decades. Some standard face detection datasets include PASCAL Face, Wider Face, and VGGFace2, arranged following the dataset size in ascending order (Chi et al., 2019). These datasets usually contain a wide array of images with large variations of pose, age, and illumination. Meanwhile, Multi-Attribute Labelled Faces (MALF) is the first face detection dataset with other annotations such as pitch and roll, facial attributes, gender, is-Wearing-Glasses, and is-occluded (Yang et al., 2015). On the other hand, Unconstrained Face Detection Dataset (UFDD) is a

dataset specifically curated for face detection in unconstrained environments (Nada et al., 2018). Thus, it is the preferred dataset to train face detection model.

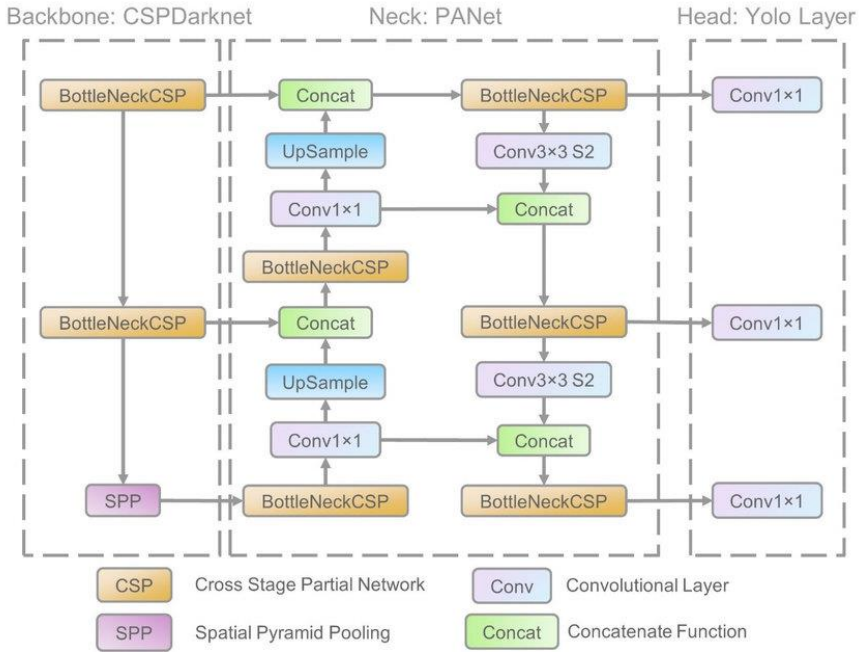

Figure 2.3: Network Architecture of YOLOv5 (Xu, Lin, Lu, Cao and Liu, 2021).

2.5.2 Comparing YOLOv5 and latest YOLOv8 for Multi-Camera Face Detection

YOLOv5 and YOLOv8 are both state-of-the-art object detection models developed by the same research group at the University of Washington, but they have some key differences that may affect their performance in multi-camera tracking applications. First, YOLOv8 is the latest model released in 2022, while YOLOv5 having been released in 2020. YOLOv8 features some enhancements over YOLOv5, including faster processing, more precise localisation, and enhanced performance on small objects. YOLOv8 is still relatively new and may not have undergone as many trials as YOLOv5 (Zhang et al., 2023).

The YOLOv5 is known for its speed and efficiency, which can be beneficial in multi-camera tracking applications where real-time performance is important. YOLOv8 is also fast, but it may not be as optimized for speed as YOLOv5 (Iftikhar et al., 2023). The tracking algorithm – DeepSORT requires a high frame rate to perform accurate object tracking, and thus YOLOv5 can

provide the necessary object detections at a faster rate than YOLOv8. Moreover, the model's size and complexity can affect how well it performs on various hardware platforms. Since YOLOv8 is a more complex model than YOLOv5, it might be harder to run it on less powerful hardware or in contexts with limited resources (Fabregat, 2023).

The YOLOv5's ability to precisely detect smaller objects makes it function well with DeepSORT. When compared to YOLOv8, YOLOv5 uses a smaller anchor box, making it more accurate at detecting smaller objects. This can be beneficial in applications for object tracking where the target object may be small or far away from the camera. Moreover, YOLOv5 may be best adapted for object tracking tasks than YOLOv8 due to its architecture and training methodology. In comparison to YOLOv8, YOLOv5 was trained using a larger and more diverse dataset, which may enable it to generalise to new and untested data more effectively (Sun, Wang and Xie, 2023).

Apart from that, the implementation of the multi-camera setup will be in an unconstrained environment, thus low light environment would need to be considered. YOLOv5 has shown superior performance in low-light conditions may be an advantage over YOLOv8. YOLOv5 uses a novel architecture called "Swish" activation, which is more robust to noise and low signal-to-noise ratio (SNR) conditions compared to the activation function used in YOLOv8 (Medak, 2022). The swish activation function uses the sigmoid function with a linear function as the multiplication factor, which produces more stable gradients during backpropagation.

Besides that, YOLOv5's training procedure includes data augmentation techniques created especially to enhance performance in low light environments (Wang, Chen, Dong and Gao, 2022). For instance, to simulate various lighting situations, the model is trained on images that have had their brightness, contrast, and exposure randomly altered. Although YOLOv8 may still function well in low-light situations, YOLOv5 may be a more advantageous option for face detection tasks in these settings due to its higher performance in this area.

2.6 Face recognition

Face recognition is a biometric technique that uses a person's face to identify or verify them. It has developed into a major tool in a variety of fields, including

security, surveillance, and law enforcement. The advancement in artificial intelligence (AI) has led to the development of more efficient and accurate face recognition models, which are being used in a wide range of applications.

Face recognition technique has evolved from traditional methods such as hand-crafted feature-based to the recent end-to-end trainable deep learning method (Fatihah, Ariyanto, Latipah and Pangestuty, 2018). Deep face recognition methods can be categorized into pair-based and triplet-based methods. Both methods learn by maximizing the similarity and dissimilarity of positive and negative pairs (Song and Ji, 2022). However, pair-based methods use either positive or negative pairs in one shot, while triplet-based methods utilize both simultaneously (Schroff, Kalenichenko and Philin, 2015). There is no clear winner between the two methods, and the research community is actively updating both.

The face recognition works by identifying or verifying the identity of a person by analyzing and comparing their facial features after being detected. This biometric identification is known as a non-intrusive method, where it brings convenience to people as compared to other biometric methods, such as fingerprint or iris recognition (Rusia and Singh, 2023). It is suitable to be implemented in public places, as it can be performed at a distance without the need for physical contact with the person being identified.

There are various AI models for face recognition which are FaceNet, SphereFace, DeepFace, Siamese Neural Network, and VGG Face. All these models are able to perform identification and verification tasks. They require a large amount of training data to optimize their performance. They are mainly adopting deep learning models that use CNN to extract features from face images and map them to a high-dimensional space.

SphereFace is a deep learning model that uses a sphere-like embedding space to improve the discriminative power of the features. It maps the input face images to a hypersphere and computes the angular distance between the face features to perform recognition (Liu et al., 2017). It is less sensitive to variations in lighting, pose, and expression than other models and achieves state-of-the-art performance on several face recognition benchmarks.

In contrast, Siamese Neural Network is another model which consists of two identical neural networks that share weights. It compares two face images

and produces a similarity score between them (Koch, Zemel and Salakhutdinov, 2015). It is computationally efficient compared to other models and can perform face recognition with only a few training examples. However, it may not perform as well as other models on large-scale face recognition tasks and is less robust to variations in lighting, pose, and expression than other models (Koch, Zemel and Salakhutdinov, 2015).

SphereFace model is chosen for this project due to its accuracy in identifying and verifying faces and robustness in handling variations in lighting, pose, and expression, as well as occlusions and other factors that may affect the face image. Other than that, the training data requirements are also taken into consideration, it is suitable to train under the SphereFace model and able to achieve optimal performance.

The face recognition model will be integrated with the face detection model that has been trained. Thus, the detected face will be processed by identifying the face of the person and stored in databases for tracking across multiple cameras. The main challenge in multi-camera face recognition is to ensure that the captured images are of good quality and can be used for recognition.

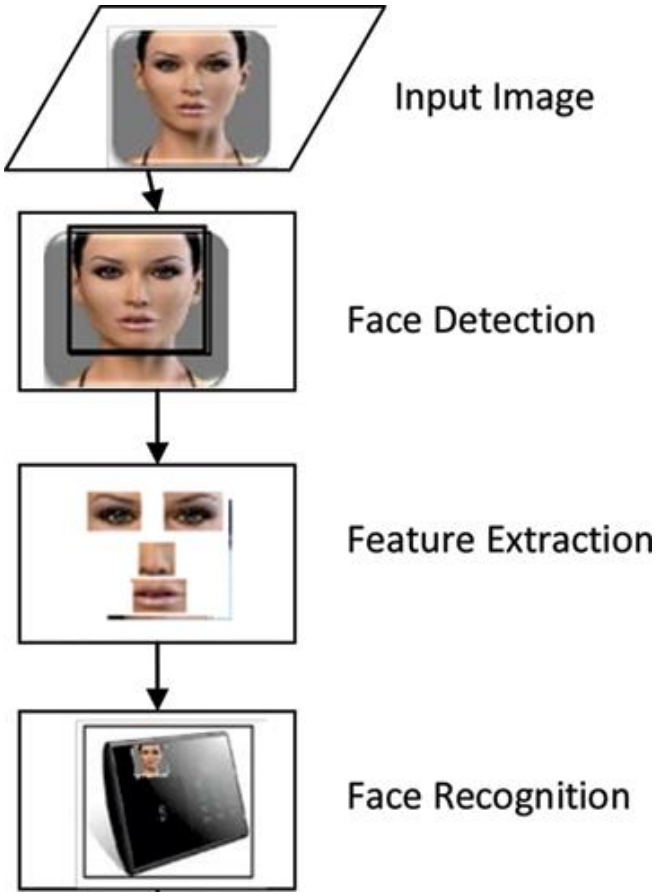

Figure 2.4: Working Principle of Face Detection and Recognition (Sundaram and Mani, 2016).

2.6.1 OpenSphere – SphereFace, SphereFace-R, SphereFace2

OpenSphere is a hyperspherical face recognition library based on PyTorch. It offers a unified and consistent platform for training and evaluation for hyperspherical face recognition research. With the help of the framework, the loss function is decoupled from the other variable components such as network architecture, optimizer, and data augmentation (Liu, Wen, Raj, Singh and Weller, 2022). It can serve as a transparent platform to reproduce published results as it can properly evaluate various loss functions in hyperspherical face recognition on well-known benchmarks.

Like OpenSphere; SphereFace, SphereFace Revived (SphereFace-R) and SphereFace2 use a spherical softmax function to map the output of the model onto a unit hypersphere. However, they are different deep learning models for face recognition that use a combination of a softmax function, and a specialized loss function called the angular softmax loss (Liu et al., 2022). The

loss function adopted by SphereFace is angular softmax loss which encourages the model to learn highly discriminative features for face recognition while SphereFace-R and SphereFace2 are both in modified angular softmax loss that further improves the discriminative power of the learned features.

There are two network architectures in these models SFNet20 and SFNet64. SFNet20 is a relatively shallow network that consists of 20 convolutional layers, followed by two fully connected layers. The input image is first passed through a set of convolutional and pooling layers to extract low-level features, which are then progressively combined and refined by deeper convolutional layers. The final output embedding is obtained by passing the features through two fully connected layers (Liu et al., 2022). As for SFNet64, it is a much deeper network that consists of 64 convolutional layers, followed by three fully connected layers. The network is similar to SFNet20 but with more layers and more complex architectures. The deeper network is capable of capturing more complex and high-level features, which can lead to better recognition accuracy (Liu, Wen, Yu, Li, Raj and Song, 2017).

As for data augmentation, the SphereFace model uses several data augmentation techniques during training to improve the robustness of the model to variations in the input data. Specifically, the model randomly crops and horizontally flips the input image and also applies random brightness and contrast adjustments (Liu et al., 2017). While SphereFace-R and SphereFace 2, are using the same data augmentation techniques as SphereFace, with the addition of random rotation and scaling of the input image. The model also uses a dynamic sampling strategy that adapts the size of the input image to the size of the detected face in the input image (Guo and Zhang, 2019).

There are multiple public FR benchmark datasets. Some notable datasets include Labeled Faces in the Wild (LFW) and MS-Celeb-1M. However, these datasets are often crawled from websites that do not represent real-world surveillance images. On the other hand, the QMUL-SurvFace dataset is a large-scale surveillance dataset with 463,507 face images of 15,573 distinct identities captured in real-world uncooperative surveillance scenes over a wide space and time (Cheng, Zhu and Gong, 2018). Each subject has 4 to 6 images taken under different conditions, such as lighting, poses, and expressions. This is the ideal dataset that fits our goal of recognizing faces in unconstrained surveillance

environments. Thus, SurvFace dataset will be used to train the face recognition model and to be adopted to the pipeline system.

2.6.2 Person re-identification

Person re-identification (Re-ID) is a feature to find the individual based on its appearance from the snapshot. This implies that an image of a person of interest acts as a query to find more images from gallery images that show the same identity. In the multi-camera tracking of individuals, single camera tracks may diverge because of occlusions or individuals leaving an area and reappearing later (Kohl, Specker, Schumann and Beverer, 2020). This will cause the appearances of the individual from completed tracks applied to compare with the latest ones to reallocate the identities of the person.

The Re-ID in this project works in a way where the images are cropped based on the detected bounding boxes and fed to its embedding network to extract appearance features. With that, the features will then connect back to individual bounding boxes in order to form tracks in a multi-camera setting. Re-ID system gets information from a database of more shots listing individuals captured by separate cameras at separate and assigns ID to the individual (Ristani and Tomasi, 2018). If the individual is matched, the ID will be matched back to the previously assigned ID. Meanwhile, if the individual is not matched with the ID, a new ID will be assigned to the individual.

There are several challenges of this Re-ID implementation such as the query and the person in the search space have different views due to different angles and distances between the camera and the persons seen by those cameras (Leng, Ye and Tian, 2019). Besides, the query is captured in a low frame rate which is common in many open-space CCTV video recordings, and has occlusions where the query is visible only in a certain period. The performance can be improved by using sophisticated training methods on a single labeled dataset (Leng, Ye and Tian, 2019).

2.7 Single Camera Tracking

The goal of single camera tracking is to locate and track an object of interest as it moves through the video sequence. The issue of tracking by face detection can be solved by data association problems with detected bounding boxes across

the video sequence in a single camera track. The data will be fed into DeepSORT in the form of data association. Deep SORT is known as Simple Online and Realtime Tracking with a Deep Association Metric, it is an object tracking framework for AI applications to improve tracking performance.

In this project, the output of the single camera tracker is used as an input to DeepSORT during the integration of single camera tracking. An initial estimate of the object's position and velocity is provided by the single camera tracker, which is then improved by DeepSORT using its deep learning algorithms. It has been demonstrated that combining DeepSORT with single camera tracking increases the tracking's durability and accuracy in a variety of applications (Yang, Ge, Yang, Tong and Su, 2022). There are three metrics used in the single camera tracking which comprised of Kalman Filter, Intersection over Unions (IoU), and Hungarian Algorithm.

Kalman Filter is an algorithm that is able to predict the future position of a particular object based on its current position (Li et al., 2015). In this project, it acts as a form of cost matrix, which is used to predict the location of the individual in the existing frame and compare it with detected bounding boxes. If the distances between detected bounding boxes and predicted positions go beyond the threshold, it indicates that detections are not associated with the existing tracks.

IoU of the bounding boxes is also used to compute a cost matrix. It is an evaluation metric to measure the intersection ratio between the area of the bounding box from the last frame of the detected individual and the area of a newly detected bounding box in the current frame. It also comes with a threshold, where the cost will be disregarded when it exceeds it.

The Hungarian Algorithm is implemented to link the detected boxes to tracks after the cost matrix is computed from Kalman Filter and IoU (Hou, Wang and Chau, 2019). It is able to solve the data association problems. It creates unique identities and breaks according to the threshold from the cost matrix. For instance, if the IoU of detection and target is less than the threshold, it will signify as an untracked object. The purpose of this technique is to maintain the IDs of the individual and solve the occlusion problem. Therefore, DeepSORT is able to perform a good track association mechanism for single

camera track which allows the implementation of multi-camera tracking smoothly later.

2.7.1 DeepSORT

DeepSORT has achieved its state-of-the-art algorithm for object tracking in video sequences. It is an extension of the SORT (Simple Online and Realtime Tracking) algorithm, which uses the Kalman filter and the Hungarian algorithm for object tracking. Deep learning models for object detection and re-identification are incorporated into DeepSORT to improve the tracking accuracy and robustness of the SORT algorithm.

The deep learning models used by DeepSORT are trained on big datasets including the Market-1501 dataset and the DukeMTMC-reID dataset, which each contain thousands of images of humans in a wide range of positions, settings, and lighting (Veeramani, Raymond and Chanda, 2018). DeepSORT is able to apply effectively to a range of tracking circumstances due to its training on such datasets.

The architecture of DeepSORT consists of several components which are detection, the object is detected by any state-of-the-art object detection algorithm, such as Faster R-CNN, YOLO, or SSD. In this project, YOLOv5 is adopted for detection. Then, feature extraction, to extract the features that describe the objects' appearance from the detected object with a deep neural network such as a convolutional neural network (CNN). The use of deep learning models for feature extraction and re-identification makes the DeepSORT algorithm more effective in tracking objects with varying appearances over time.

After extracting the features, re-identification will take place to track an object over multiple frames. DeepSORT uses a metric learning approach to learn a distance metric between object features. This metric is used to match objects across frames, even if the object's appearance changes over time. The last phase of the DeepSORT pipeline is object association. It is to link objects found in various frames, DeepSORT employs the Hungarian algorithm in conjunction with the Kalman filter. The Hungarian algorithm links the detected objects to the projected object tracks, while the Kalman filter forecasts the object's position and velocity. The Kalman filter and Hungarian algorithm

ensures that the tracking is robust and precise even in challenging tracking conditions.

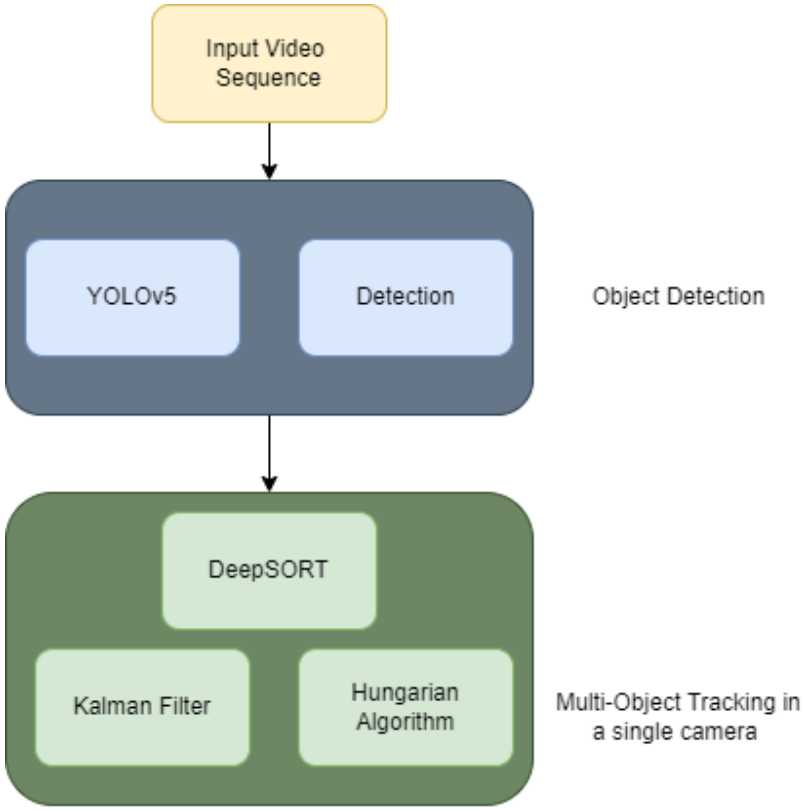

Figure 2.5: Architecture of DeepSORT.

2.8 Multi Camera Tracking

The multi-cameras tracking setup is from the linking of more than one single camera track. It has gained popularity in recent years due to its ability to provide comprehensive coverage of a large area. It involves combining data from multiple cameras to track faces; and can be applied in surveillance and crowd monitoring. The process includes three main steps which are face detection and recognition, data association, and trajectory fusion from single camera tracking.

First, it is the computation of individual bounding boxes with the appearance features from person face detection. Then, the bounding boxes of an individual are redirected to the single camera tracking stage where it computes and yields the tracklets of every camera view individually (Kohl, Specker, Schumann and Beverer, 2020). The output tracklets are then forwarded to track differences by computing various feature distances among the tracks. Last, all tracklets are combined from the weighted aggregation of track distances with the hierarchical clustering approach.

The data association method involves comparing faces found in various cameras. This can be difficult, especially if the cameras are at different angles or if people's faces change. This problem has been addressed using a number of different methods, such as appearance-based matching, geometry-based matching, and data association algorithms.

In trajectory fusion is the process of combining the trajectories of objects across multiple cameras to obtain a complete picture of their movements. This can be achieved with WDA; the WDA includes a total of five individual distances which include the time constraint of single camera and multi-camera, linear prediction discount, homography matching distance, and appearance feature distance (Kohl, Specker, Schumann and Beverer, 2020). With the consideration of these constraints and metrics, it allows the system to form a cluster of an individual tracks through multiple cameras setup.

Meanwhile, the purpose of track data association in hierarchical clustering is to group the tracklets that belong together which means from the same individual. The idea is that every track will initially have its cluster, then two of the clusters that are having shortest distance will be merged until a desired distance is achieved. (Gan, Ma and Wu, 2020).

2.9 Datasets – UFDD & SurvFace

Multi-camera face detection and recognition in unconstrained environments is a challenging task due to variations in lighting conditions, camera angles, and occlusions. To address this challenge, researchers have developed a range of datasets for training and evaluating face detection and recognition models.

The Unconstrained Face Detection Dataset (UFDD) is one such dataset that has been widely used in recent years. UFDD consists of 6,425 photos and 10,897 face annotations with major degradations and conditions in an unconstrained environment, making it a valuable resource for training and evaluating face detection models (Nada, et al., 2018). UFDD includes annotations for face bounding boxes, pose, and occlusion, allowing researchers to evaluate the performance of their models in real-world scenarios.

In addition to face detection, the ability to recognize faces across multiple camera viewpoints is another crucial component of multi-camera face recognition. The SurvFace dataset is a useful resource for this task, as it consists

of over 463,507 face images of 15,573 distinct identities captured in real-world uncooperative surveillance scenes over a wide space and time (Fang et al., 2020). Each subject has 4 to 6 images taken under different conditions, such as different lighting, poses, and expressions. The dataset was created by capturing images from video frames and manually annotating them with the subject's identity, pose, expression, and lighting information (Du et al., 2020). It is a useful tool for developing and evaluating face recognition models that can adapt to changes in camera viewpoint as it contains annotations for the face bounding boxes and the camera viewpoint for each image.

This dataset has been used in numerous research to assess how well different face detection and recognition methods perform. For instance, one study achieved high accuracy rates in detecting faces across various camera viewpoints using UFDD to assess the effectiveness of a YOLOv5 face detection model (Zhao and Qin, 2023). Another study used the SurvFace dataset to train a SphereFace model, achieving state-of-the-art performance in cross-view face recognition (Cheng et al., 2020). Thus, these datasets offer a useful standard for comparing the effectiveness of various models and can promote advancement in this challenging field.

2.10 Multi camera setup

The multi-camera setup is crucial so that the system is able to detect and recognize the faces. The hardware configuration has to function well, it includes cameras, computers, and other peripherals. The cameras used in multi-camera systems need to be synchronized to capture images or video frames simultaneously. High-resolution cameras are needed to take clear pictures of faces.

The position of the camera’s setup is important. In order to prevent redundant data collection, the camera fields of view must be positioned and oriented in a way that maximises the coverage of the area to be monitored (Unterberger et al., 2019). The optimal positioning of the cameras depends on the specific application and environment. For instance, in a stadium, cameras may be positioned at key points around the perimeter to record crowd activity, similar to how cameras in a retail store may be fixed on the ceiling to cover the

entire floor space. The number of cameras required for a particular application also depends on the size and complexity of the area to be monitored.

Apart from that, the accuracy of face detection and recognition is greatly influenced by lighting conditions. Shadows, glare, and other lighting abnormalities must be kept to a minimum so that face detection and recognition algorithms can operate as accurately as possible (Muller, Fregin and Dietmayer, 2017). In order to enhance the visibility of faces in low light, cameras may also be fitted with infrared illuminators. To ensure that the photos or video frames captured by each camera are aligned and have the same scale, the cameras must be calibrated (Kettnaker and Zabih, 1999). This is essential for accurate tracking of individuals across multiple cameras.

A central computer that processes the picture or video frames in real-time must be connected to the cameras. To perform the intensive computations needed for face detection and recognition, the computer must have a strong processor and graphics card. Also, the computer needs to have enough memory to retain the picture or video frames together with the associated facial recognition information.

Multi-camera face detection and recognition systems also need peripherals including power supply, network hardware, and storage devices in addition to the cameras and computers. The photos or video frames and the associated facial recognition data must be kept in storage devices (Unterberger et al., 2019). Network equipment is also required to link the computers and cameras together and enable remote access to the system.

Overall, the hardware setup of a multi-camera face detection and recognition system requires careful consideration of various factors, including the number and positioning of cameras, camera calibration, lighting conditions, and other environmental factors. A proper design and configured system allow accurate and reliable face detection and recognition across multiple cameras which is able to apply in surveillance, security, and other applications.

2.11 Privacy concern

The multi-camera face detection and recognition system has brought convenience to surveillance and security application. However, this technology has caused severe privacy concerns due to the potential for misuse and abuse of personal data. Multi-camera face detection and recognition systems can collect and process large amounts of personal data, including facial images, biometric data, and tracking information. This data can be used to identify and track individuals, monitor their movements and activities, and even predict their behavior. Such use of personal data creates major privacy concerns, particularly in unconstrained environments where individuals are unaware of being monitored.

The possibility for misuse and abuse of personal data is one of the main privacy concerns with multi-camera face detection and recognition. This may involve identity theft, data breaches, and unlawful access to personal information. Concerns regarding civil liberties and the potential for discriminatory targeting of people based on their race, ethnicity, and other characteristics are also raised by the use of this technology in law enforcement and surveillance (Dietlmeier, Antony, McGuinness and O'Connor, 2021).

Another privacy concern of multi-camera face detection and recognition is the potential for false positives and false negatives. False positives happen when a system recognises someone who is not who they claim to be, while false negatives happen when a system misses someone who is indeed a match. These mistakes may have detrimental effects, such as harassment, wrongful arrest, and reputational harm.

Moreover, another privacy risk is the potential for the re-identification of people through the cross-linking of various data sources (Ryan, Dietlmeier, O'Connor and McGuiness, 2022). Even if the original data is anonymised, it may still be possible to re-identify a person by connecting it to other data sources like social media accounts, credit card records, or other publicly accessible information. Therefore, to address these concerns, it is crucial to have transparent policies and regulations controlling the use of this technology. Besides, it also requires the public to fully aware of the data being obtained and how it is being used.

2.12 Summary

The overall multi-camera tracking setup from convolutional neural network for face detection and recognition to single camera to multi-camera tracking has been reviewed. It begins with the current face detection and recognition technology, specifically for multi-camera setup, CNN models in face detection and recognition, as well as single-camera and multi-camera tracking. Several metrics have been discussed and constraints have been reviewed in order to optimize the tracking performance. It also covers ethical and privacy concerns associated with this technology.

The literature review also provides an overview of the AI models such as YOLOv5 for face detection, and SphereFace for face recognition. It also covers the human face detection system that relies on the accuracy of detection to assign an identity to the individual and track through multiple cameras with feature extractions from the SphereFace model. It also concludes on how this technology can be applied in various fields and applications while addressing ethical and privacy concerns.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter discusses the methodology used in this study and the work plan from the software to implement the project with the required hardware. The flow will start with a structured pipeline, software and hardware, dataset exploration, models training for face detection and recognition, face tracking algorithm, and finally, system implementation.

3.2 Work Plan

The input for this project is videos that are being processed from frame by frame. The frames from each camera will then be passed to the CNN network which consists of face and person detection, and feature extraction. The model being used is the face detection model – YOLOv5. It will first be trained with a similar unconstrained dataset from an external source before being implemented to the system.

Then, similarly, the face recognition model will be trained with an external dataset before being fed with the actual project dataset. This allows the models to be identified and recognized the faces more accurately to meet the project's objectives.

After the face detection and recognition are completed, tracking will take place from single camera tracking with DeepSORT. The track distance will be computed with weighted distance aggregation for multiple distances. Last, the multi-camera tracks are formed with hierarchical clustering. Figure 3.1 shows the pipeline of the entire project system from face detection, face recognition, single camera face tracking, and multi-camera face tracking.

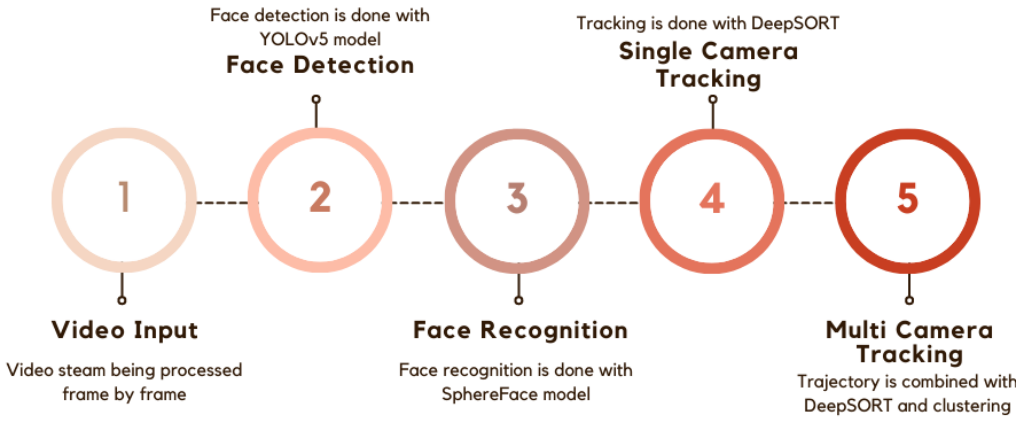

Figure 3.1: Pipeline for the Multi-Camera Face Detection and Recognition System.

3.2.1 Software

The software such as Ubuntu 20.04.4 and Python 3 are being applied to this project. Ubuntu 20.04.4 provides a better environment in the system application as it consists of various tools and libraries that are related to artificial intelligence, machine learning, and deep learning. The framework such as TensorFlow, Keras, and OpenCV are also supported by Ubuntu. Ubuntu also has great stability, continuous updates, and security that provides better support for the user during the system implementation.

The programming language that will be used in this project is mainly Python 3. This is because Python 3 supports most of the applications for artificial intelligence, machine learning, and deep learning. As artificial intelligence and machine learning require the most complex algorithm, the Python environment provides ease for the developers to code. Python also has a higher speed of execution especially in deep learning applications where training sessions are expected to be long. Furthermore, there are numerous library system such as TensorFlow, NumPy, and Keras which provides an advantage in learning application to process the necessary data.

3.2.2 Hardware

The hardware devices that going to be used in this project are cameras and external Graphics Processing Unit (eGPU). As this is a multi-camera setup, thus there will be at least two cameras needed to detect the individual and track the trajectory of it. The chosen cameras would require a higher resolution to allow the system able to detect the faces of individuals across the targeted area. There are a total of two cameras being used to record the actual dataset. The cameras used are Full HD Machine Vision USB 2.0 Camera from UP AI Edge. The resolution is full HD 1920 x 1080, with a picture format of MJPEG / YUV2 (YUYV) and supported with 30 and 60 fps.

Figure 3.2: UP AI Edge – Full HD Machine Vision USB 2.0 Camera.

Figure 3.3: Setup of UP AI Edge – Full HD Machine Vision USB 2.0 Camera.

The training for all models is done on NVIDIA GeForce GTX 1080 Ti Graphic cards. The configuration of the training platform is detailed in Table 3.1.

Table 3.1: Configuration of Training Platform.

Name	Configuration
Operating System	Ubuntu 20.04
CPU	Intel Core i3-7100 CPU @ 3.90GHz
RAM	16 GB
GPU	NVIDIA GeForce GTX 1080 Ti
GPU Acceleration Library	CUDA 11.4, cuDNN 8.2.4

After a series of training and setting up the pipeline, the files will then transfer to a mini pc – Intel NUC BXNUC10i7FNH. It comes with Intel Core i7-10710U processor and two DDR4 SO-DIMM slots that support up to 64GB of memory. With its flexibility and high performance, it serves as a device to run real-time tracking with cameras setup that are connected to it.

Figure 3.4: Intel NUC BXNUC10i7FNH.

3.3 Methodology

The input video streams will be processed and form an output in the form of rectangular bounding boxes which specify the locations of the detected faces of the individuals. This is done with the YOLOv5 face detection model. Then, the face recognition model – SphereFace will be implemented to recognize and look for the individual based on their appearance from the snapshot by frames. This will result in tracking the individual on its trajectory in one single camera with

the DeepSORT model. The complete trajectories will be formed in a multi-camera setup with the Weighted Distance Aggregation method and hierarchical clustering.

There are a total of two models that need to be trained before implementing into the system. These models are the face detection model – YOLOv5 and the face recognition model – SphereFace. External datasets with similar conditions that the project required need to be acquired and feed into the model for training. The performance of the trained models will then be evaluated and integrated into the pipeline.

Then, DeepSORT and wda tracker algorithms need to be implemented for the system to track the faces recorded by the cameras. These algorithms need to be in the same format and able to be integrated along with the models trained. With that, a single pipeline of multi-camera face detection and recognition with tracking across the cameras will be formed.

An actual dataset will then be recorded and tested on the pipeline. There will be two cameras being set, with subjects in an unconstrained environment, to assess the overall performance according to the objectives stated.

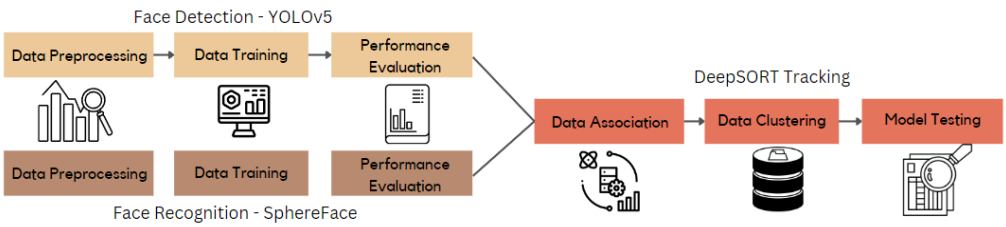

Figure 3.5: Methodology of the Project.

3.4 Datasets

The dataset is a crucial component in order to produce a successful AI model. Training the AI models is essential, and it depends on the quality of the dataset used. A dataset is a collection of data that is used to train machine learning algorithms to perform specific tasks (Mahesh, 2020). It is a key component in the AI model as it gives the model the data it needs to learn and generate accurate and reliable predictions.

Datasets are essential in model training as they provide the necessary information for the model to learn and make accurate predictions. Large volumes of information can be found in datasets, which the AI model can use to learn from and generate predictions from. For instance, a dataset of photos of cats and dogs can be utilised to accurately train an AI model to distinguish between the two species.

The dataset that has been confirmed to be trained in the AI model will need to undergo dataset splitting. It is an important step in training the AI models, the dataset will be split into training, validation, and test sets. The model is trained using the training set, validated using the validation set, and tested against the test set to determine how well it performed. This procedure makes sure the AI model can generalize to new data and does not overfit the training data (Ying, 2019). In this study, most of the datasets used to train the AI model are split into 70% for training and 30% for testing.

Another crucial stage in the training of AI models is annotation. In order to give the AI model more information, annotation entails labeling the dataset. For instance, annotation in an image dataset can entail labeling each image with details about the items in the image. The additional data aids the AI model's understanding of the dataset and aids in the development of more precise predictions. This annotation method has been implemented into the study to annotate the faces of the UFDD datasets in order to train the YOLOv5 model so that the face can be detected even in an unconstrained environment.

There are a number of sources to obtain datasets such as Kaggle, UCI Machine Learning Repository, Google Dataset Search, and some journal papers. These sources offer a wide variety of datasets to train AI models for different applications. Normally, datasets are sorted according to their attributes, such as size, complexity, and format. For example, a set of image data may be sorted according to the resolution, quantity, and type of the photos. The datasets that have been used in this study is mainly from research papers or journal, it is obtained by personally approaching the author. These datasets were carefully evaluated to cross-check whether it is essential in helping the models predict more precisely.

The size, relevance, and quality of the data, as well as the intended application for the AI model, are all important considerations when choosing

the best dataset for AI training (Lee and Shin, 2020). It is essential to choose a dataset that is representative of the problem being solved and that provides enough information for the AI model to learn from. All in all, datasets provide the necessary information for the model to learn and make accurate predictions.

3.4.1 Dataset – MTA

Prior to the actual setup of the system pipeline, the dataset "The MTA Dataset for Multi-Target Multi-Camera Pedestrian Tracking by Weighted Distance Aggregation" by Kohl, et. al (2020) is used for exploration of the multi-camera detection with tracking purposes. It consists of more than 2800 person identities across 6 cameras. The video length is more than 100 minutes per camera in day and night periods. The proposed method for the system implementation is 80% for training and 20% for validation of the training model. It has shown that the dataset works well in the proposed system.

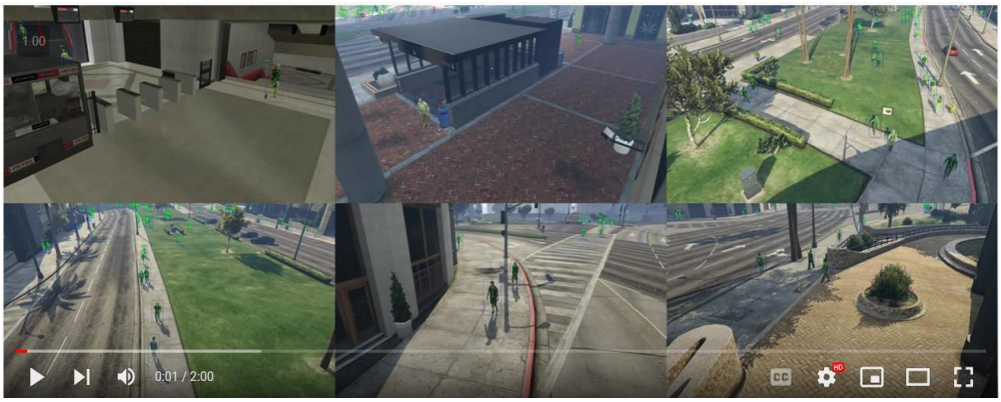

Figure 3.6: Sample of Dataset – MTA Dataset.

3.4.2 Dataset – Unconstrained Face Detection Dataset (UFDD)

The Unconstrained Face Detection Dataset (UFDD) is used to train the YOLOv5 face detection model. This dataset is the best fit for the YOLOv5 model as it includes a collection of 6,425 pictures and 10,897 face annotations with significant flaws or situations like rain, haze, lens obstructions, snow, blur, lighting shifts, and distractions which are matched with the objective – the unconstrained environment. With this dataset being trained, it allows the model to detect faces in an unconstrained environment more precisely. This dataset was formed as the paper found out that the accuracy of face detection reduces when it comes to real-world situations (Nada, et al., 2018).

Figure 3.7: Sample of UFDD Dataset (Blur).

Figure 3.8: Sample of UFDD Dataset (Illumination).

Figure 3.9: Sample of UFDD Dataset (Motion).

3.4.3 Dataset – SurvFace

The SurvFace dataset is a dataset of facial images that is useful in training face recognition models such as the SphereFace model. This is because it contains a large and diverse dataset that contains images of subjects with different ethnicities, ages, and genders, which helps the model to learn to recognize faces accurately and generalize well to new data (Fang et al., 2020). The dataset also includes pictures shot in various lighting situations, which can assist the model in learning to detect faces in various lighting situations. With that, this dataset is able to match this study's objectives by recognizing the faces in unconstrained environments with different postures and lighting conditions.

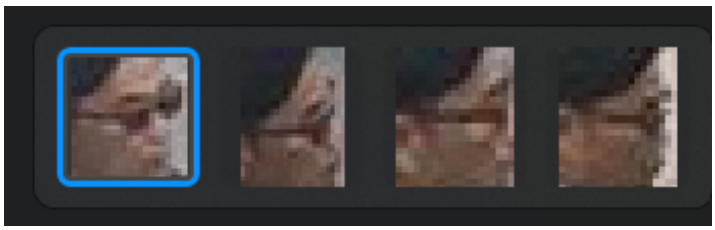

Figure 3.10: Sample of SurvFace dataset with a series of cropped images of the subject in different poses and expression.

3.5 Face Detection - YOLOv5

Prior to the training of the YOLOv5 model, the dataset is chosen which is the UFDD dataset as stated. The data is first pre-processed to ensure that all images are of the same size and format. The dataset was also split into 70% for training and 30% for validation and testing to prevent overfitting. Then, the YOLOv5 model is installed from the official documentation from GitHub with the required dependencies.

The YOLOv5 model is then configured for training with the UFDD dataset. This involves modifying the configuration file to specify the number of classes, the input image size, and other parameters. The location of the dataset and the location of the configuration file have been specified and start training the model with the train.py script provided. After the model has been trained, the performance of the model is evaluated with the validation set to see how well the model is performing and whether it is overfitting or underfitting. The model is then fine-tuned by using a hyperparameter with initializing the model with weights from a pre-trained model. Finally, the trained and fine-tuned model

will be tested with a pre-recorded video to evaluate how well it performs in real-world scenarios.

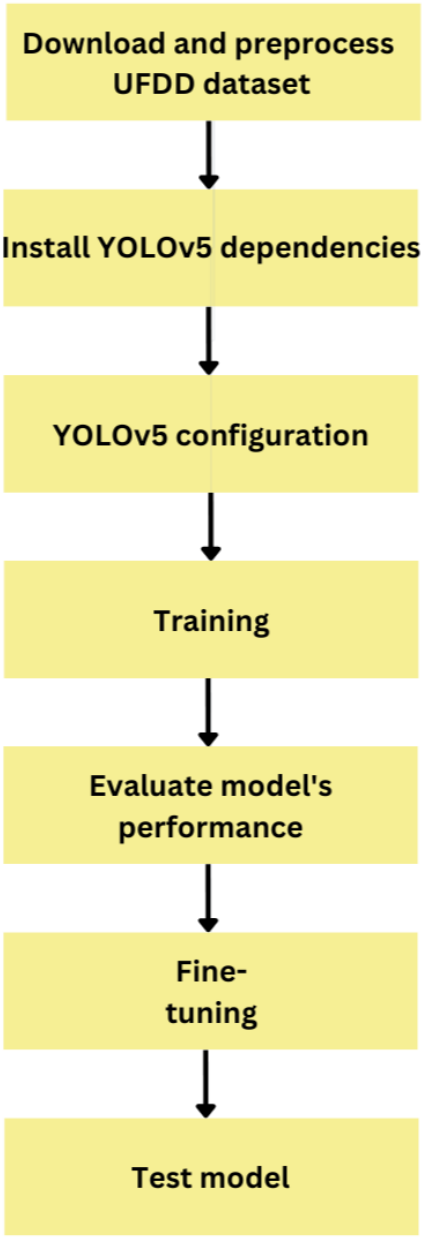

Figure 3.11: YOLOv5 flowchart.

3.5.1 Training of YOLOv5 model

The training was done multiple times with different versions, batch sizes, and epochs. This is to observe which training arises the ideal result that is able to implement in the system. There were a few options for the versions used which are YOLOv5x and YOLOv5n. The difference between these versions is the network architecture and performance. The batch size and epoch were varied

during the training to examine how it affects the model with the changes in these variables.

Generally, a larger batch size can lead to better training accuracy and faster convergence, but it can also increase the risk of overfitting. Meanwhile, the optimal number of epochs depends on the complexity of the dataset and the model architecture. Thus, during the training, it first started with a smaller number of epochs and increases gradually until the model performance on the validation set plateaus or starts to degrade (Zhou, Zhao and Nie, 2021).

The YOLOv5n and YOLOv5x are trained with an initial learning rate of 0.01, which is slowly decayed to 0.001 via cosine decay. The batch size of YOLOv5n is set to the default 64, while YOLOv5x is set to 8 due to resource constraints of the graphic card. The training epoch is set to 300 for both models. The hyperparameters are listed in Table 3.2.

Table 3.2: Configuration of Face Detection Training.

Hyperparameter	***Values***
Initial learning rate	0.01
Learning rate decay	Cosine decay with 0.1 factor
Batch size (YOLOv5n)	64
Batch size (YOLOv5x)	8
Epochs	300

The performance of the model is evaluated with mAP (mean Average Precision). A higher mAP score indicates better performance. It is a useful metric for comparing the performance of different models and for evaluating the impact of changes to the model architecture or training data. Monitoring mAP with time allows to identify when the model performance is plateauing or when it is starting to overfit.

3.6 Face Recognition – OpenSphere SphereFace

The chosen face recognition AI model in this project's system is SphereFace from OpenSphere deep learning library. There are a few versions of SphereFace

models available in the library which include SphereFace, SphereFace2, and SphereFace-R. The dataset that will be trained in the SphereFace models is the SurvFace dataset as mentioned.

First, the SurvFace dataset will be downloaded and pre-processed to ensure all are in the same size and same orientation. Then, the dependency for the model is installed and configuration is done for both the models and datasets. The SphereFace model architecture is defined using the OpenSphere library. This architecture consists of convolutional and fully connected layers that map the input images to a high-dimensional feature space. Besides, other parameters such as model, step, and batch size are also defined to train the model.

After the training is done, the performance of the trained model is then evaluated with the metric which is the accuracy score. The training of the model is repeated for different architecture, model, step, and batch size in order to get the best-trained model that best fit the system. The evaluation is based mainly based on the metrics – accuracy, equal error rate, and area under the ROC curve. The trained model will then be tested on the pre-recorded video to observe how well it works in the actual situation that is able to meet the objectives.

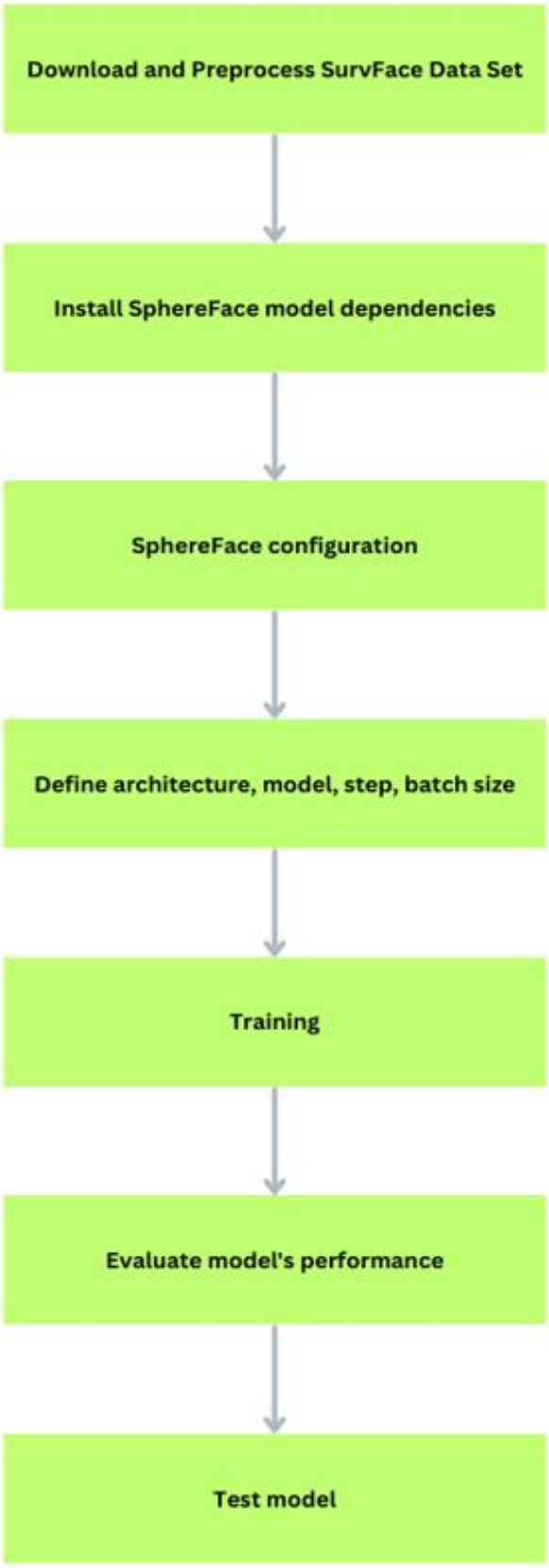

Figure 3.12: SphereFace Flowchart.

.

3.6.1 Training of Face Recognition Model

OpenSphere repository comes with three different face recognition model architectures, which are SFNet20, SFNet64, and IResNet100. SFNet20 and SFNet 64 are custom-designed face recognition models which have 20 and 64 layers, respectively. Meanwhile, IResNet100 is a deep network with 100 layers. Only SFNet20 and SFNet64 are used due to computing constraints.

The SFNet20 and SFNet64 are trained on the SurvFace dataset using all three SphereFace variations for 10,000 training iterations. The batch size is set to be 64 for SFNet20 and SFNet64 to be 512 and 256, respectively. The initial learning rate is set as 0.1 and is decayed by a factor of 0.1 at training iterations 5,000, 7,800, 9,400, and 10,000. All the hyperparameters are listed in Table 3.3.

Table 3.3: Configuration of Face Recognition Training.

Hyperparameter	***Values***
Initial learning rate	0.1
Learning rate decay	Step decay with 0.1 factor at training iteration 5,000, 7,800, 9,400, and 10,000
Batch size (SFNet20)	512
Batch size (SFNet64)	256
Training iterations	10,000

3.7 Tracking

The tracking process in this project is done in single-stage algorithms which means the model will skip the proposing regions of two-stage algorithms and promptly regress the locations of the objects and probabilities of categories in a single stage (Zhao, Huang and Lv, 2022). The single-stage algorithms show a higher inference speed and the examples included YOLO and SSD. The output will show the combination of object classification as well as the locations. YOLOv5 will be implemented in the tracking method over SSD as it shows a higher accuracy and speed (Kim et al., 2020).

In order to form a trajectory, it needs the help from a multi-camera and the data has to be associated. This can be achieved with the DeepSORT algorithm. DeepSORT algorithm is an algorithm known as tracking-by-detection algorithm with appearance features in the tracker. It consists of two major parts which are the object detection algorithm which generates a bounding box and the data association algorithm to associate the data with the latest and previous detected object to develop a trajectory.

The DeepSORT is needed along with the implementation of trained YOLOv5 along with the trained face recognition model SphereFace.The DeepSORT involves two algorithms which are Kalman Filter and the Hungarian algorithm. Kalman Filter is to predict the object detected based on the previous frame with the calculation in a mathematical model. This allows the object detection improves its precision of the predicted position. Meanwhile, the Hungarian algorithm is to allocate the latest detection bounding box to the expected box in the previous frames.

In the single camera tracking, the YOLOv5 model receives input videos or frames from the camera and processes them to detect objects. The result is a series of bounding boxes with corresponding confidence scores for each object that was detected. Along with the features obtained from the SphereFace model for face recognition, the bounding boxes are fed to DeepSORT. DeepSORT predicts the position of the object in the following frame by associating each bounding box with a distinct identity. Then, DeepSORT produces a set of tracked objects along with their corresponding identities and anticipated positions in the following frame.

Meanwhile, in multi-camera tracking, each camera feed is processed separately using the pipeline for single-camera tracking. A unified set of tracked objects is then created by combining the outputs of each camera stream. A global identity assignment is done to assure that the same identity is assigned to the same individual across all cameras in order to prevent conflicts between the identities assigned by various cameras.

DeepSORT algorithm does not require any training process but can be fine-tuned on a smaller dataset to adapt to specific tracking scenarios (Punn, Sonbhadra, Agarwal and Rai, 2020). In this project, no training is done on DeepSORT as most of the datasets used to train the face detection and

recognition models are in a multi-camera setting which allow the DeepSORT to adapt to the scenario.

3.8 Integration of Face Detection, Face Recognition and Tracking

After training the face detection and recognition models, it is essential to integrate these models to achieve the study's objectives. To integrate these three technologies into a single pipeline, the input videos from the cameras are first fed into the trained YOLOv5 face detection algorithm. The spatial location (bounding box) will then be extracted and passed to the Sphereface face recognition model, which extracts features and assigns unique IDs to each face. These IDs are then passed to the DeepSORT tracking algorithm, which tracks the individuals across multiple camera views.

The DeepSORT algorithm will determine if a detected face is part of active tracking. If that is the case, DeepSORT will update the latest location of the given face. Else, DeepSORT will initialize a new tracking to track that particular face, where the identity of this face is identified via the face recognition model. If the face is from an unknown identity, the face recognition model will assign a new identity to it, and the face's feature (extracted by SFNet20/SFNet64) and its identity are kept in the database.

In a single-camera setup, the pipeline would involve face detection, feature extraction, and face recognition, followed by tracking of the individuals using DeepSORT. Meanwhile, in a multi-camera setup, the pipeline would involve clustering the data from multiple cameras based on the IDs assigned by SphereFace and then tracking the individuals across the different camera views. The overall flowchart of the pipeline is shown in Figure 3.13.

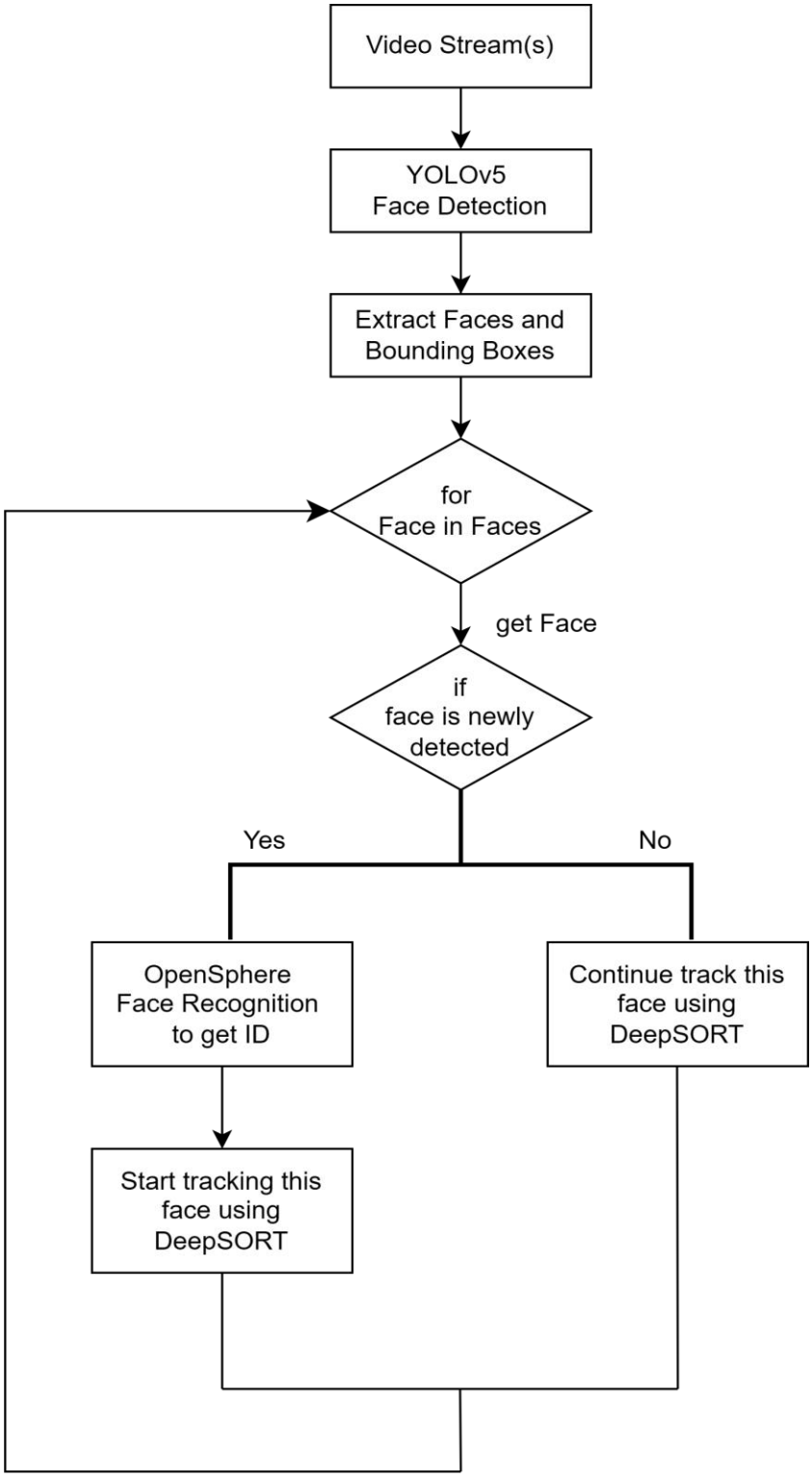

Figure 3.13: Flowchart of the Integrated System.

3.9 Pre-recorded Multi Camera Setup Dataset

A multi-camera setup dataset was recorded in the open environment at the campus compound. There is a total of two cameras set up in a L shape corridor, to detect the subject's face and track the subject from one camera to another. The video is about 4 minutes long with a total of 5 subjects. The subjects are in an unconstrained environment without any intention to show their faces on camera. The purpose of creating this dataset is to test whether the system pipeline is able to work in a real-world scenario.

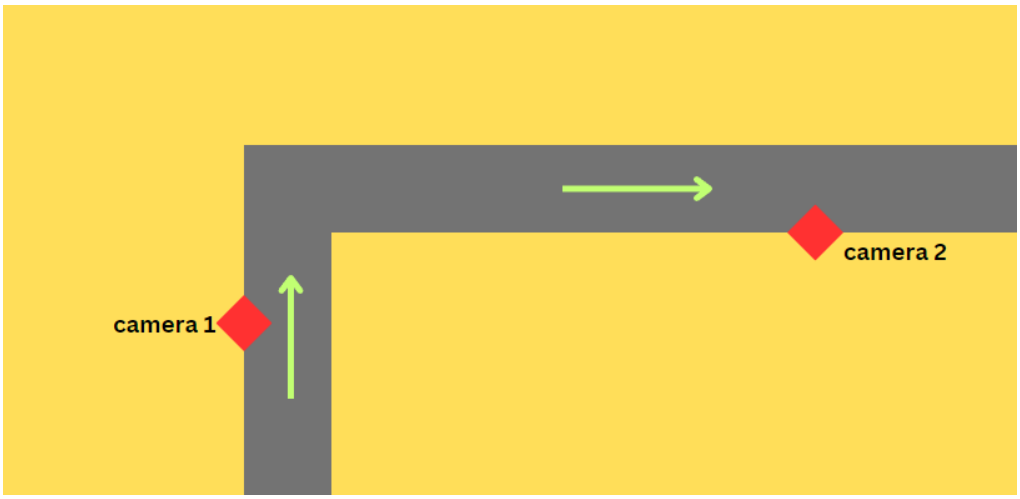

Figure 3.14: Layout of Multi Camera Setup.

Figure 3.15: Sample Frame Captured from Camera 1.

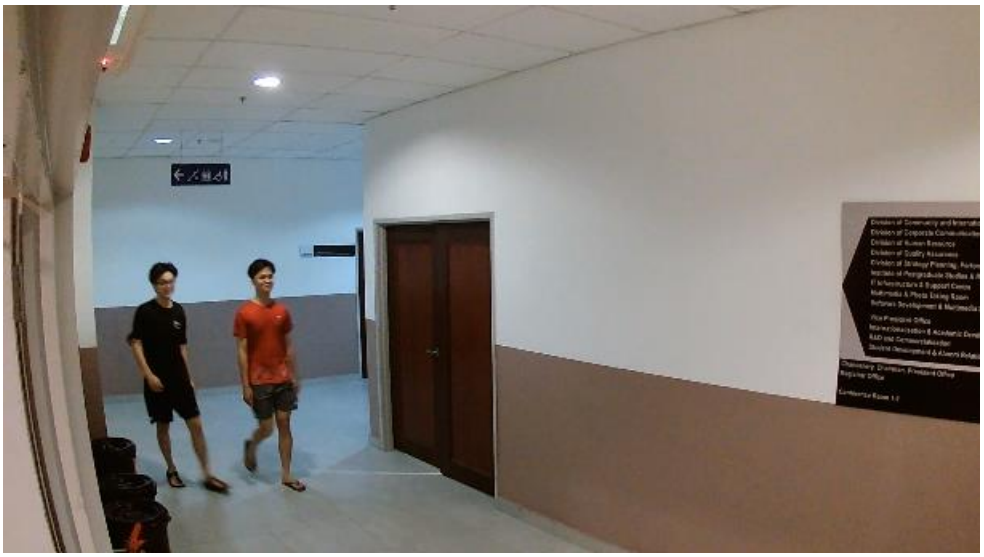

Figure 3.16: Sample Frame Captured from Camera 2.

3.10 Gantt Chart

Timeline: 13th June 2022 to 7th October 2022.

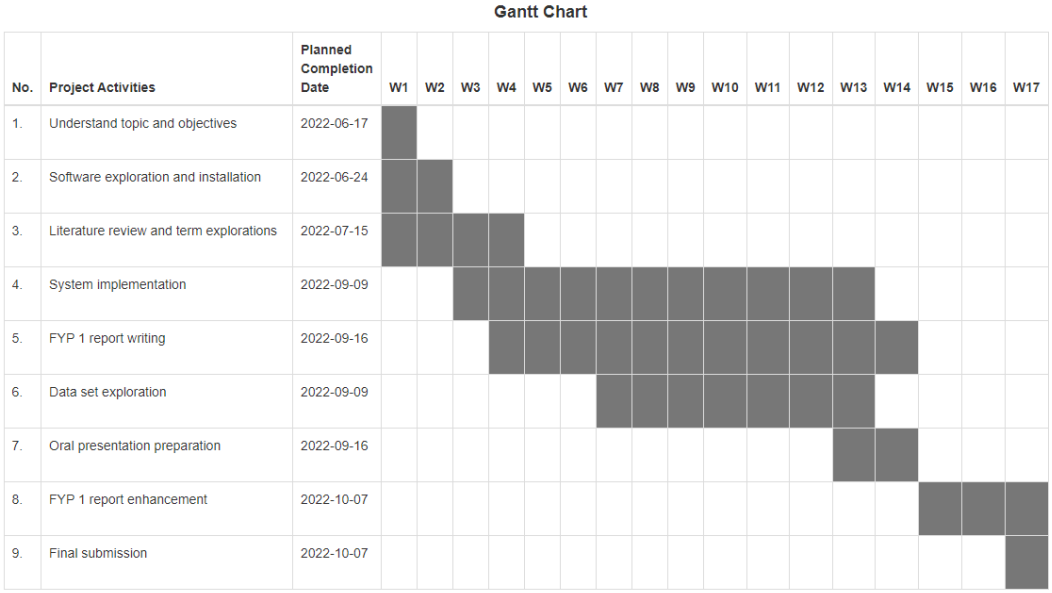

Figure 3.17: Gantt Chart Phase 1.

Timeline: 30th January 2023 to 22nd May 2023.

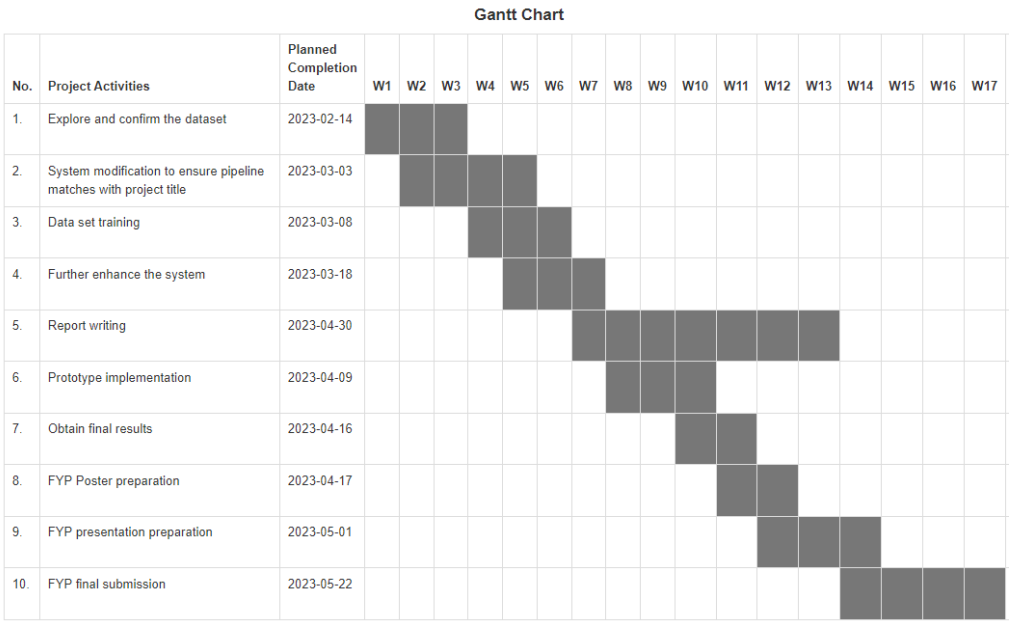

Figure 3.18: Gantt Chart Phase 2.

3.11 Summary

The chapter discusses the methodology used in a project that involves implementing a multi-camera face detection and recognition system. The work plan involves processing videos frame by frame through a CNN network, along with the YOLOv5 face detection model, and feature extraction using the face recognition model – SphereFace which will be trained with external datasets. For single-camera tracking, the output will be tracked using the DeepSORT model, and for multi-camera setups, the trajectories will be created using the Weighted Distance Aggregation approach and hierarchical clustering.

The software used includes Ubuntu 20.04.4, Python 3, TensorFlow, Keras, OpenCV, and OpenVINO, while the hardware devices used are Full HD Machine Vision USB 2.0 Cameras from UP AI Edge and an AORUS RTX 3080 Gaming Box. The methodology used involves processing the input video streams, generating output in the form of rectangular bounding boxes, recognising people based on their facial features, and tracking them in a single camera before generating full trajectories in the multi-camera setup using the Weighted Distance Aggregation method and hierarchical clustering.

The quality of the dataset is an essential factor that determines the success of an AI model. A data set is split into training, validation, and testing sets. Annotation is also important in providing additional information for the model's understanding. The amount, relevance, and quality of the data must be taken into account, as well as the intended use of the AI model, while choosing the optimum dataset. This study used the MTA, UFDD, and SurvFace datasets for exploring multi-camera face detection and recognition with tracking purposes, training YOLOv5 face detection models for unconstrained environments, and training face recognition models such as the SphereFace model, respectively. The training process is discussed and evaluated.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The performance of the trained models such as the YOLOv5 face detection model and SphereFace face recognition model will be discussed and explained on how the evaluation is done. The metrics used to evaluate the YOLOv5 face detection model include mean average precision (mAP), precision, and recall. Meanwhile, for the SphereFace face recognition model, it includes accuracy (ACC), equal error rate (EER), and area under the ROC curve (AUC).

$$mean\ Average\ Precision, mAP\ = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4.1}$$

$$Average\ Precision, AP = \int_0^1 P(R)\, dR \tag{4.2}$$

$$Precision, P = \frac{TP}{(TP+FP)} \tag{4.3}$$

$$Recall, R = \frac{TP}{(TP+FN)} \tag{4.4}$$

$$Accuracy, ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{4.5}$$

$$AUC = \int ROC\ (fpr, tpr)\ d(fpr) \tag{4.6}$$

where,

mAP = mean Average Precision

AP_i = Average Precision of Class i

N = Number of Classes

P = Precision

R = Recall

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

ACC = Accurarcy

AUC = Area Under ROC Curve

ROC = Receiver Operating Characteristic curve

fpr = False Positive Rate

tpr = True Positive Rate

The mean average precision (mAP) measures the model's average precision across various recall levels. If the model achieves a high mAP score, it likely has high precision and recall, which allows it to detect faces in a variety of settings and orientations (Chen, Cao and Wang, 2022). Precision (P) is the metric to measure how accurate the model's positive predictions are, which indicates the model is able to identify faces without many false positive detections. It is the fraction of true positive detections out of all positive predictions made by the model. Recall (R) measures how well the model can detect all the positive samples in the dataset, it is the fraction of true positive detections out of all actual faces in the dataset. Average Precision (AP) can be derived from P and R.

The accuracy (ACC) is used to determine whether the model is able to accurately recognize the faces, it is the percentage of all face shots in the dataset that were correctly recognized. Equal error rate (EER) is the metric that measures the point at which the false positive rate and false negative rate are equal. A model with low EER indicates that it is able to recognize faces with lesser false positives or false negatives. The ROC curve is a plot of the true positive rate vs. false positive rate, and the AUC measures the area under this curve. It indicates the overall performance of the model across a range of threshold values for recognition. A high AUC score shows that the model can correctly recognize faces over a wide range of threshold values.

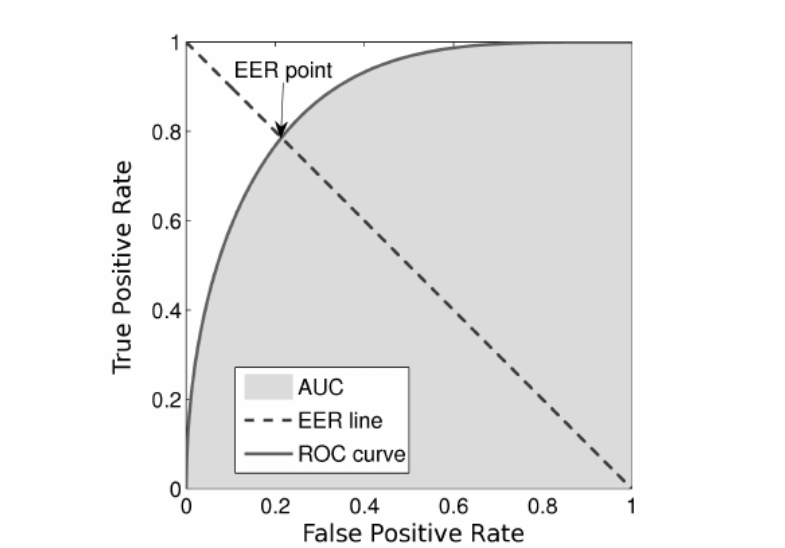

Figure 4.1: ROC curve with AUC and EER shown (Tronci, Giacinto and Roli, 2009).

Evaluation is a crucial part in developing AI models as it allows the researchers to assess the performance and determine the models' suitability in various applications. It also allows the researchers to identify the limitations of models trained, and later to improve and optimize the models based on the parameters such as epochs and batch size. With that, it ensures the models trained are robust and able to apply to real-world scenarios with optimum results achieved.

The performance of the evaluation will be breakdown into three sections, which include the performance of the face detection model, performance of the face recognition model, and performance of the tracking. These performances are evaluated based on the different variables applied. The results will be analyzed and the desired outcome that will be deployed into the system will be chosen based on the analysis.

4.2 Performance of Face Detection Model

A total of 1500 images with annotations from the UFDD dataset are used for YOLOv5 face detection model training. A series of training is done by manipulating the YOLOv5 versions, batch size, and epoch, to observe which output provides the best results that are able to detect the faces accurately. The dataset is able to enhance the capability of the face detection model, and able to detect faces with other datasets that are being used for testing. Table 4.1 shows the metric results with different parameters being applied on the training. The parameters include model architecture (version), batch size, and epoch, while the metrics used to evaluate the training results are mean average precision (mAP), precision (P), and recall (R).

Table 4.1: Metric Results for YOLOv5 Training with Different Parameters.

Exp	Parameters			Metrics		
	Architecture	Batch size	Epoch	mAP	P	R
1	YOLOv5x	8	300	0.473	0.87319	0.70144
2	YOLOv5n	64	100	0.487	**0.89369**	0.74157
3	YOLOv5n	64	300	**0.495**	0.86799	**0.78103**

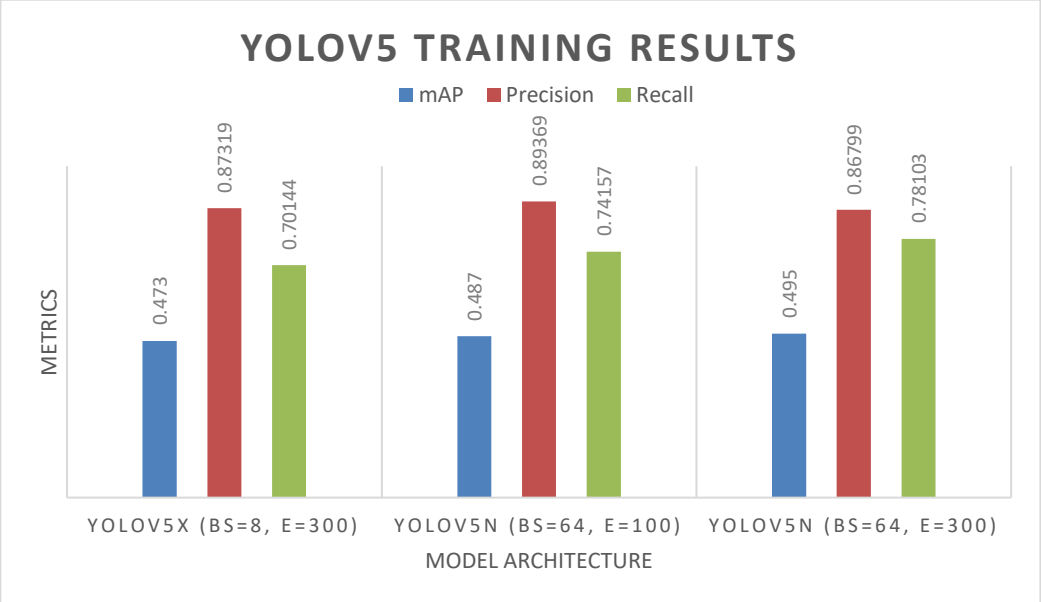

Figure 4.2: Chart of YOLOv5 Training Results (BS: Batch size, E: Epoch).

4.2.1 Analysis on the YOLOv5 Training Results

Three experiments are being carried out in the training process, where these experiments are varied between the hyperparameters and model architecture. The reason of doing so is because it can show a different configuration and look for the optimal combination to achieve the study's objectives. By varying the parameters, it allows the back end to understand how certain variables affect the model's performance. Thus, able to improve performance, prevent overfitting, and reduce training time and memory usage. With that, it can make a more informed decision in future experiments.

The hyperparameters – batch size and epoch have an impact on how a neural network is trained. Epoch refers to the number of times the model depicts the entire training dataset, while the batch size refers to the number of samples processed by the model in one forward/backward pass. The number of epochs affects how many times the model observes the whole dataset during training. A higher number of epochs can result in better performance, but it also may increase the risk of overfitting. As for the batch size, it can have an impact on how stable the training process is; while a higher batch size may speed up convergence, it also increases the chance of overfitting (Zhang et al., 2019). On the other hand, a smaller batch size could require more epochs to get the same level of performance, but it might improve the model's ability to generalise.

mAP is the major metric used to evaluate the performance. According to Table 4.1, experiment 3 with YOLOv5n architecture with a batch size of 64 and 300 epochs, has the highest mAP value of 0.495, indicating that it has a high accuracy in detecting objects overall. The combination of a larger batch size and longer training period (300 epochs) could have caused the model to learn more complex features and improve its performance.

P and R values are fairly important in evaluating the performance as the purpose of the model is used for face detection. The model has to detect small objects accurately, with higher precision and recall values, it measures the ability of a model to correctly identify positive samples (faces) and avoid false positives and false negatives. From Table 4.1, it is shown that experiment 2 has the highest precision value, and experiment 3 has the highest recall value. It is important to note that in evaluating the face detection model, it is essential to achieve the balance between precision and recall. Thus, with the consideration

of both precision and recall values, it can be seen that the recall value in experiment 2 is lower than in experiment 3, despite having a slightly higher precision value. It also indicates that experiment 3 can detect more faces overall, even if it has slightly lower precision.

Experiment 1 uses YOLOv5x architecture as compared to experiment 2 and 3 which use YOLOv5n. It is shown that the YOLOv5n far outweighs the performance of YOLOv5x. This is because YOLOv5n which is a smaller and lighter model is more suitable for detecting small objects such as faces, due to its smaller size and higher processing speed. On the other hand, YOLOv5x is more suitable for detecting larger objects, as it has more parameters and can handle more complex features. Thus it is less computationally expensive for training and inference.

With that, it can be concluded that experiment 3 has the most desired performance out of all experiments. It has the highest mAP value which indicates that it has better overall performance in detecting objects. Besides, it is also better at balancing between precision and recall values, even though its precision value is slightly lower than experiment 2. Thus, resulting in higher overall accuracy. The experiment 3 trained model has also been adapted to other datasets and has shown a desired result with its objective achieved.

4.3 Performance of Face Recognition Model

The SurvFace dataset is used to train the face recognition model – OpenSphere. A series of training is done with different architecture and versions of SphereFace, and the results are tabulated in Table 4.2. The OpenSphere architecture is known as SFNet, there are a total of three variations which include SFNet20, SFNet64, and SFNet64BN. There are varied between the different numbers of layers and batch normalization.

The SFNet20 architecture has 20 layers and is composed of convolutional layers, activation functions, and fully connected layers. Meanwhile, SFNet64 is an extension of SFNet20, with 64 layers. It also includes max-pooling layers that reduce the size of the feature maps and improve the effectiveness of the computational of the model. Similar to SFNet64 design, the SFNet64BN architecture comes with an extra feature which is the batch normalization layers. Batch normalization is a technique used to enhance the

training of deep neural networks by normalizing the input data to each layer. However, only SFNet20 and SFNet64 are used for training the face recognition model, the SFnet64BN is dropped as it does not show an ideal result in this study.

The loss functions for face recognition models include SphereFace, SphereFace-R, and SphereFace2. These loss functions aim to improve the discriminative power of the learned features by explicitly optimizing the angular margin between different face classes (Tan et al., 2022). SphereFace comes with multi-class classification training with angular margin and multiplicative margin. Meanwhile, SphereFace-R unifies all hyperspherical face recognition methods and provides stable training. The latest SphereFace2 supports binary classification training and is robust in labeling noises. Generally, these loss functions are effective in face recognition tasks, improving the discriminative power of the learned features.

All the training is done in 10000 steps. While for the batch size (bs), SFNet20 is set to 512, as suggested by OpenSphere (Liu et al., 2022). As for the bigger SFNet64, the bs is set to 256 due to GPU memory constraints. An ablation study is conducted to show the best combination of the SphereFace version and SFNet model architecture. The evaluation metrics are adopted from OpenSphere (Liu et al., 2022), which include accuracy (ACC), equal-error rate (EER), the area under ROC curve (AUC), and TPR at various FPR (TPR@FPR).

Table 4.2: Performance of each combination of model architecture and SphereFace loss functions.

Archi-tecture	*Loss function*	*step*	*bs*	*ACC (%)*	*EER (%)*	*AUC (%)*	*TPR@FPR (%)*				
							1.00 E-04	5.00 E-04	1.00 E-03	5.00 E-03	5.00 E-02
SFNet20	Sphere Face	10000	512	**82.05**	**19.17**	**88.87**	10.90	17.05	29.93	**46.34**	**66.77**
	Sphere Face-R	10000	512	81.87	19.4	88.73	**15.66**	**20.66**	**34.46**	44.25	66.05
	Sphere Face2	10000	512	81.13	19.87	87.91	8.85	19.79	27.65	37.95	63.08
SFNet64	Sphere Face	10000	256	**81.98**	**18.87**	**89.04**	**8.95**	**23.70**	**27.39**	**41.86**	**67.71**
	Sphere Face-R	10000	256	77.99	22.74	85.17	8.70	16.07	19.23	31.94	55.41
	Sphere Face2	10000	256	81.24	19.64	88.20	3.44	13.90	21.58	33.32	62.58

Bold indicates best performance, while underlined indicates the second-best performance.

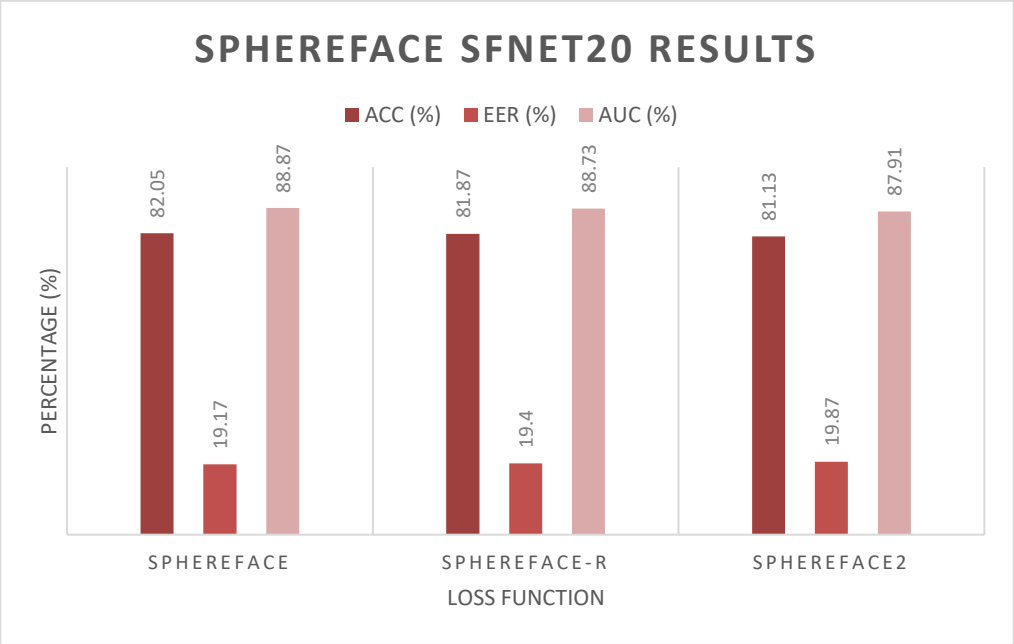

Figure 4.3: Chart of SphereFace SFNet20 Results.

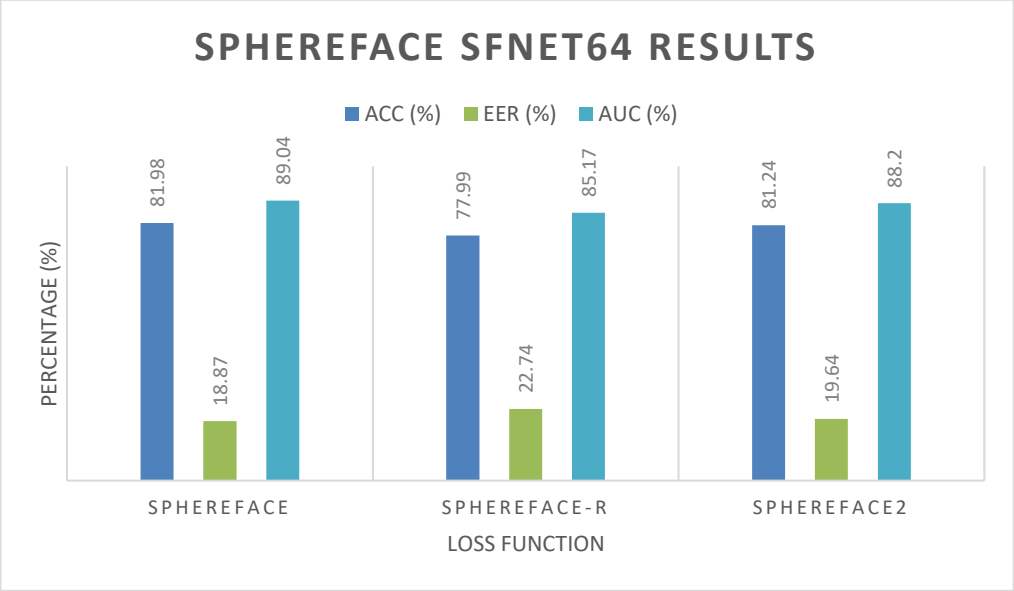

Figure 4.4: Chart of SphereFace SFNet64 Results.

4.3.1 Analysis on the OpenSphere Training Results

SphereFace-R and SphereFace2 are the updated versions of the original SphereFace loss function. Theoretically, the performance of the two updated versions should outperform the original SphereFace. However, it is found that the SphereFace loss function works best with both SFNet20 and SFNet64 for the SurvFace dataset. Both networks achieved their peak performance in all metrics when trained using SphereFace. Hence, we deduced that SphereFace is the best loss function for the SurvFace dataset.

For SFNet20, the SphereFace loss function achieved the highest accuracy of 82.05%, while SphereFace-R and SphereFace2 achieved lower accuracies of 81.87% and 81.13%, respectively. SphereFace also achieved the highest TPR@FPR value of 66.77%, followed by SphereFace-R with 66.05%, and then SphereFace2 with 63.08%. As for SFNet64, SphereFace also achieved the highest accuracy of 81.98% and the highest TPR@FPR value of 67.71%

On the other hand, the performance of SphereFace-R and SphereFace2 is not as consistent. SphereFace-R works better for SFNet20, while SphereFace2 converges faster for SFNet64. Nevertheless, both are still comparatively inferior to SphereFace. Hence, we adopt the SphereFace loss function for both SFNet20 and SFNet64.

There is no clear winner between SFNet20 and SFNet64 in terms of the best overall performance for face recognition on the SurvFace dataset. For

instance, SFNet20 achieved an accuracy of 82.05% and a TPR@FPR value of 66.77% with SphereFace, while SFNet64 achieved an accuracy of 81.98% and a TPR@FPR value of 67.71% with the same loss function. It is supposed to deduce that SFNet20 is better. However, when it comes with SphereFace-R, SFNet20 achieved a higher accuracy of 81.87% and a TPR@FPR value of 66.05%, while SFNet64 achieved a lower accuracy of 77.99% and a lower TPR@FPR value of 55.41%. Thus, both SFNet20 and SFNet64 show promise for face recognition on the SurvFace dataset. At last, SFNet20 is adopted in the system's pipeline as it is more lightweight and has a higher inference rate than SFNet64.

4.4 Performance of Tracking

The pipeline has been formed after both the face detection model and face recognition model are fed into DeepSORT for tracking. It has achieved an FPS of 3.34 with two cameras and no GPU. Specifically, the inference time required for YOLOv5 is 0.107 seconds, and the inference time for DeepSORT tracking is 0.192 seconds. Figure 4.5 and Figure 4.6 show the successfully matched face reidentification using the pipeline.

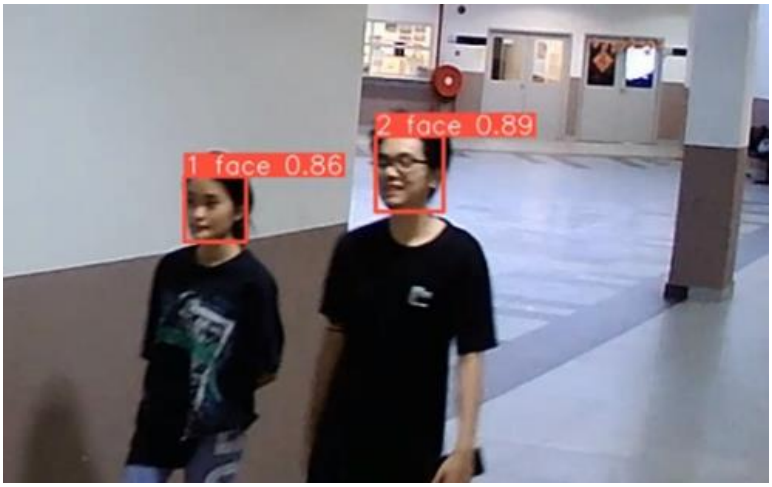

Figure 4.5: Face ID 1 and 2 are shown in camera 1.

Figure 4.6: The system detected and recognized two faces in camera 2 that were previously appeared in camera 1.

4.5 Summary

The evaluation metrics used for the YOLOv5 face detection model are mean average precision (mAP), precision, and recall. Meanwhile, SphereFace face recognition model, the metrics include accuracy (ACC), equal error rate (EER), and area under the ROC curve (AUC).

For YOLOv5, the training results are evaluated based on the different parameters applied, including model architecture, batch size, and epoch. The metric results, including mAP, precision, and recall, are presented in Table 4.1, showing the performance of the model with different parameter configurations. The analysis of the training results and how it helps to improve the model's performance have been reviewed and finally adopted YOLOv5n due to high mAP value and having a balance between precision and recall values.

For the face recognition model, the performance of the SphereFace loss function with its updated versions, SphereFace-R and SphereFace2, on the SurvFace dataset with SFNet20 are evaluated. It is found that the SphereFace works well with SFNet20, achieving the highest accuracy, and thus adopted to the system. SphereFace-R and SphereFace2 are comparatively inferior and not consistent in performance. SFNet20 is adopted in the system's pipeline as it is more lightweight and has a higher inference rate.

Both models were then integrated into DeepSORT for tracking purposes and showed the desired results. The detected faces are assigned with specific IDs with a confidence value.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The face detection and recognition technology has been reviewed with a multi-camera setup and in an unconstrained environment. The pipeline of this project consists of face detection, face recognition, and tracking. In this project, the UFDD dataset is used to train the different versions of the face detection model – YOLOv5, it has shown a desired performance with an mAP value of 0.495, precision value of 0.86799, and recall value of 0.78103. As for the face recognition model, the SurvFace dataset is trained on different loss functions and network architecture of SphereFace; at last, the SphereFace SFNet20 model is adopted into the system with an accuracy of 82.05% and an equal-error rate of 19.17%.

Both of the trained models have achieved the objectives and are integrated into a single pipeline with DeepSORT for single and multi-camera tracking. The final pipeline is able to detect and recognize the face of the subject across multiple cameras for tracking purposes. Our pre-recorded dataset and real-time detection with an unconstrained environment setting are applied to the system and have shown a desired outcome where faces are detected in bounding boxes with a confidence value, recognized with the assigned ID, and tracked across multiple cameras.

5.2 Recommendations for future work

Although the face detection model – YOLOv5 has shown an ideal result with high accuracy in detecting faces in an unconstrained environment the detection speed and processing rate are lower as compared to the latest version of YOLOv8. YOLOv8 features some improvements over YOLOv5, including faster processing and more precise localisation. However, it is relatively new, and has not been applied as thoroughly as YOLOv5, thus once the YOLOv8 state-of-the-art is mature, it is recommended for an upgrade on the face detection model.

Besides, for the face recognition model – SphereFace, it is suggested to have a larger and more diverse dataset for training, as it can further improve the ability to recognize faces in a variety of settings and under different conditions. There is also a need to integrate this model with other computer vision models to create more robust and effective face recognition systems.

Furthermore, the face detection and face recognition models can be optimized using OpenVINO, which is a toolkit that compresses and redesigns models for optimal execution on edge devices. Then, the model can be deployed in low-cost edge AI devices.

A database system that stores the data outcome of tracking from different cameras can also be added as a feature in future work. This feature will be effective to use in many real-life situations such as surveillance, crowd behaviour analysis, and anomaly detection. With this feature, the back-end user can identify the target in a short time and reduce workload.

REFERENCES

Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)* (pp. 1-6). IEEE.

Al-Smadi, Y., Alauthman, M., Al-Qerem, A., Aldweesh, A., Quaddoura, R., Aburub, F., Mansour, K. and Alhmiedat, T., 2023. Early Wildfire Smoke Detection Using Different YOLO Models. *Machines*, *11*(2), p.246.

Ben Fredj, H., Bouguezzi, S. and Souani, C., 2021. Face recognition in unconstrained environment with CNN. *The Visual Computer*, *37*(2), pp.217-226.

Bialkowski, A., Denman, S., Sridharan, S., Fookes, C. and Lucey, P., 2012, December. A database for person re-identification in multi-camera surveillance networks. In *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)* (pp. 1-8). IEEE.

Chen, Z., Cao, L. and Wang, Q., 2022. Yolov5-based vehicle detection method for high-resolution UAV images. *Mobile Information Systems*, *2022*.

Cheng, Z., Zhu, X. and Gong, S., 2018. Surveillance face recognition challenge. *arXiv preprint arXiv:1804.09691*.

Chi, C., Zhang, S., Xing, J., Lei, Z., Li, S.Z. and Zou, X., 2019, July. Selective refinement network for high performance face detection. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 8231-8238).

Coşkun, M., Uçar, A., Yildirim, Ö. and Demir, Y., 2017, November. Face recognition based on convolutional neural network. In *2017 International Conference on Modern Electrical and Energy Systems (MEES)* (pp. 376-379). IEEE.

Deng, J., Guo, J., Xue, N. and Zafeiriou, S., 2019. 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).

Dietlmeier, J., Antony, J., McGuinness, K. and O'Connor, N.E., 2021, January. How important are faces for person re-identification?. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 6912-6919). IEEE.

Du, H., Shi, H., Liu, Y., Wang, J., Lei, Z., Zeng, D. and Mei, T., 2020. Semi-siamese training for shallow face learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16* (pp. 36-53). Springer International Publishing.

Fang, H., Deng, W., Zhong, Y. and Hu, J., 2020. Generate to adapt: Resolution adaption network for surveillance face recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16* (pp. 741-758). Springer International Publishing.

Fatihah, N.N., Ariyanto, G., Latipah, A.J. and Pangestuty, D.M., 2018, August. Face Recognition Using Local Binary Pattern and Nearest Neighbour Classification. In *2018 International Symposium on Advanced Intelligent Informatics (SAIN)* (pp. 142-147). IEEE.

Gan, G., Ma, C. and Wu, J., 2020. *Data clustering: theory, algorithms, and applications*. Society for Industrial and Applied Mathematics.

Guo, G. and Zhang, N., 2019. A survey on deep learning based face recognition. *Computer vision and image understanding*, *189*, p.102805.

He, Y., Wei, X., Hong, X., Shi, W. and Gong, Y., 2020. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, *29*, pp.5191-5205.

Hou, X., Wang, Y. and Chau, L.P., 2019, September. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

Hsu, H.M., Cai, J., Wang, Y., Hwang, J.N. and Kim, K.J., 2021. Multi-target multi-camera tracking of vehicles using metadata-aided Re-id and trajectory-based camera link model. *IEEE Transactions on Image Processing*, *30*, pp.5198-5210.

Hussain, T., Hussain, D., Hussain, I., AlSalman, H., Hussain, S., Ullah, S.S. and Al-Hadhrami, S., 2022. Internet of things with deep learning-based face recognition approach for authentication in control medical systems. *Computational and Mathematical Methods in Medicine*, *2022*.

Iftikhar, S., Asim, M., Zhang, Z., Muthanna, A., Chen, J., El-Affendi, M., Sedik, A. and Abd El-Latif, A.A., 2023. Target Detection and Recognition for Traffic Congestion in Smart Cities Using Deep Learning-Enabled UAVs: A Review and Analysis. *Applied Sciences*, *13*(6), p.3995.

Israel, L. and Bolton, A., 2020. Multi-Target, Multi-Camera Tracking. *MARS* 81: 83-5.

Kalake, L., Wan, W. and Hou, L., 2021. Analysis based on recent deep learning approaches applied in real-time multi-object tracking: a review. *IEEE Access*, *9*, pp.32650-32671.

Kettnaker, V. and Zabih, R., 1999, June. Bayesian multi-camera surveillance. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)* (Vol. 2, pp. 253-259). IEEE.

Kim, J.A., Sung, J.Y. and Park, S.H., 2020, November. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In 2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia) (pp. 1-4). IEEE.

Koch, G., Zemel, R. and Salakhutdinov, R., 2015, July. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (Vol. 2, No. 1).

Kumar, A., Kaur, A. and Kumar, M., 2019. Face detection techniques: a review. *Artificial Intelligence Review*, *52*, pp.927-948.

Kohl, P., Specker, A., Schumann, A. and Beyerer, J., 2020. The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1042-1043).

Lee, I. and Shin, Y.J., 2020. Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, *63*(2), pp.157-170.

Leng, Q., Ye, M. and Tian, Q., 2019. A survey of open-world person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(4), pp.1092-1108.

Li, K., Ma, W., Sajid, U., Wu, Y. and Wang, G., 2020. Object detection with convolutional neural networks. In *Deep Learning in Computer Vision* (pp. 41-62). CRC Press.

Li, Q., Li, R., Ji, K. and Dai, W., 2015, November. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)* (pp. 74-77). IEEE.

Liu, W., Wen, Y., Raj, B., Singh, R. and Weller, A., 2022. Sphereface revived: Unifying hyperspherical face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(2), pp.2458-2474.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B. and Song, L., 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 212-220).

Mahesh, B., 2020. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*, pp.381-386.

Mailewa, A. and Herath, J., 2014, April. Operating systems learning environment with VMware. In *The Midwest Instruction and Computing Symposium. Retrieved from http://www. micsymposium. org/mics2014/ProceedingsMICS_2014/mics2014_submission_14. pdf*.

Medak, D., Posilović, L., Subašić, M., Budimir, M. and Lončarić, S., 2022. DefectDet: A deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images. *Neurocomputing*, *473*, pp.107-115.

Mittal, V. and Bhushan, B., 2020, April. Accelerated computer vision inference with AI on the edge. In *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 55-60). IEEE.

Nada, H., Sindagi, V.A., Zhang, H. and Patel, V.M., 2018, October. Pushing the limits of unconstrained face detection: a challenge dataset and baseline results. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (pp. 1-10). IEEE.

O'Shea, K. and Nash, R., 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Pulli, K., Baksheev, A., Kornyakov, K. and Eruhimov, V., 2012. Real-time computer vision with OpenCV. *Communications of the ACM*, *55*(6), pp.61-69.

Punn, N.S., Sonbhadra, S.K., Agarwal, S. and Rai, G., 2020. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv preprint arXiv:2005.01385*.

Pranav, K.B. and Manikandan, J., 2020. Design and evaluation of a real-time face recognition system using convolutional neural networks. *Procedia Computer Science*, *171*, pp.1651-1659.

Rambach, J., Huber, M.F., Balthasar, M.R. and Zoubir, A.M., 2015, August. Collaborative multi-camera face recognition and tracking. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

Ristani, E. and Tomasi, C., 2018. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6036-6046).

Rusia, Mayank Kumar, and Dushyant Kumar Singh. "A comprehensive survey on techniques to handle face identity threats: challenges and opportunities." *Multimedia Tools and Applications* 82, no. 2 (2023): 1669-1748

Ryan, F., Hu, F., Dietlmeier, J., O'Connor, N.E. and McGuinness, K., 2022, August. Beyond social distancing: application of real-world coordinates in a multi-camera system with privacy protection. In *Irish Machine Vision and Image Processing Conference*. Irish Pattern Recognition & Classification Society.

Saffar, M.T., Rekabdar, B., Louis, S. and Nicolescu, M., 2015, July. Face recognition in unconstrained environments. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.

Saha, S., 2018. A comprehensive guide to convolutional neural networks—the ELI5 way. *Towards data science*, *15*.

Schroff, F., Kalenichenko, D. and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

Simonyan, K., Vedaldi, A. and Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smith, M. and Miller, S., 2022. The ethical application of biometric facial recognition technology. *Ai & Society*, pp.1-9.

Song, C. and Ji, S., 2022. Face Recognition Method Based on Siamese Networks Under Non-Restricted Conditions. *IEEE Access*, *10*, pp.40432-40444.

Sun, G., Wang, S. and Xie, J., 2023. An Image Object Detection Model Based on Mixed Attention Mechanism Optimized YOLOv5. *Electronics*, *12*(7), p.1515.

Tan, Z., Liu, A., Wan, J., Lei, Z. and Guo, G., 2022. Exploring the Limits of Hard Example Mining for ID Document to Selfie Matching. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *4*(4), pp.570-581.

Tronci, R., Giacinto, G. and Roli, F., 2009. Dynamic score combination: A supervised and unsupervised score combination method. In *Machine Learning and Data Mining in Pattern Recognition: 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009. Proceedings 6* (pp. 163-177). Springer Berlin Heidelberg.

Unterberger, A., Menser, J., Kempf, A. and Mohri, K., 2019, September. Evolutionary camera pose estimation of a multi-camera setup for computed tomography. In *2019 IEEE International Conference on Image Processing (ICIP)* (pp. 464-468). IEEE.

Veeramani, B., Raymond, J.W. and Chanda, P., 2018. DeepSort: deep convolutional networks for sorting haploid maize seeds. *BMC bioinformatics*, *19*, pp.1-9.

Wang, J., Chen, Y., Dong, Z. and Gao, M., 2022. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Computing and Applications*, pp.1-13.

Wolf, L., Hassner, T. and Maoz, I., 2011, June. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011* (pp. 529-534). IEEE.

Xu, R., Lin, H., Lu, K., Cao, L. and Liu, Y., 2021. A forest fire detection system based on ensemble learning. *Forests*, *12*(2), p.217.

Xu, Q., Zhu, Z., Ge, H., Zhang, Z. and Zang, X., 2021. Effective face detector based on yolov5 and superresolution reconstruction. *Computational and Mathematical Methods in Medicine*, *2021*, pp.1-9.

Yang, J., Ge, H., Yang, J., Tong, Y. and Su, S., 2022. Online Pedestrian Multiple-Object Tracking with Prediction Refinement and Track Classification. *Neural Processing Letters*, *54*(6), pp.4893-4919.

Yang, B., Yan, J., Lei, Z. and Li, S.Z., 2015, May. Fine-grained evaluation on face detection in the wild. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (Vol. 1, pp. 1-7). IEEE.

Ying, X., 2019, February. An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.

Yuan, S., Du, Y., Liu, M., Yue, S., Li, B. and Zhang, H., 2022. YOLOv5-Ytiny: A Miniature Aggregate Detection and Classification Model. *Electronics*, *11*(11), p.1743.

Zafeiriou, S., Zhang, C. and Zhang, Z., 2015. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, *138*, pp.1-24.

Zervos, M., 2013. *Multi-camera face detection and recognition applied to people tracking*.

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O., 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), pp.107-115.

Zhang, L., Sun, L., Yu, L., Dong, X., Chen, J., Cai, W., Wang, C. and Ning, X., 2021. ARFace: attention-aware and regularization for face recognition with reinforcement learning. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *4*(1), pp.30-42.

Zhang, S., Yang, H., Yang, C., Yuan, W., Li, X., Wang, X., Zhang, Y., Cai, X., Sheng, Y., Deng, X. and Huang, W., 2023. Edge Device Detection of Tea Leaves with One Bud and Two Leaves Based on ShuffleNetv2-YOLOv5-Lite-E. *Agronomy*, *13*(2), p.577.

Zhang, G., Li, L., Nado, Z., Martens, J., Sachdeva, S., Dahl, G., Shallue, C. and Grosse, R.B., 2019. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, *32*.

Zhang, Z., Luo, P., Loy, C.C. and Tang, X., 2014. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13* (pp. 94-108). Springer International Publishing.

Zhao, N. and Qin, Z., 2023. A Survey and Test of the Face Detection Model–Based on Yolov5. *Applied and Computational Engineering*, pp.312-322.

Zhao, X., Huang, Z. and Lv, Y., 2022, August. Research on Real-Time Diver Detection and Tracking Method Based on YOLOv5 and DeepSORT. In *2022 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 191-196). IEEE.

Zhou, F., Zhao, H. and Nie, Z., 2021, January. Safety helmet detection based on YOLOv5. In *2021 IEEE International conference on power electronics, computer applications (ICPECA)* (pp. 6-11). IEEE.