DEVELOPMENT OF PERSON IDENTIFICATION APPLICATION FOR VIDEO SURVEILLANCE

By

Soon Phaik Ching

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: DEVELOPMENT OF PERSON IDENTIFICATION APPLICATION FOR VIDEO SURVEILLANCE

Academic Session: 2022 JAN

I, SOON PHAIK CHING **(CAPITAL LETTER)** declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

Phaikching

(Author's signature)

(Supervisor's signature)

Address:

24, LEBUH RAMBAI,
15 PAYA TERUBONG,
11060, AYER ITAM, PENANG.

Date: 20/04/2022

DR. NG HUI FUANG
Supervisor's name

Date: 20/04/2022

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF INFORMATION COMMUNICATION AND TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 20/04/2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***SOON PHAIK CHING*_** (ID No: ***18ACB03005***) has completed this final year project/ dissertation/ thesis* entitled *"Development of Person Identification Application For Video Surveillance"* under the supervision of Dr. Ng Hui Fuang (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology, and Dr. Chai Meei Tyng (Co-Supervisor)* from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

Phaikching

____

(SOON PHAIK CHING)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DEVELOPMENT OF PERSON IDENTIFICATION APPLICATION FOR VIDEO SURVEILLANCE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Phaikching

Signature : ____________________________

Name : SOON PHAIK CHING

Date : 20/04/2022

ACKNOWLEDGEMENT

I would like to express my sincere appreciation and gratitude to my supervisor, Dr. Ng Hui Fuang who has provided me this opportunity to design a person identification application for video surveillance. This is my first milestones in the image processing and deep learning career. Thank you so much for guiding me throughout the development of the application. Furthermore, I would like to thank my parents, who provides me love, support, and encouragement throughout the whole Final Year Project. Thank you for your patience and love.

ABSTRACT

In the real world, CCTVs are implemented and allocated in public and private environment, to ensure public safety. However, it seems to like observing the video surveillance, to figure out a target, is not a simple task. It increases the human cost of video surveillance. Hence, person identification by the video surveillance application is developed in this project.

In this application, user inserts at least two CCTV videos and at most 4 CCTV videos, either in the public environment or private environment. Simultaneously, user inserts images of the target to be identified in these videos. In the end, the output is the videos with the green bounding boxes, which indicates as the target.

After inserting the necessary images and videos, it is time for the back-end process. There are three important processes in this application, such as person detection, person tracking, and person identification. First, person detection implemented YOLOv3. YOLOv3 is a common neural network algorithm for detection. It can detect 80 classes of objects such as a person, car, pot, and more. Therefore, it was set and adjusted to detect persons only. Second, person tracking is important to track the detected person. In this application, person tracking implemented the DeepSORT algorithm for tracking by the tracker. The trackers contained information such as track ID, class type, and bounding boxes. The information of the tracker is necessary for person identification. Third, person identification implemented the CNN model for training. After training, the videos with green bounding boxes are displayed, when the person is predicted as the target.

In conclusion, this person identification application for video surveillance improves the efficiency and accuracy of person identification in multiple videos, instead of physical surveillance by humans, which is resources inefficiency.

TABLE OF CONTENTS

TITLE PAGE **i**

REPORT STATUS DECLARATION FORM **ii**

FYP THESIS SUBMISSION FORM **iii**

DECLARATION OF ORIGINALITY **iv**

ACKNOWLEDGEMENT **v**

ABSTRACT **vi**

TABLE OF CONTENTS **vii**

LIST OF FIGURES **xi**

LIST OF TABLES **xv**

LIST OF ABBREVIATIONS **xvi**

CHAPTER 1 INTRODUCTION **1**

1.1 Problem Statement and Motivation 1

1.2 Project Scope 3

1.3 Objectives 4

1.4 Contributions 5

1.5 Historical Development 6

1.6 Report Organisation 8

CHAPTER 2 LITERATURE REVIEW **10**

2.1 Background 10

2.2 Face Processing 11

2.3 Inter-person Distance 11

2.4 Low Resolution Videos 12

2.5 Domain Gap Between the Reidentification Images and Pose Estimator Images 13

2.6 Semantic Gap 14

2.7 Large Intra-class Variation 15

2.8 Feature Matching Issues 15

2.9 Time Consuming Ranking List Production 16

2.10 Long Transition Time 16

2.11 Mutual Interference Between the Output 17

2.12 Summary of Problems and Solutions 18

CHAPTER 3 SYSTEM METHODOLOGY/APPROACH 21
3.1 System Flowchart for the Application 21
3.2 Use Case Diagram for Application 22
3.2.1 Use Case Description for Uploading Images 22
3.2.2 Use Case Description for Uploading Videos 23
3.2.3 Use Case Description for Watching Person Identification Videos 23
3.2.4 Use Case Description for Downloading Person Identification Videos 24
3.3 Activity Diagram for Application 25
3.4 Timeline 26
3.4.1 Proposal Writing Timeline 26
3.4.2 FYP1 Timeline 27
3.4.3 FYP2 Timeline 28
CHAPTER 4 SYSTEM DESIGN 29
4.1 System Design Concepts 29
4.1.1 Person Detection: YOLOv3 29
4.1.2 Person Tracking: DeepSORT 33
4.1.3 Person Identification: Self-Implemented Model 35
4.1.3.1 Reason of Implementation of Self-Implemented Model 35
4.1.3.2 Data Augmentation 37
4.1.3.3 Convolutional Layers 37
4.1.3.4 Max Pooling 39
4.1.3.5 ReLU Activation 39
4.1.3.6 Dropout 40
4.1.3.7 Flatten 41
4.1.3.8 Dense Layer 42
4.1.3.9 Adam Optimization 42
4.1.3.10 Sparse Categorical Cross Entropy 42
4.1.3.11 Softmax 43
4.1.3.12 Summary of the Model 43
4.2 System Design Procedures 45
4.2.1 Person Identification Library 45
4.2.2 Building Own Model for Person Identification 46
4.2.3 Target Prediction 49

4.2.4 Person Detection 50
4.2.5 Person Tracking 51
4.2.6 Video Dataset (Person Tracking) 51
4.2.7 Model Training 56
4.2.8 Person Identification 58
4.2.8.1 Initial Person Identification 59
4.2.8.2 Predict None 62
4.2.8.3 Model Retraining 65
4.2.9 Remove Directory 66
4.2.10 User Interface Design 66
CHAPTER 5 SYSTEM IMPLEMENTATION 71
5.1 Methodologies and General Work Procedures 71
5.2 Tools to Use 71
5.2.1 Hardware Setup 71
5.2.2 Software Setup 72
5.3 User Requirement 74
5.4 System Performance 74
5.5 Verification Plan 75
5.6 System Operation (with Screenshot) 81
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 88
6.1 System Testing 88
6.2 Use Case Testing 93
6.3 Objectives Evaluation 93
6.4 Project Challenges 94
CHAPTER 7 CONCLUSION AND RECOMMENDATION 99
7.1 Project Review and Discussion 99
7.2 Novelties and Contributions 100
7.3 Conclusion with Supportive Remarks 101
7.4 Future Work 102
REFERENCES 104
APPENDIX A-1
A-1 Weekly Report A-1
A-2 Poster A-11

PLAGIARISM CHECK RESULT **A-12**

FYP 2 CHECKLIST **A-15**

LIST OF FIGURES

Figure Number	Title	Page
Figure 1-1-1	Tracking multiple persons under multiple cameras by application.	1
Figure 1-5-1	Person identification from camera.	7
Figure 2-1-1	Types of Shooting Angles	11
Figure 2-3-1	Difference Between High Inter-Person Distance and Low Inter-Person Distance.	12
Figure 2-4-1	Differences between low-resolution image and normal image	13
Figure 2-6-1	Multi-level sematic representation (a) Market-1501. (b)DukeMTMC-reID	14
Figure 2-7-1	Images with Eigenvector and without Eigenvector	15
Figure 2-11-1	Positive Ranking List and Negative Ranking List	18
Figure 3-1-1	Flowchart of Person Identification Application	21
Figure 3-2-1	Use Case Diagram for Application	22
Figure 3-3-1	Activity Diagram for Application	25
Figure 3-4-1	Proposal Writing Timeline	26
Figure 3-4-2	FYP1 Timeline	27
Figure 3-4-3	FYP2 Timeline	28
Figure 4-1-2	Architecture of YOLOv3	30
Figure 4-1-3	Before and After Computing Class Probability	31
Figure 4-1-4	Intersection over Union	32
Figure 4-1-5	Bounding Boxes Before and After NMS	33

Figure 4-1-6 Classes in YOLOv3 33

Figure 4-1-7 Manhattan Distance Formula 35

Figure 4-1-8 Euclidean Distance Formula 35

Figure 4-1-9 Cosine Distance Formula 35

Figure 4-1-10 Examples of Data Augmentation 37

Figure 4-1-11 Padding 38

Figure 4-1-12 Max Pooling 39

Figure 4-1-13 Sigmoid Function Graph 40

Figure 4-1-14 Neural Network with and without Dropout 41

Figure 4-1-15 Examples of Flatten 41

Figure 4-1-16 Sparse Category Cross Entropy Loss Function 43

Figure 4-1-17 Summary of Model 44

Figure 4-1-18 Gradient Varnishing 45

Figure 4-1-19 Gradient Exploding 45

Figure 4-2-1 The Accuracy and Loss of Training and Validation for Resnet-50 Model 46

Figure 4-2-2 Flowchart for Initial Person Prediction 50

Figure 4-2-3 Videos and Resized Videos 52

Figure 4-2-4 Images Extracted from Different Cameras in Training Set 55

Figure 4-2-5 Person Tracking in Video 1 56

Figure 4-2-6 Person Tracking in Video 2 56

Figure 4-2-7 Summary of Training Set, Validation Set and Class Names 57

Figure 4-2-8	Data Augmentation of Model	57
Figure 4-2-9	Visualisation of the model	58
Figure 4-2-10	Flowchart for Person Identification	60
Figure 4-2-11	Coordinate of Bounding Box	61
Figure 4-2-12	Relationship between ‘id’ and ‘PID’	62
Figure 4-2-13	Person Identification Videos	62
Figure 4-2-14	Flowchart of Predict None	63
Figure 4-2-15	PID directories in Training Directory	64
Figure 4-2-16	For Loop in PredictNone()	64
Figure 4-2-17	Image Uploading UI	67
Figure 4-2-18	Images Uploaded Display	68
Figure 4-2-19	Video Uploading UI	68
Figure 4-2-20	Video Uploading with Loading Bar	68
Figure 4-2-21	Video Uploaded Successfully	69
Figure 4-2-22	Display Person Identification Videos	70
Figure 4-2-23	Download Person Identification Videos	70
Figure 5-1-1	Flowchart of System Overview	71
Figure 5-6-1	Uploading Images	82
Figure 5-6-2	Showing A Row of Uploaded Images	82
Figure 5-6-3	Uploading Videos	83
Figure 5-6-4	Processing the Video for Upload	83
Figure 5-6-5	Video Uploading Status	83

Figure 5-6-6	Person Identification Status of Video 1	84
Figure 5-6-7	Person Identification Status of Video 2	84
Figure 5-6-8	Person Identification Status of Video 3	85
Figure 5-6-9	Person Identification Status of Video 4	85
Figure 5-6-10	Showing Person Identification Videos Concurrently	86
Figure 5-6-11	Download Person Identification Videos	86
Figure 5-6-12	Downloaded Person Identification Videos in Download Directory	86
Figure 5-6-13	Downloaded VideoIdentification1	87
Figure 6-1-1	Target for Person Identification	88
Figure 6-1-2	Persons Cropped	89
Figure 6-1-3	Person Identified in The Video	89
Figure 6-1-4	Person Tracking for Image Retrieval	90
Figure 6-1-5	Frames in the Video Identification 1	91
Figure 6-1-6	Frames in the Video Identification 2	91
Figure 6-1-7	Frames in the Video Identification 3	92
Figure 6-1-8	Frames in the Video Identification 4	92
Figure 6-4-1	Predicted Accurately Before the Persons were Close	95
Figure 6-4-2	DeepSORT Tracker issue	96
Figure 6-4-3	Undetected Person in The Frame.	97
Figure 7-3-1	Person identification in Video Surveillance Application	102

LIST OF TABLES

Table Number	Title	Page
Table 2-12-1	Summary of Problems and Solutions.	19
Table 3-2-1	Use Case Description for Uploading Images	22
Table 3-2-2	Use Case Description for Uploading Videos	23
Table 3-2-3	Use Case Description for Watching Person Identification Videos	23
Table 3-2-4	Use Case Description for Downloading Person Identification Videos	24
Table 4-2-1	Summary of All Models	47
Table 4-2-2	Summary of Models with 6 convolutional layers Without Dropout and With Dropout	48
Table 5-2-1	Hardware Requirements	72
Table 5-2-2	Libraries in Python	73
Table 5-5-1	Validation of Person Occlusion	76
Table 5-5-2	Validation of Interchange of Person Identifier	77
Table 5-5-3	Validation of Person Image Retrieval with Low Resolution	79
Table 5-5-4	Validation of Person Identification Camera Issues	80
Table 6-3-1	Use Case Testing	93

LIST OF ABBREVIATIONS

2D	2-Dimensional
acc	Accuracy
AdaRSVMs	Adaptive Ranking Support Vector Machines
AFIS	Automated Fingerprint Identification System
BOF	Bag of Features
CCTVs	Closed-Circuit Televisions
cls	Classification
CMC	Cumulative Match Curve
COCO	Common Objects in Context
CPU	Central Processing Unit
DNA	Deoxyribonucleic Acid
EDANet	Efficient Dense Modules with Asymmetric Convolution
EGLRN	Extended Global-Local Representation Learning Network
eps	epochs
et al.	And others
etc.	And more
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ID	Identifier

IoU	Intersection over Union
LDA	Linear Discriminant Analysis
LIBSVM	Lightweight Library for Support Vector Machines
max	Maximum
mAP	Mean Average Precision
MLR-DUKEMTMC-REID	Duke Multi-Tracking Multi-Camera Reidentification
MSA-SR PREID	Multi-Scale Adaptive Super-Resolution Person Re-Identification
MTCNN	Multi-Task Cascaded Convolutional Neural Network
NMS	Non-Maximum Suppression
no.	Number
PAAN	Part-Based Attribute-Aware Network for Person Re-Identification
PCA	Principal Component Analysis
PID	Person Identifier
PLDA	Pseudo-Inverse Linear Discriminant Analysis
PNG	Portable Graphics Format
pp.	Page/pages
RAM	Random Access Memory
R-CNN	Region-Based Convolutional Neural Network
reg	Regression
ResNet	Residual Neural Network
RGB	Red, Green and Blue

RNN	Recurrent Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
RPN	Region Proposal Network
SDM	Supervised Descent Method
SNS	Supervised Non-Local Similarity
SSN	Social Security Number
SURF	Speeded Up Robust Feature Algorithm
SVM	Support Vector Machine
UI	User Interface
vol.	Volume (journal)
XQDA	Cross-View Quadratic Discrimination Analysis

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Motivation

1. Video surveillance from multiple video cameras by humans is ineffective.

Video surveillance is an action of observing improper behaviour of people which may trigger the emergency alert for safety and security purposes. It increases the overall security system and lowers criminal actions. However, video surveillance through multiple Closed-Circuit Televisions (CCTVs) is not an easy task, for the security and safety department to maintain peace and safety of the public. It requires sufficient manpower and a long duration of observing and watching the CCTV. Also, the cost of security increases indirectly as the cost of security maintenance and manpower is high. The person sitting in front of the screens with multiple CCTV videos, must focus when they are observing videos, to identify target or for normal observation. It requires sufficient manpower and long duration of observation. The person in charge must concentrate when watching the real-time recording video to prevent any oversight because oversight may lead to higher financial cost and safety issues. Arguably, humans make mistakes. Therefore, video surveillance by humans is not an efficient way. In Figure 1-1-1, it visualised the implementation of person tracking under multiple cameras.

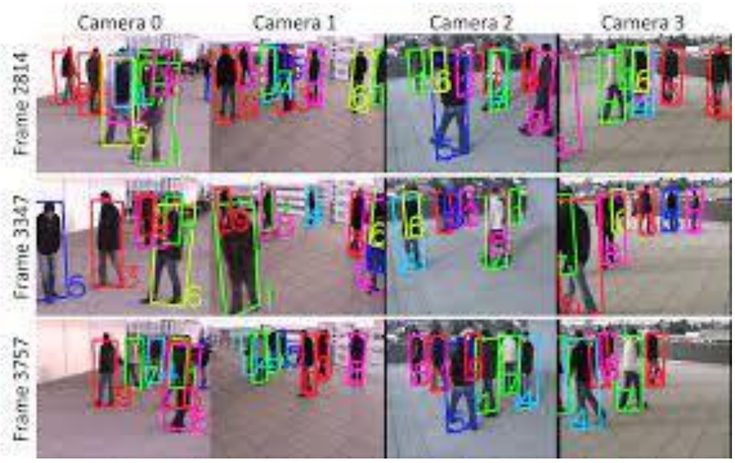


Figure 1-1-1: Tracking multiple persons under multiple cameras by application. [1]

2. Extraction methods for person identification differ in different situations.

With the person identification in video surveillance application, these problems will be solved efficiently. By applying video surveillance associated with person identification, the targeted person will be detected and identified over multiple real-time camera videos. When the cameras are capturing, the person identification system tracks down the person simultaneously. If any suspicious target is detected, the security alarm will be sounded for security purposes. However, identifying extraction method for person identification may be difficult. Different environmental factors apply different extraction method to identify a person from the videos. Some researchers found out a Faster R-CNN, but it was able to work under controlled requirements only. It did not solve other feature matching challenges such as intra-class variation and inter-class confusion [2]. Therefore, different extraction method may solve and apply in different challenging situations.

3. Person identification is applied for video surveillance in multiple low-resolution videos.

Low-resolution videos may lead to additional time for feature extraction in both local features and global features. The examples of low-resolution videos are blurred videos, high or low light exposure videos etc. In different light exposure, the same person may look different. This may lead to false identification or misidentification. At the same time, features in multiple videos will be extracted for person identification. Therefore, the extraction of features may consume a lot of time, causing the delay of person identification [3]. Hence, the upscaling of videos or images extracted is needed, to reduce time-wasting on feature extraction. However, it is not enough for person identification, as it increases the visibility of for person detection only. Therefore, it should work with other features extraction, to increase the effectiveness of the person identification for video surveillance.

1.2 **Project Scope**

In this project, a person identification application for video surveillance will be developed. Person identification is a method to identify a person through different methods such as palm prints, fingerprints, face, iris, etc. Nowadays, video surveillance is common in the public environment to reduce criminal activities, especially in high-security locations such as banks and financial companies. However, the surveillance maintenance fees to identify and analyse the suspicious person may cost higher than the installation of closed-circuit televisions (CCTVs). Therefore, through the application of person identification for video surveillance, it may reduce the cost of maintenance of the CCTVs and improve the efficiency of video surveillance.

The input of the person identification application are at least two videos and eight images of the target. On the other hand, the output is the videos with green bounding boxes, indicating that it is the target. To transform and process the inputs into outputs, there are several back-end processes, such as person detection, person tracking and person identification.

First, person detection implements YOLOv3 model which is common for object detection. YOLOv3 is pretrained with the Coco datasets. Coco datasets has 80 classes of object, but this application focuses on person class only. After the person detected, person tracking is necessary. In DeepSORT, NMS prevents the multiple detections. When the NMS was set as 0.8, it means that only those with NMS of 0.8 are considered as positive detections. To increase the accuracy of detections, confidence is a very important measure. It is a measure for the positively detected object with the predefined classes. The greater the confidence, the greater the accuracy of detections.

Second, in person tracking, DeepSORT algorithm with Nearest Neighbour algorithm associate the relationship between the tracking objects. Nearest Neighbour algorithm measures the metric distance and similarities with cosine distance metrics. During person tracking, the tracker in DeepSORT tracks the detected person. At the same moment, the tracker stores information such as track ID, class type and bounding boxes information for person identification.

Third, person identification implements the model trained to predict the target and people in the videos. The CNN model consists of 6 convolutional layers with dropout. The model implements the Sparse Categorical Cross Entropy in classification layer and Adam optimizer.

After these back-end process, the videos with multiple bounding boxes will be displayed. The person in green bounding boxes is target, as predicted with the CNN trained model.

1.3 **Objectives**

1. To develop a person identification application for video surveillance for monitoring suspicious person.
2. To develop a model to perform person identification and tracking in surveillance video continuously.
3. To develop a model that performs person identification over multiple videos at the same moment.

First, the main objective of this research ‘development of person identification application for video surveillance' is to develop a person identification application for video surveillance that monitoring suspicious targeted person. Through the CNN model, the person will be identified by using the embedding of features. Although there is no control on the feature to be trained in CNN model, but it learns itself to identify in model training. To increase to the performance of the model, fine-tune the CNN model.

To achieve the main objective, the second objective of this research is to develop a model for person identification and person tracking in surveillance video continuously. In DeepSORT for person tracking, create a tracker for each detected person. Hence, the path of the pedestrian including the targeted person will be tracked in multiple CCTVs automatically in this application. Therefore, the security department who installs this application can identify the latest path of the targeted person from multiple videos inserted.

Third, multiple video cameras will be applied on person identification at the same moment. Every video that has been streamed by the CCTVs have some blind spots. Blind spot is described as the area that camera cannot capture. Therefore, multiple CCTVs work concurrently to identify and track a targeted person for better video surveillance. The more the videos, the lesser the blind spots in the certain area. For better security, more CCTVs are needed. Therefore, a person identification application with multiple videos for video surveillance is developed.

1.4 **Contributions**

1. Person identification for video surveillance with CNN model: This application allows multiples of videos to be streamed from the video cameras such as CCTVs. Therefore, it can trigger the alarm warning once any suspicious person has been identified directly. This prevents any wanted person from escaping, as the path of the targeted person will be tracked through multiple video cameras. To achieve this objective, person identification model is crucial. Therefore, CNN with consistent precision and accuracy will be chosen.

2. Person identification through multiple videos: The video camera will have blind spots that the camera cannot be streamed. Therefore, the more the video cameras are allocated, the more videos are captured, the lesser the blind spot will have. Hence, the effectiveness of the person identification application increases.

1.5 Historical Development

Identity verification has been around for hundreds of centuries. The initial purpose of identity verification allowed others to identify a person from the crowd. The oldest physical method was jewellery recognition, as the jewellery reflected status of a person such as wealth, familial ties, and person identity. Furthermore, tattooing was an old person identification method. Visible tattoos reflected one's identity. In 1940, British emphasised the personal documents of an individual, such as passport and Social Security Number (SSN) to identify a person. In 1858, Sir William Herschel broke through the biometric person identification, he implemented ink fingerprints as fingerprint was a unique identity of a person. Also, it was further improved into the Automated Fingerprint Identification System (AFIS). In 2004, United States was the first to introduce the automated palm print database.

After the development of palm print technique, advanced biometrics person identification through speech, iris, face, DNA sequencing, and vascular pattern were developed. In this research, person identification in video surveillance through appearances will be conducted. Through multiple surveillance videos, videos will be analysed by image processing algorithms and deep learning algorithms, to track the person without his/ her notice in the public [4].

According to [5], person identification is categorized into traditional approach and biometrics approach. The traditional identification approach relied on the changeable or manipulated parameters such as card identification and password identification. On the other hand, biometrics approach depends on unique human traits such as fingerprint, face detection, and voice. These two approaches support the person identification. However, the biometric approach is more reliable than the traditional approach as biometrics which is derived from behavioural human characteristics establishes an identity of a person. In our daily life, person identification acts an important role to ensure that our privacy is protected securely. An example was shown in Figure 1-5-1, although the person is partially blocked by other person, the application should be able to match the blocked person with target accurately.

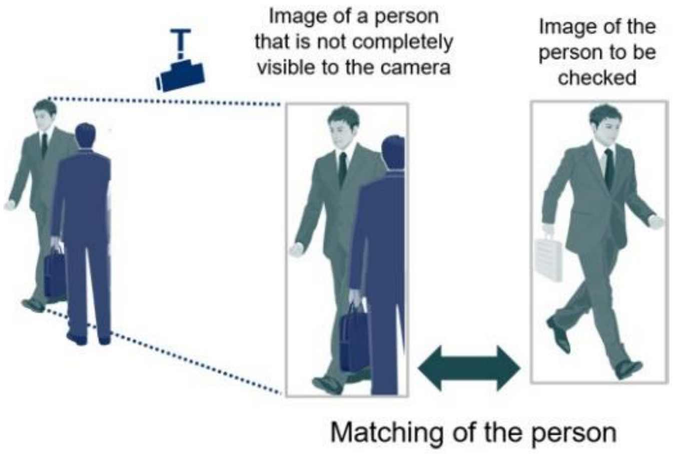


Image of a person that is not completely visible to the camera
Image of the person to be checked
Matching of the person

Figure 1-5-1: Person identification from camera. [6]

In specific, person identification is outlined as a face matching and recognising from the face detection database. However, the requirements of person identification are strict as person identification faces problems such as person scale, person orientation, light exposure, and more. For example, the targeted person in the crowd must be close enough towards the capturing cameras for person identification, it reflects the strict requirement for identification. Due to the restrictions of person identification, researchers have explored person reidentification. The researchers [7] theorized that person reidentification is a method of identifying a person over the multiple non-overlapping cameras in various locations or/and times automatically. The requirements for person reidentification are looser than person identification. The person reidentification can identify the target over multiple low-quality cameras, far distance target, or non-salient body parts of the target. In brief, person reidentification is an advanced version of person identification.

Person identification methods are based on two features, dynamic features, and physical features. The dynamic feature is known as the gait approach, while the physical feature is known as the appearance-based approach. Gait approach depends on the body joints motion, the accuracy of person identification will be reduced due to the slope of the surfaces. On the other hand, appearance-based relies heavily on the approach visual features such as shape, colour, and expressiveness of data [8].

1.6 Report Organisation

This project report consisted of seven chapters. Chapter 1 is an introduction to the development of a person identification application for video surveillance. The problem statement, motivation, project scope, objectives, and historical development were included in Chapter 1, to provide a brief introduction. Chapter 2 consists of the literature reviews and comparison between the research papers related to person identification. In each literature review, the strengths and weaknesses of the person identification algorithm were reviewed and summarized.

In addition, Chapter 3 is about the system methodology or approach. In Chapter 3, the flowchart and use case diagrams of the person identification application were shown with a description. The timeline was proposed in this chapter. In chapter 4, the system design is divided into 2 parts, such as system design concepts and system design procedures. System design concepts were explained and described for person detection, person tracking, and person identification. To have a better idea of this application, the system overview section explained three important stages to be implemented in this application, such as person detection using YOLOv3, person tracking using DeepSORT, and person identification using the CNN model. In the system design procedures, the ups and downs during the stages of the person identification implementation were listed from the beginning until the end of the person identification project.

Chapter 5 proposes system implementation. The system implementation lists the general work procedure, tools to use, user requirements, system performance, and system validation plan and system operation. The system UI operations were shown step by step from the head to the toes, with the screenshots and description. Chapter 6 proposes system evaluation and discussions, such as testing setup and result, project challenges, use case testing, and objectives evaluation. System testing showed how well the application performs. Challenges in this application were listed in this chapter. Last, chapter 7 is the conclusion, which provided the summary of the whole report on the development of person identification applications for video surveillance. Chapter 7 proposed the novelties and future implementation too.

CHAPTER 2

LITERATURE REVIEW

2.1 **Background**

In real world, video surveillance systems capture the activities, time by time to protect the people and reduce crimes. As people are under monitoring of the video surveillance system, people may feel safer if they are protected. However, video surveillance is a huge burden for security and safety department. Therefore, person identification in the video surveillance system works here. Person identification identifies the targeted persons through multiple video cameras. The path of the person detected and identified will be tracked down for security and safety purposes of the public. Even the video camera may have blind spots which are unable to capture the person, the person identification system will analyse and detect the targeted person from multiple under controlled video cameras. This is the advantage of the person identification for the video surveillance, which humans can't work this out manually. Hence, person identification is an important tool to prosecute criminal and enforce security.

However, person identification encounters technical and environmental problems. The problems are categorized into feature representation and feature matching. Feature representation is about the conditions of the environment as there are always camera noises such as cross-view of the people, illumination condition, background, human position, human gesture, shooting angle, blocking and more. Capturing angles of cameras are divided into four categories, they are face-shot, first-person view, slope view, and aerial view as shown in Figure 2-1-1 [8]. Furthermore, the low resolution of the videos and the difference of time transition between cameras, which reduce the effectiveness of person identification are considered as feature representation. Therefore, both environment and humans may reduce the effectiveness of the person-reidentification system. On the other hand, feature matching contains small people size, intra-class variation, inter-class confusion and more. Problems such as face processing, inter-person distance, low-resolution videos, domain gap, semantic gap, large intra-class variation, singular scatter matrices, feature representation and feature matching, time consuming ranking list production, long transition time, and mutual interference between the outputs were discussed below with the previous research solutions.

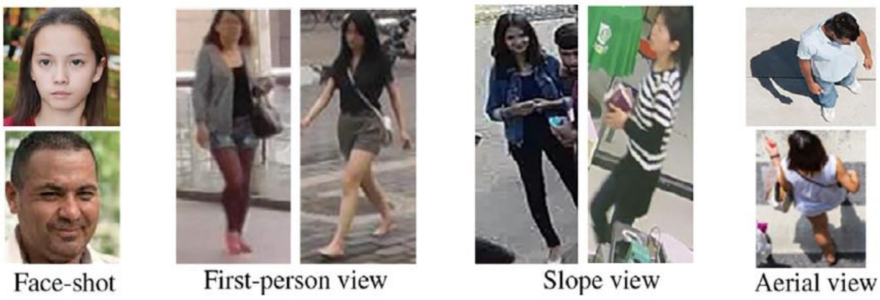


Figure 2-1-1: Types of Shooting Angles [8]

2.2 **Face Processing**

Image processing and deep learning are common skills required for person reidentification. The researchers [2] theorized the application of a Faster R-CNN, a face detection algorithm under controlled requirements for indoor video surveillance. It detected the facial feature and obtained the landmark points through the supervised descent method (SDM), then recognised the face by the joint Bayesian model. The problems such as face detection and verification were solved, as some of the audience wore a face mask, goggle or cap which reduced the visibility of the face. In addition, the appearance of a person looked differently in different environment, due to the posture, partial occlusions, emotional facial expression, illuminations, and low video resolutions. However, it was suitable for indoor video surveillance under a controlled environment only.

2.3 **Inter-person Distance**

Despite the face detection problems, the inter-person distance which means the physical distance between people was a challenge of person identification. Figure 2-3-1 showed the difference between the high inter-person distance and low inter-person distance. The scholars [9] had developed Cross-view Quadratic Discrimination

Analysis (XQDA), in which the intra-person distance was minimized, and the inter-person distance was maximized. Nguyen, et al combined and applied both the late fusion algorithm and metric learning algorithm for the person identification. Furthermore, the Residual Neural Network (ResNet) which was a neural network that extracts classifiers and features, had been modified for local features extraction from multiple person pictures in the video surveillance. However, its drawback was that the single shot was not as effective as multiple shots because the feature was extracted for unique images only. To identify the 2D joint cross-view correspondences in the presence of noise, the China researchers [10] deployed a 3D hypothesis clustering for unlabelled movement data of multiple people from the video. The 3D-hypothesis clustering matched the cross-view of the 2D joint with the noisy detection from the 2D joint across multiple videos to robust the multiple person posture reconstructions. Another contribution of this research was that this method allowed the motion capture of multiple people automatically.

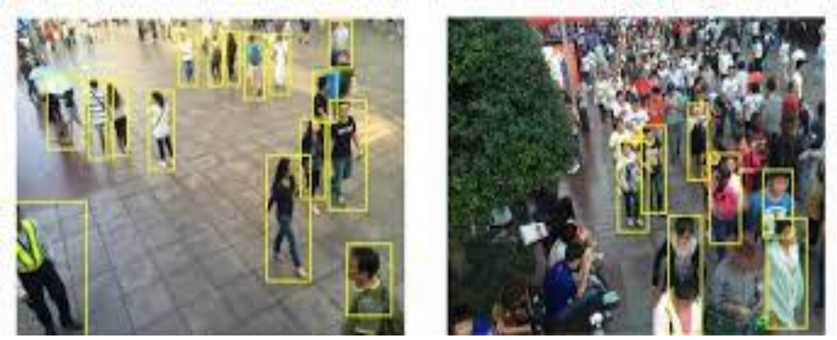


Figure 2-3-1: Difference Between High Inter-Person Distance and Low Inter-Person Distance. [11]

2.4 **Low Resolution Videos**

In addition, the low resolution of the video wasted the time on image extraction. There were two team of researchers had solved the low video resolution problem. [3] implemented the extended global-local representation learning network (EGLRN) which had two streams. One, was for local feature learning on videos, while another stream, allowed extracting of global representation, then channelling attention to Convolutional neural network (CNN) networks. Both local and global features improved the discrimination of the final spatial-temporal feature for the low resolution of the videos. EGLRN was the time-consuming solution. However, it resolved frames weight issues as RNN captured cues between next frames. Similarly, to solve the low-resolution of the camera videos, [2] deployed MSA-SR PREID which was an enhancement of Generative Adversarial Network (GAN). GAN was a super resolution image upscaling method to upgrade the images from low-resolution to high-resolution. The low-resolution images are known as the images that have low pixel intensity, pixel gradient, colour orientation, etc, as shown in Figure 2-4-1. Due to the low-resolution problems, GAN was implemented for the low-resolution feature extraction. Comparing between both, MSA-SR PREID performed better as it had achieved 79.06% accuracy on Duke Multi-Tracking Multi-Camera Reidentification (MLR-DUKEMTMC-REID) dataset. However, it focussed on the super-resolution images for future reidentification network, but it did not emphasize on the domain gap between the real-world images and the human parser training images.

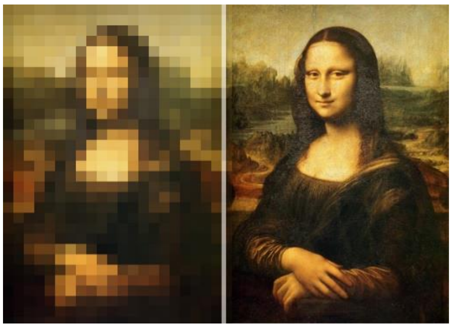


Figure 2-4-1: Differences between low-resolution image and normal image [13]

2.5 **Domain Gap Between the Reidentification Images and Pose Estimator Images**

The domain gap between the reidentification images and pose estimator training images was a critical person reidentification issue. [14] had solved by the application of Supervised Non-Local Similarity (SNS) which was a feature learning for person reidentification. It created anchors by locating the geometric centres on a convolutional feature map. The anchors compared and learned with other pixels in the convolutional feature map, then generated similar feature by absorbing similar pixels. To evaluate SNS for overall framework, SNS has been applied on types of human parser such as Efficient Dense modules with Asymmetric convolution (EDANet), DeepLabV2, and DeepLab V3+. The SNS learning solved the noisy human pose estimation such as inaccurate detection of human body parts. However, it did not resolve the low resolution of videos.

2.6 **Semantic Gap**

The semantic gap differs from the domain gap. The semantic gap is the difference or contrast between features such as local features and global features or high-level and mid-level features. Figure 2-6-1 showed multi-level sematic representation in Market-1501 dataset and DukeMTMC-reID dataset. The target in Market-1501 dataset was described by the features such as female, long, hair, teenage, white upper clothing, red bottom skirt, handbag, short sleeve, short dress and more. Furthermore, the target in DukeMTMC-reID dataset was described by the clothing features, gender feature, accessories feature and more, like in the target in Market-1501 dataset. In PAAN, the semantic gap between high-level and mid-level features was bridged by semantic bridge architecture. It enhanced the expressiveness of global representation. By implementing PAAN, layer partition strategy, and semantic bridge, global or local semantic information were exploited as guidance for the optimization of global representation. In addition, images issue with non-salient, complex, and incomplete information which were hard to portray for the global features had been solved by applying PAAN. However, it required a high quality of cameras as the method did not suitable for low-resolution images [15].

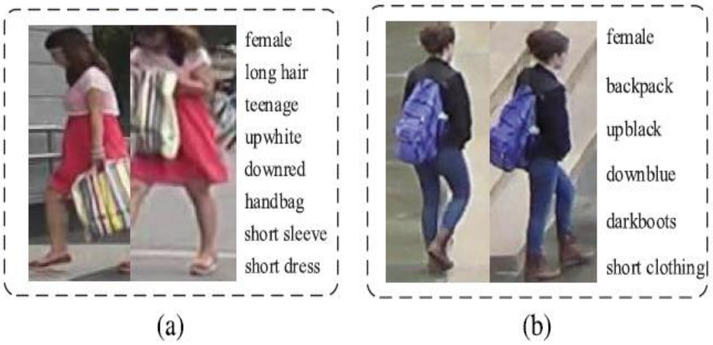


Figure 2-6-1: Multi-level sematic representation (a) Market-1501. (b)DukeMTMC-reID [16]

2.7 **Large Intra-class Variation**

Due to the change in occlusion across views and person pose, large intra-class variation occurs. Large intra-class variation is defined as the occurrence of image variation within a class. To solve this large intra-class variation problem, minimize a within-class variance metric, and maximize the between-class variance metrics. However, all the scatter matrices were singular, due to the dimension of sample size was generally smaller than that of the sample feature vector. It did not meet the main requirement for the metric which the scatter matrices should be non-singular. Hence, the metric could not proceed its learning as the scatter matrix was singular. Therefore, [17] solved the singularity issue on person reidentification with PLDA. To solve the singularity problem, an orthogonal transformation is learned with within-class scatter matrix, which was the pseudo-inversed, instead of direct inverse of that, as regular LDA which suffered from degeneration of eigen value was not optimized. To understand eigen value better, Figure 2-7-1 showed the difference between image with eigen vector and image without eigen vector. Furthermore, the researcher developed a kernel version orthogonal transformation learning, to robust the person reidentification. This learning was against the common data distribution which was non-linear. To improve the effectiveness, a fast version of performance remained was generated. In this research, the features were extracted from unsupervised algorithm. Nevertheless, semi-supervised algorithm which is more effective will be considered by the researcher in further.

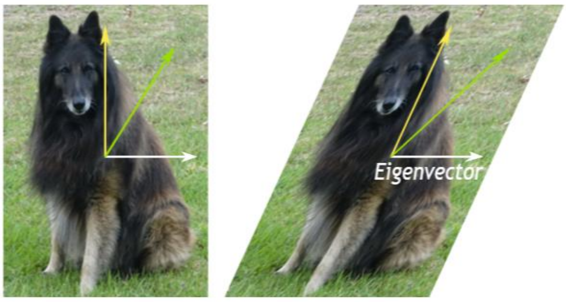


Figure 2-7-1: Images with Eigenvector and without Eigenvector [18]

2.8 **Feature Matching Issues**

According to [7], as there was no good research based on feature matching schema, scholars had proposed the improved Bag of Features (BOF) based on Speeded Up Robust Feature algorithm (SURF) for person reidentification. This algorithm had solved the feature representation and feature matching problems. Feature representation is defined as the variety of the video environment such as body posture, brightness, occlusion, angle of the video captured, etc. On the other hand, the examples of feature matching are small samples size, intra-class variation, etc. In addition, LIBSVM method increased effectiveness of the classification. In addition, the covariance descriptor needed fewer sample datasets when matching. Both LIBSVM and covariance descriptor increased effectiveness and efficiency of the person reidentification. The final match and classification were affected by traditional BOF algorithm, even the BOF had been improved. Also, the researcher team applied the local feature only, local features fused with more features such as the colour and the texture. In addition, the calculation of this method was too complex, and could be simplified in the future.

2.9 **Time Consuming Ranking List Production**

From a similar standpoint, [16] encountered the time-consuming problem by designing a ranking list named Loopnet. Loopnet has a coincident characteristics with the AdaRSVMs, as it preserves the ranking list for global hard sample mining. The ranking list has two categories, which is positive and negative. However, production of ranking lists is timing consuming as it is the outcome between the distance calculation layer, which is output and the sampling layer, which is input. To overcome this challenge, ranking lists were created progressively by the multiplet loss which aimed for hard and easy training sampling. Also, global hard sample mining optimized and improved the efficiency of the learning model by working with the multiplet loss which conducted training through positive and negative samples. This was the solution developed for time consuming ranking list production of person reidentification by [16].

2.10 **Long Transition Time**

Positive and negative training sets are applied commonly in the person reidentification application. Positive training set is the trained datasets with the correct target or person identified. Oppositely, negative training set is the inaccurate and incompatible target dataset. Through adaptive ranking support vector machines (AdaRSVMs), the estimation of targeted positive information based on labelled data from unlabelled negative data was improved. This was the inspiration from the adaptive learning algorithms. AdaRSVMs had a high target positive mean and low target negative image pairs. High target positive mean is the high mean of the positive dataset, while low target negative image pairs consist of negative targets. In addition, AdaRSVMs performed better than the existing domain adaptation method, asymmetric domain adaptation method was applied for estimation of target positive mean. Due to the transition time problem, some of the labels were unable to be collected from certain video cameras. The transition time between the cameras as collecting data from a large-scale camera network was time-consuming. Hence, [19] applied an AdaRSVMs method to solve the problem. AdaRSVMs identified the person through the targeted domain camera without collecting the labels from every training subject to the camera. By application of AdaRSVMs for person reidentification, the cost and time-consuming was reduced, as person labels were not needed.

Differing from Loopnet, the researchers of AdaRSVMs did not apply a mini batch algorithm, which was commonly applied by other researchers. It is because the mini batch algorithm mined the hardest sample only, this caused bad local optimal training solutions. Although AdaRSVMs reduced the time-consuming initialisation challenge, it may generalize the detection and classification issues for person reidentification.

2.11 **Mutual Interference Between the Output**

Also, [20] employed deep group shuffling dual random walk with label smoothing which performed random walk for positive and verification information, despite of applying metric learning-based methods, as the application cost metric learning-based methods was high, and it was not suitable for large datasets. Deep group shuffling dual random walk with label smoothing reduced the mutual interference between the output, without separating them into probe and gallery training sets. The conventional random walk predicted the results as normal distribution or log-normal distribution. However, this leaded mutual interference problem between two-dimensional outputs when applying. Therefore, [20] deployed deep group shuffling dual random walk to solve the mutual interference which caused by the conventional random walk. The result of deep group shuffling dual random walk algorithm was shown in Figure 2-11-1. The images in positive ranked list were bounded with red frames, as the person in the probe set was identified from the pedestrian. On the other hand, those images which were in green boxes in negative ranked list indicated that they are not the target to be identified. Through the application of this method, feature representation noises such as variation of viewpoints, a variation of illumination, a variation of weather and the light exposure, and the complex background had been solved.

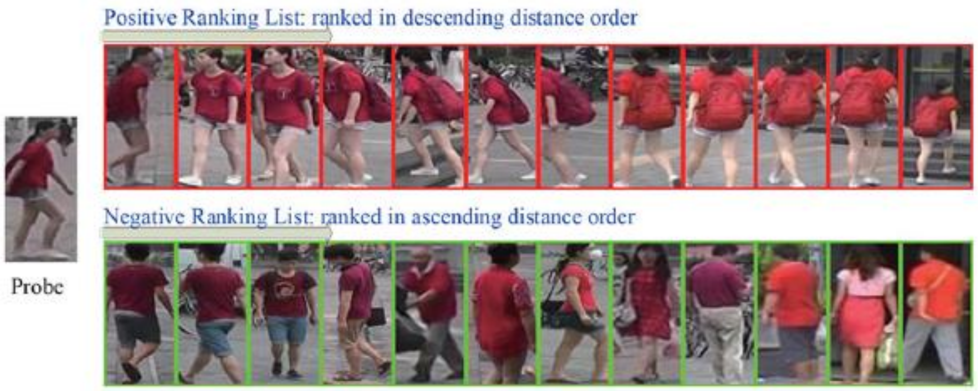

Positive Ranking List: ranked in descending distance order
Negative Ranking List: ranked in ascending distance order
Probe

Figure 2-11-1: Positive Ranking List and Negative Ranking List [16]

2.12 Summary of Problems and Solutions

After reviewing the research, all the research has 5 strategies. The strategies are categorized into pre-processing and augmentation, post-processing, scalability, architecture design and noise-robust for person reidentification. Pre-processing and augmentation generate of new poses and occlusion of body parts without additional cost. Post-processing is about the reranking and rank-fusion. For scalability, the researchers scale the efficient model and transfer the model for learning. Architecture design is divided into input-based architecture and customized module architecture.

Lastly, noise robustness is categorised into three categories. They are partial reidentification, reidentification with label noises and reidentification with sample noises. These are the similarities of the person reidentification research [8]. On the other hand, Table 2-12-1 is the summary of the challenges that the researchers faced and the solutions to solve the problems.

Table 2-12-1: Summary of Problems and Solutions.

Problems	**Solutions**
Face Processing Issues	- Faster Region-Based Convolutional Neural Network (Faster R-CNN)
Short Inter-Person Distance	- Cross-View Quadratic Discrimination Analysis (XQDA) - 3D Hypothesis Clustering for Unlabelled Movement Data of Multiple People
Low Resolution Videos	- Extended Global-Local Representation Learning Network (EGLRN) - Multi-Scale Adaptive Super-Resolution Person Re-Identification (MSA-SR PREID)
Large Domain Gap Between the Reidentification Images and Pose Estimator Images	- Supervised Non-Local Similarity (SNS) Learning
Semantic Gap	- Part-Based Attribute-Aware Network for Person Re-Identification (PAAN)
Large Intra-Class Variation	- Pseudo-Inverse LDA (PLDA)
Feature Matching Issues	- Improved Bag of Features (BOF) Based on Speeded Up Robust Feature Algorithm (SURF)
Time Consuming Ranking List Production	- Loopnet Ranking List with Multiplet Loss

Long Transition Time	- Adaptive Ranking Support Vector Machines (ADARSVMS)
Mutual Interference Between the Outputs	- Deep Group Shuffling Dual Random Walk with Label Smoothing

CHAPTER 3

SYSTEM METHODOLOGY/APPROACH

3.1 System Flowchart for the Application

In the application, the user uploads the images of the target one by one. After uploading all the images. The user sees the rows of images of the targets are displayed. This shows that the images are uploaded to the directory successfully. After uploading the images, user uploads the videos for person identification one by one. Due to the size of the videos being large, the application may take a longer time to process. Once a video is uploaded to the directory successfully, the application displays 'Video is uploaded successfully' on the screen. After all, videos are uploaded successfully, the application proceeds to target prediction, to predict a PID for all images of targets.

If the images are unable to be predicted, it proceeds to the person tracking using one of the videos. The images are cropped from the videos and saved into the directories. After that, the system retrains the model and predicts the person again. The images should be predicted with a PID. After prediction, the system proceeds to the person identification stage. On the other hand, if the images of targets are predicted successfully, the application proceeds to the person identification stage directly. Figure 3-1-1 shows the flowchart of the Person Identification Application.

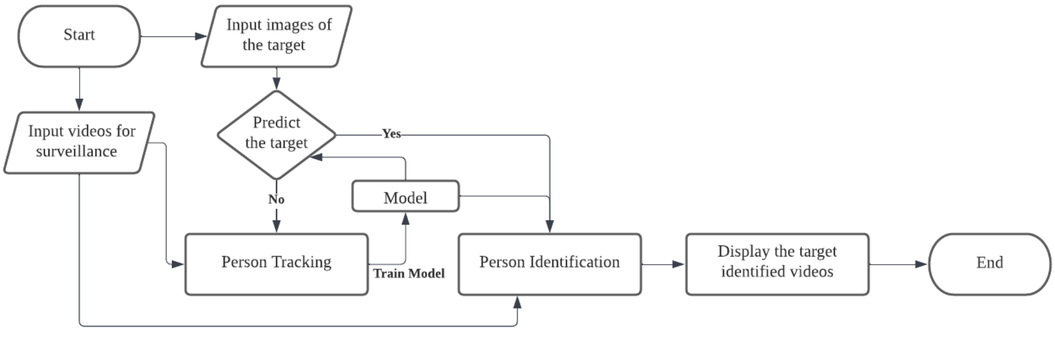

Start
Input images of the target
Input videos for surveillance
Predict the target
Yes
No
Model
Person Tracking
Train Model
Person Identification
Display the target identified videos
End

Figure 3-1-1: Flowchart of Person Identification Application

3.2 Use Case Diagram for Application

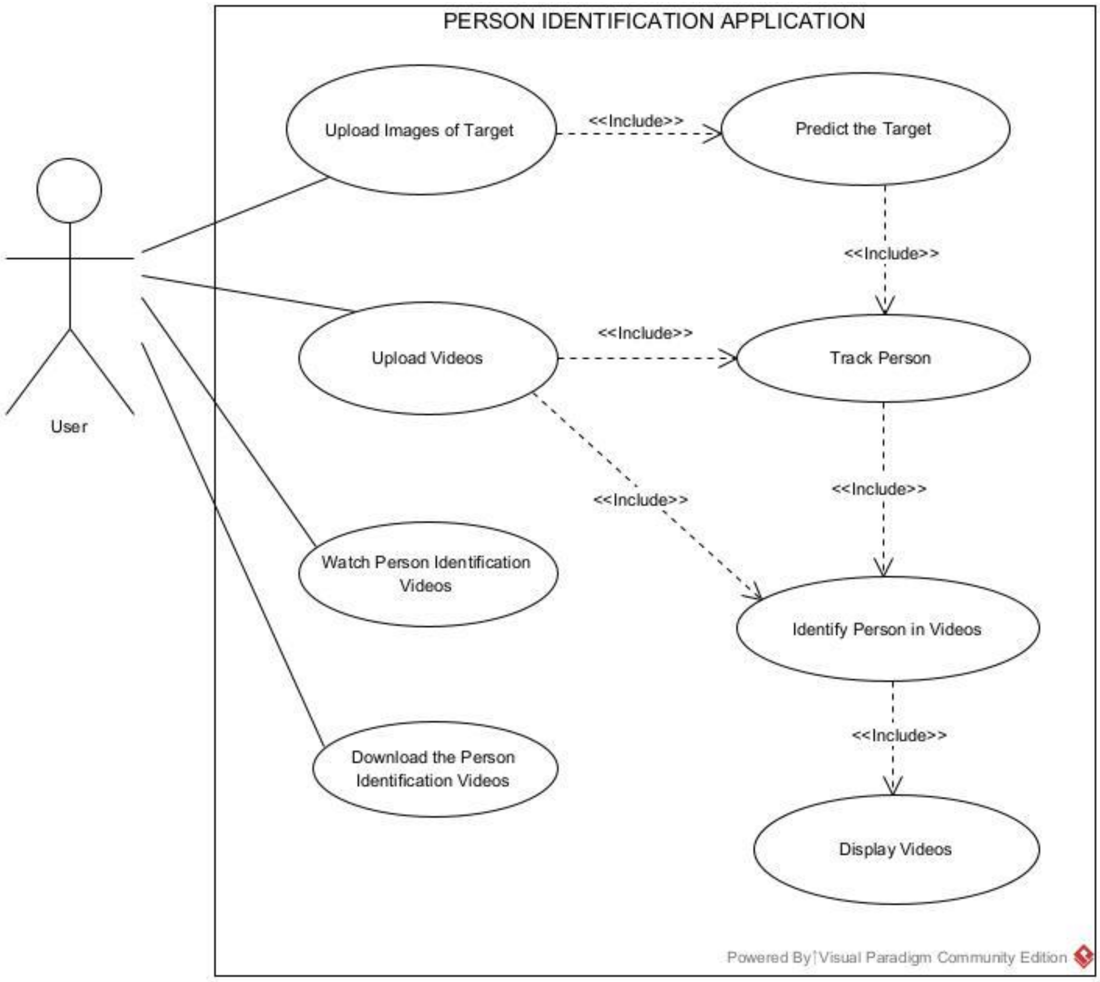

PERSON IDENTIFICATION APPLICATION
User
Upload Images of Target
<<Include>>
Predict the Target
<<Include>>
Upload Videos
<<Include>>
Track Person
<<Include>>
<<Include>>
Watch Person Identification Videos
Identify Person in Videos
<<Include>>
Download the Person Identification Videos
Display Videos
Powered By Visual Paradigm Community Edition

Figure 3-2-1: Use Case Diagram for Application

3.2.1 Use Case Description for Uploading Images

Table 3-2-1: Use Case Description for Uploading Images

Use Case ID	**UC01**
Feature	User uploads the image.
Purpose	To allow the user to upload image.
Actor	User

Trigger	User presses ‘Browse’ button to search the image and press ‘Submit’ button to upload the image.
Precondition	User has a very stable internet connection.
Main Flow	1. User presses ‘Browse’ button. 2. User chooses the image of target. 3. User presses ‘Submit’ button. 4. User repeats these steps for another 7 times. 5. System displays row of 8 images of target.

3.2.2 Use Case Description for Uploading Videos

Table 3-2-2: Use Case Description for Uploading Videos

Use Case ID	**UC02**
Feature	User uploads the video.
Purpose	To allow the user to upload video.
Actor	User
Trigger	User presses ‘Browse’ button to search the video and press ‘Submit’ button to upload the video.
Precondition	User has a very stable internet connection.
Main Flow	1. User presses ‘Browse’ button. 2. User presses ‘Submit’ button. 3. User chooses a video. 4. System displays video uploading status. 5. User repeats these steps for another 3 times.

3.2.3 Use Case Description for Watching Person Identification Videos

Table 3-2-3: Use Case Description for Watching Person Identification Videos

Use Case ID	**UC03**
Feature	User watches the person identification videos.
Purpose	To allow the user to watch person identification videos.
Actor	User
Trigger	User presses ‘Show Person Identification Videos’ button to watch the person identification videos concurrently.
Precondition	User has a very stable internet connection.
Main Flow	1. User presses ‘Show Person Identification Videos’ button. 2. User watches the four-person identification videos concurrently.

3.2.4 Use Case Description for Downloading Person Identification Videos

Table 3-2-4: Use Case Description for Downloading Person Identification Videos

Use Case ID	**UC04**
Feature	User downloads the video.
Purpose	To allow the user to download video.
Actor	User
Trigger	User presses ‘Download Video 1’ hyperlink or ‘Download Video 2’ hyperlink or ‘Download Video 3’ hyperlink or ‘Download Video 4’ hyperlink to download the person identification video.
Precondition	User has a very stable internet connection.

Main Flow	1. User presses 'Download Video 1' hyperlink or 'Download Video 2' hyperlink or 'Download Video 3' hyperlink or 'Download Video 4' hyperlink. 2. User watches person identification videos using his/her video player.

3.3 Activity Diagram for Application

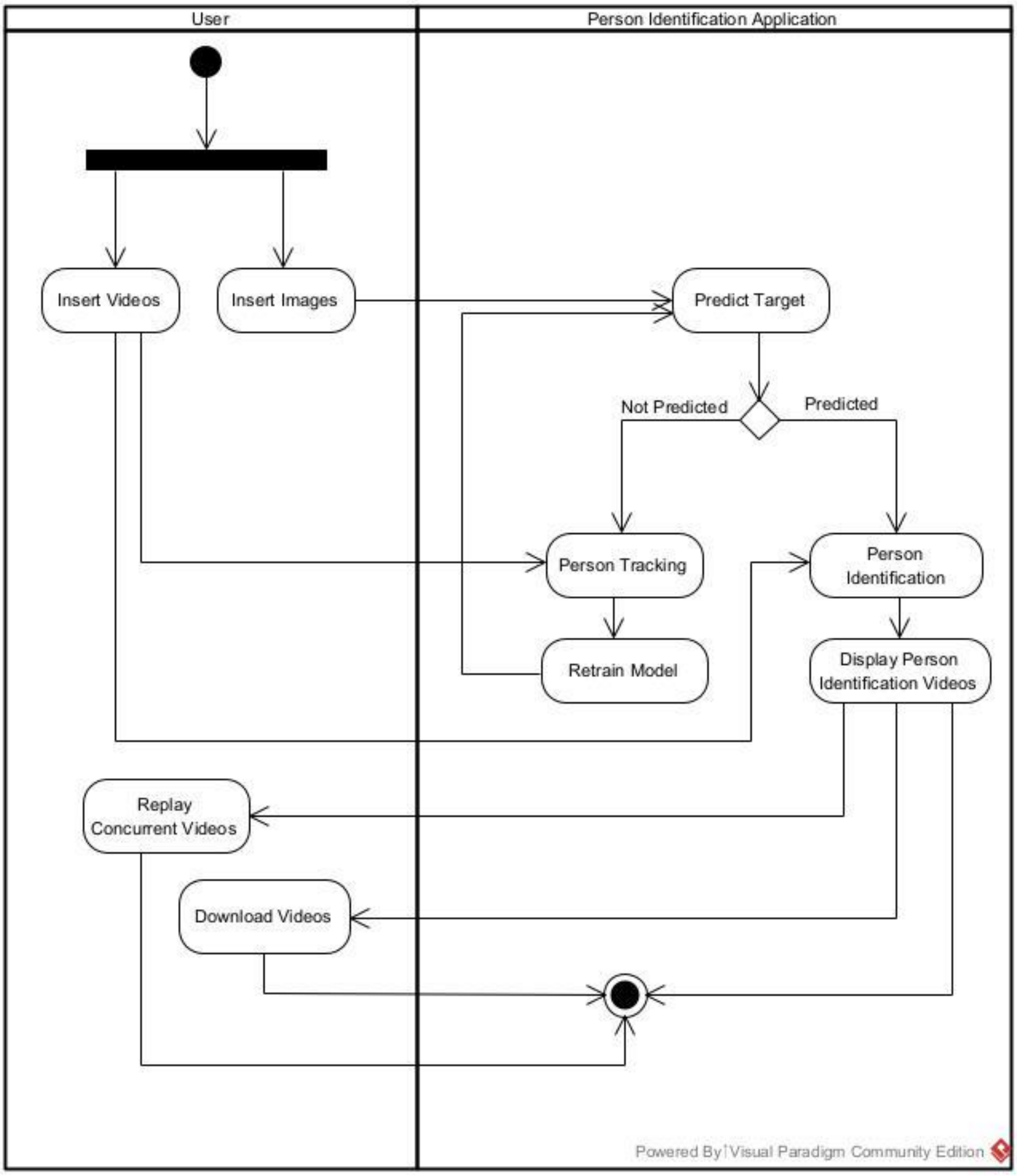
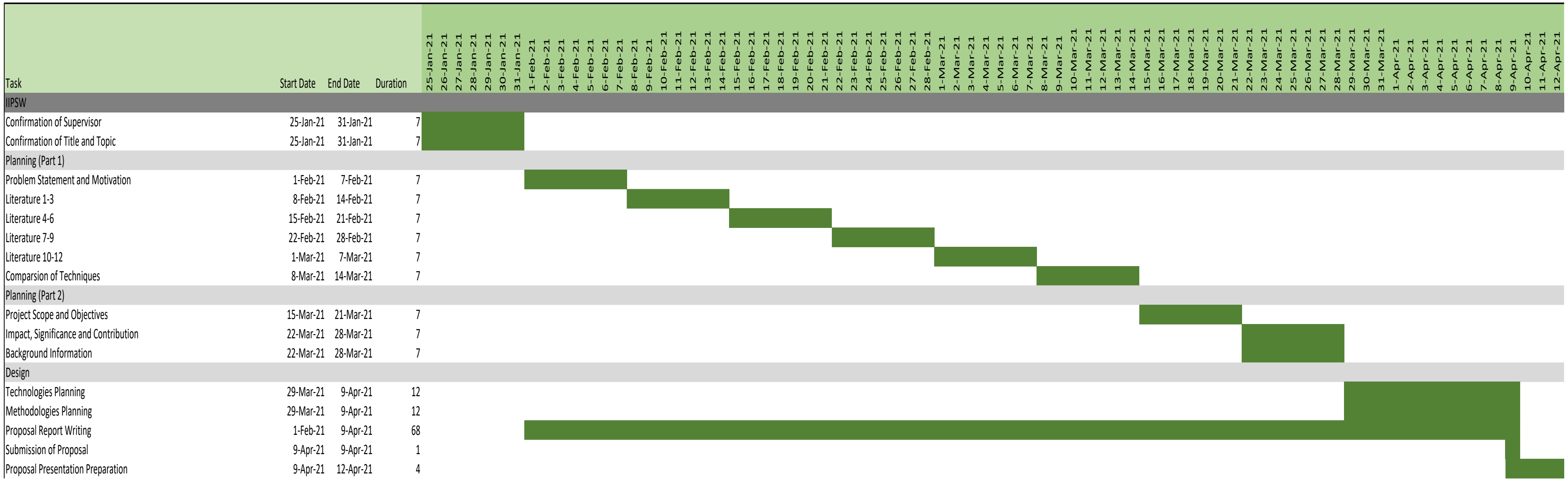


Figure 3-3-1: Activity Diagram for Application

3.4 Timeline

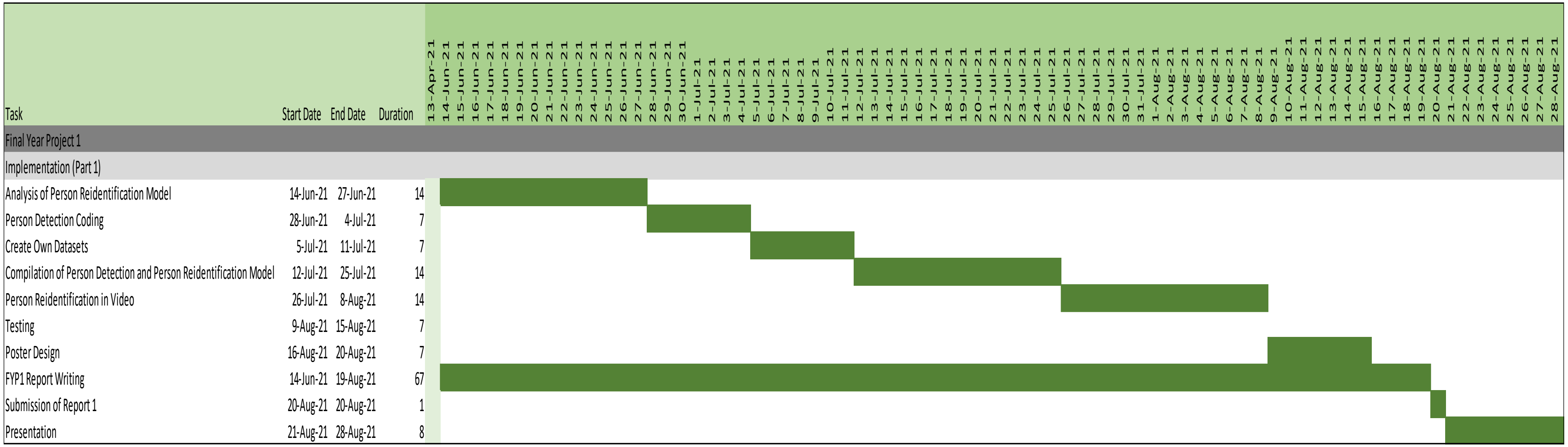
3.4.1 Proposal Writing Timeline



Task	Start Date	End Date	Duration
IIPSW			
Confirmation of Supervisor	25-Jan-21	31-Jan-21	7
Confirmation of Title and Topic	25-Jan-21	31-Jan-21	7
Planning (Part 1)			
Problem Statement and Motivation	1-Feb-21	7-Feb-21	7
Literature 1-3	8-Feb-21	14-Feb-21	7
Literature 4-6	15-Feb-21	21-Feb-21	7
Literature 7-9	22-Feb-21	28-Feb-21	7
Literature 10-12	1-Mar-21	7-Mar-21	7
Comparsion of Techniques	8-Mar-21	14-Mar-21	7
Planning (Part 2)			
Project Scope and Objectives	15-Mar-21	21-Mar-21	7
Impact, Significance and Contribution	22-Mar-21	28-Mar-21	7
Background Information	22-Mar-21	28-Mar-21	7
Design			
Technologies Planning	29-Mar-21	9-Apr-21	12
Methodologies Planning	29-Mar-21	9-Apr-21	12
Proposal Report Writing	1-Feb-21	9-Apr-21	68
Submission of Proposal	9-Apr-21	9-Apr-21	1
Proposal Presentation Preparation	9-Apr-21	12-Apr-21	4

Figure 3-4-1: Proposal Writing Timeline

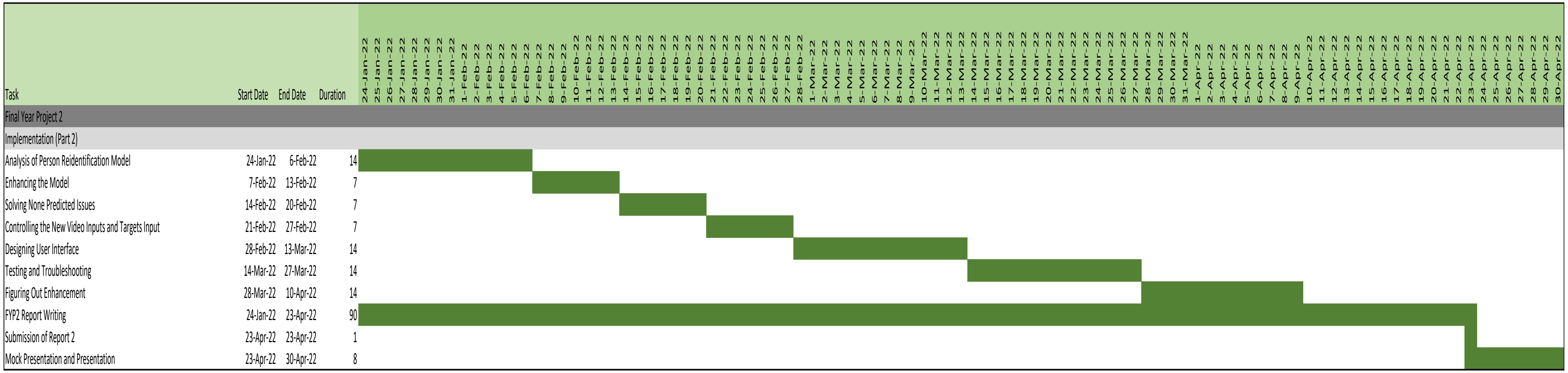
3.4.2 FYP1 Timeline



Task	Start Date	End Date	Duration
Final Year Project 1			
Implementation (Part 1)			
Analysis of Person Reidentification Model	14-Jun-21	27-Jun-21	14
Person Detection Coding	28-Jun-21	4-Jul-21	7
Create Own Datasets	5-Jul-21	11-Jul-21	7
Compilation of Person Detection and Person Reidentification Model	12-Jul-21	25-Jul-21	14
Person Reidentification in Video	26-Jul-21	8-Aug-21	14
Testing	9-Aug-21	15-Aug-21	7
Poster Design	16-Aug-21	20-Aug-21	7
FYP1 Report Writing	14-Jun-21	19-Aug-21	67
Submission of Report 1	20-Aug-21	20-Aug-21	1
Presentation	21-Aug-21	28-Aug-21	8

Figure 3-4-2: FYP1 Timeline

3.4.3 FYP2 Timeline



Task	Start Date	End Date	Duration
Final Year Project 2			
Implementation (Part 2)			
Analysis of Person Reidentification Model	24-Jan-22	6-Feb-22	14
Enhancing the Model	7-Feb-22	13-Feb-22	7
Solving None Predicted Issues	14-Feb-22	20-Feb-22	7
Controlling the New Video Inputs and Targets Input	21-Feb-22	27-Feb-22	7
Designing User Interface	28-Feb-22	13-Mar-22	14
Testing and Troubleshooting	14-Mar-22	27-Mar-22	14
Figuring Out Enhancement	28-Mar-22	10-Apr-22	14
FYP2 Report Writing	24-Jan-22	23-Apr-22	90
Submission of Report 2	23-Apr-22	23-Apr-22	1
Mock Presentation and Presentation	23-Apr-22	30-Apr-22	8

Figure 3-4-3: FYP2 Timeline

CHAPTER 4
SYSTEM DESIGN

4.1 System Design Concepts

In this development of person identification application, there were three main methodologies to be implemented, such as person detection, person tracking, and person reidentification. Before person detection, the inputs such as multiple concurrent surveillance videos from multiple angles and images of the targeted person were inserted. After the person detection, the person tracking and person reidentification, the targeted person would be bounded in green rectangle frame in every video that identifies the person, while the pedestrian who was non-target will be ignored. There were 2 scenarios, target predicted, or target not predicted. When the target was predicted, proceed to person identification directly. On the other hand, the target was not predicted, it proceeded to person tracking. Then, the model was trained again with the newly cropped images in Training directory after person tracking. After that, predict the target again, the target should be predicted with a PID. Same as before, proceed to person identification. In Figure 4-1-1 , it showed the overview of Person Identification Application.

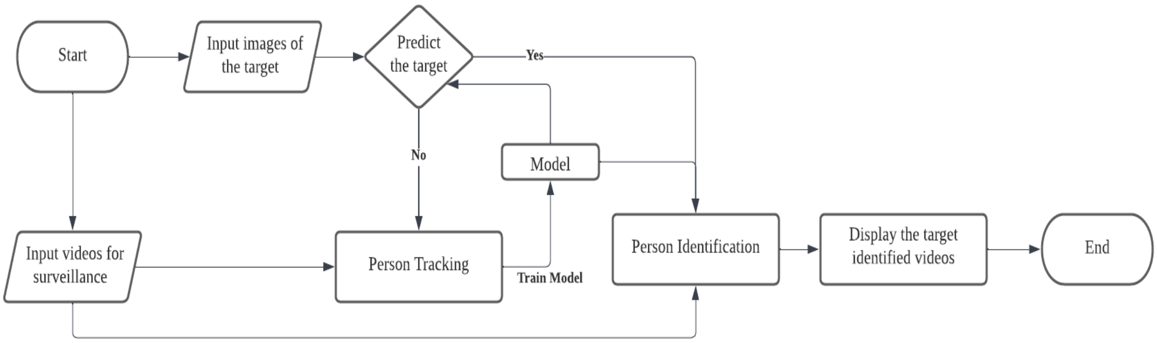

Start
Input images of the target
Predict the target
Yes
No
Model
Input videos for surveillance
Person Tracking
Train Model
Person Identification
Display the target identified videos
End

Figure 4-1-1: Overview of Person Identification Application

4.1.1 Person Detection: YOLOv3

First, person detection was implemented by applying the library You only look once Version 3 (YOLOv3). YOLOv3 is a common object detection for real time, that can detect objects in videos or images accurately. According to [21], YOLOv3 was enhanced from version one and version 2 in 2018. YOLOv3 was written based on Darknet, which is an open-source neural network algorithm. It implements a deep convolutional neural network (CNN) in the environment of Keras. Coincide with the name YOLO, the prediction layer has 1 x 1 convolutions as shown in Figure 4-1-2 , and the size of the predicted map is the same as the size of the previous feature.

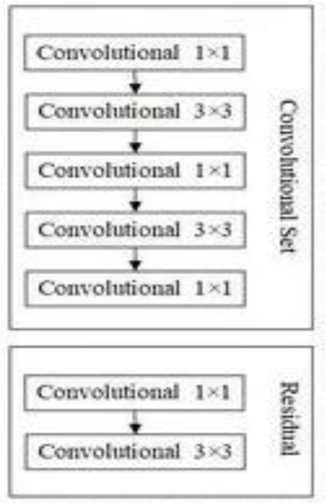


Figure 4-1-2: Architecture of YOLOv3 [21]

YOLOv3 algorithm breaks down an image into grids. When an object is positively detected, each of the grid cells predicts the bounding boxes that surround the positive detected object, which has high confidence of the predefined classes. It works similarly to R-CNN, such as Fast R-CNN, Faster R-CNN, and Mask R-CNN. The difference between YOLO and other R-CNN is that YOLO does bounding box regression and classification at the same moment. In addition, when computation time is the concern, YOLOv3 is better than Faster-RCNN. Faster R-CNN, which implements SVM for region classification, has higher accuracy than YOLOv3, but Faster R-CNN requires more computational time and greater network memory. Therefore, YOLOv3 was chosen instead of Faster R-CNN, for real time detection.

In YOLOv3, there are few important terminologies such as confidence, anchor box, and Non-Maximum Suppression. As mentioned before, confidence is the measure for the positively detected object with the predefined classes. To predict and detect in

YOLOv3, it is based on the similarity towards the predefined classes, with the scientific term of confidence. The higher the confidence, the higher the accuracy of a positive detection. Confidence is divided into 2, such as class confidence and box confidence in YOLOv3. When the object is detected, form a bounding box on the object. In the bounding box, it has an x-coordinate (x) and y-coordinate (y) of the left-upper corner of the bounding box, width (w) of the bounding box, height (h) of the bounding box, and the box confidence score. The box confidence score shows the accuracy of the bounding box. Through these characteristics, the bounding box of the detected can be visualised. In addition, class confidence score is calculated using the class confidence formula. It is the multiply of bounding box confidence with conditional class probability. The conditional class probability is the probability of the detected object within a specific class. In Figure 4-1-3 , it showed how does the image look like after computing the class probability with pixel grids. There are few detected classes such as car, road sign, tree, traffic light, sky, and background [22].

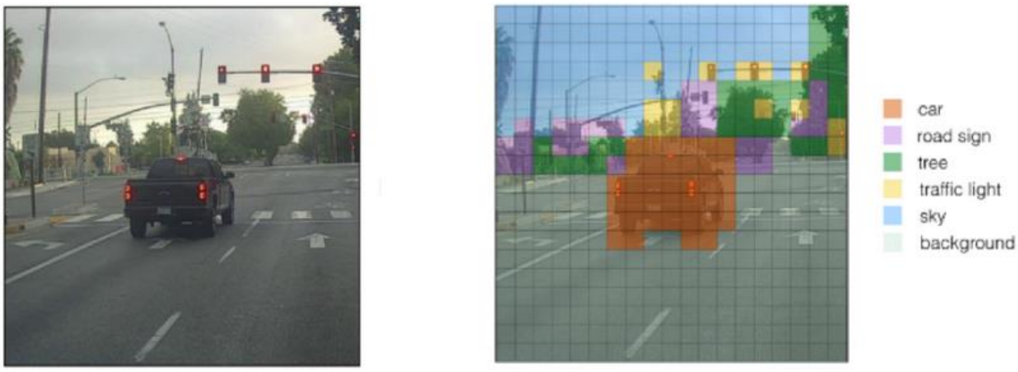


Figure 4-1-3: Before and After Computing Class Probability [23]

Formula of Class Confidence:

$$Class\ Confidence = Bounding\ Confidence * Conditianal\ Class\ Probability$$

Furthermore, the anchor box, which is known as the bounding box predicts the log-space transform. The log-space transforms are default bounding boxes that are predefined. It will be applied to the anchor box for prediction. Particularly, YOLOv3 has three anchors, which means that it has three bounding boxes per neuron. Neurons are described as the features for detection in the model architecture. In addition, Non-Maximum Suppression (NMS) prevents multiple detection situations, by passing them if they are not detected accurately. When more than one bounding box detect an object as a positive detection multiple times, non-maximum suppression works by eliminating the overlapping bounding box with the highest-class probability through Intersection over Union (IoU). Then, the IoU which has greater value than the IoU threshold, will be rejected, as it is overlapping to other bounding boxes. IoU is the division of intersection of two bounding boxes by the union of two bounding boxes, as shown in Figure 4-1-4 . When IoU has greater value than the threshold, the bounding box will be accepted as positive detection. The greater the IoU value, the greater the accuracy of the correct detection. However, there will be more than one positive detection on an object. Hence, NMS is applied to reject other overlapping bounding boxes, and accept the only bounding box with maximum score. For better understanding on NMS, refer Figure 4-1-5. When there were multiple positive detections, the figure on the left showed bounding boxes without NMS, while the figure on the right showed the bounding after NMS. Only the bounding box with highest score will be selected as the correct detection, even other IoU bounding boxes is greater than the threshold, they will be discarded.

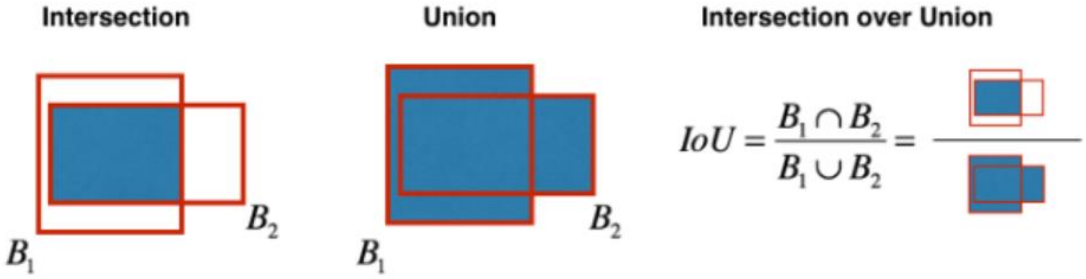


Figure 4-1-4: Intersection over Union [23]

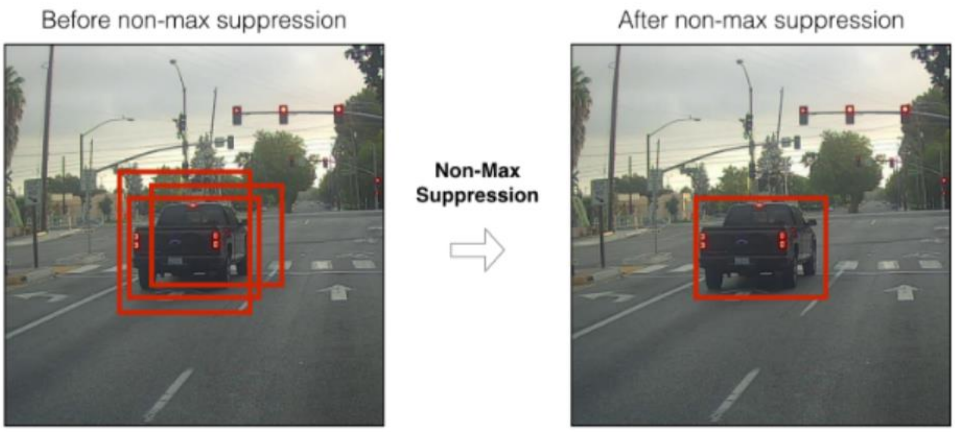
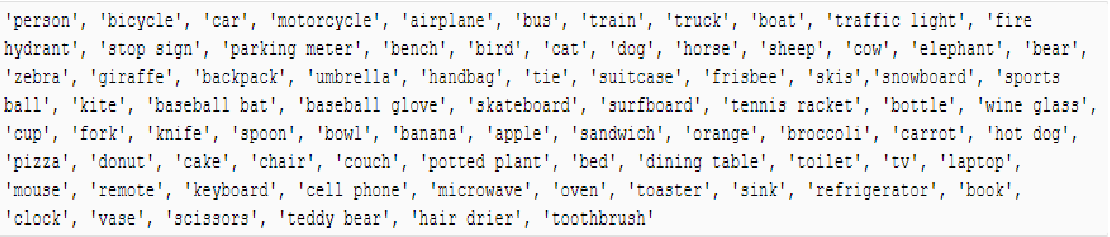


Figure 4-1-5: Bounding Boxes Before and After NMS [23]

In YOLOv3, there are two very important files, which are ‘coco. names’ and ‘yolov3.weight’. In Coco dataset, it consists of all the categories that can be detected by YOLOv3, such as a person, bicycle, car, and more. It has a total of 80 categories, as shown in Figure 4-1-6 . It means that YOLOv3 can detect everything that is listed in ‘coco.name’. In this application, the person is our target detection. Therefore, the person class was detected only, and other detected objects were ignored, even they are detected. In the last step, only those prediction boxes with high confidence scores will be kept as final predictions.



```
'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire
hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear',
'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis','snowboard', 'sports
ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass',
'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog',
'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop',
'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book',
'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
```

Figure 4-1-6: Classes in YOLOv3

4.1.2 Person Tracking: DeepSORT

After the person detection, person tracking with the DeepSORT algorithm comes next. To differentiate between person detection and person tracking in the video, a tracker tracks and assigns an identifier for the object tracked, while a detector detects the object only. During tracking, each frame has multiple persons to be tracked. To deal with this challenge, the algorithm has two main steps, which are detection and association were implemented. [24]

The feature vector is known as appearance descriptor. To obtain the vector of features, build a classifier for dataset training, until an optimum training accuracy and training loss, and strip into the final classification layer. And the last step flattens the feature vector into a single feature vector in a dense layer.

Nearest neighbour algorithm is the simplest object classifier to be implemented in DeepSORT. The speciality of the nearest neighbour algorithm is that no assumption on the dataset, it means that it does not need to create a learning model for it to learn and train. It is implemented to establish the association for tracking detects, through measuring the distance and similarities between detects. A nearest neighbour distance metric returns the closest or nearest distance for each target to be observed and tracked. Although it is simple and common, but it works well [24].

There are few different distance metrics to be implemented in Nearest Neighbour such as Manhattan distance metric, Euclidean distance metric and Cosine distance metric. Manhattan distance metric, which was considered in nineteenth century, performs the sum of the absolute differences between the vectors, then square root them, as shown in Figure 4-1-7. On the other hand, Euclidean distance performs the sum of the square differences between the vectors, then square root them, as shown in Figure 4-1-8. Cosine Distance metric is derived from the cosine similarity, which is the angle between two vectors. After that, it obtains the cosine distance by subtraction of the cosine similarity from one [24], as shown in Figure 4-1-9. In brief, it calculates the distance between the two detects with the cosine angle, in DeepSORT. Among three distance metrics, Cosine distance performs better than others, as it has an extra consideration, which is the angle between vectors. Commonly, the cosine similarity is applied in the text data. But it is applied in our project for considering appearance information, which is useful for identities recovering, especially when there are some occlusions or less differentiable motion. The formulas of distance metrics were shown as below [25].

$$MD(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Figure 4-1-7: Manhattan Distance Formula [24]

$$ED(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Figure 4-1-8: Euclidean Distance Formula [24]

$$CosD(x, y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Figure 4-1-9: Cosine Distance Formula [24]

However, DeepSORT does not overcome the switching of ID between different detected frames, when the person is close to each other. For instance, when two detected objects are detected closely, they will be switched implicitly. Therefore, the identifier will be assigned to different targets after they are close to each other.

4.1.3 Person Identification: Self-Implemented Model

4.1.3.1 Reason of Implementation of Self-Implemented Model

[27] built a software person reidentification library, Torchreid in the PyTorch environment across multiple different angle cameras. In this library, Torchreid provides a general-purpose data loader which supports as many as 15 different datasets, such as Market-1501, CUHK-03 and more, for image and video datasets. Torchreid implements the re-identification CNN architecture which is reproducible for future research.

Along with the Torchreid library, train the training datasets and testing datasets from the person tracking system. After that, the images of the person to be identified were allocated in the query set. For good practice, the person of the image to be identified should be from different cameras, which has a different angle. Each camera should at least have one image for the person identified for model training.

Before model training, choose a model architecture that performs the best is important. Therefore, three different types of model architecture such as ResNet-34, ResNet-50, and DenseNet-121 were trained with the Market-1501 datasets. This step was to make sure that the most suitable model architecture will be chosen, before implementing the model in the application. In the end, Resnet-50 was chosen as the model, as it performed the best among these three models.

However, the Torchreid library is suitable for image person identification, which is more suitable for research purposes, but not for application or business purposes. In addition, the Torchreid library consumes a huge amount of GPU memory, which requires an external powerful GPU server to support it. The application requires video person identification and tracking the movement of the person, this step requires huge memory. If the model requires a huge memory, the server will be crashed. Due to the application does not support cross-platform model training, it is not suitable for the application. Therefore, the implementation of the CNN model with Tensorflow, which does not consume much memory, was implemented.

There are few terminologies such as data augmentation, convolutional layers, max pooling, ReLU activation, dropout, flatten, dense layer, Sparse Categorical Cross Entropy, Softmax and Adam Optimizer.

4.1.3.2 Data Augmentation

Data Augmentation is a technique to generate more positive samples for model training. It is divided into two categories, such as geometric methods, and photometric methods. Geometric transformation alters the image geometry, by mapping the individual pixel values to a new image. Examples of geometric transformation are horizontal flipping, vertical flipping, cropping, rotation, and more. Photometric transformation alters the RGB according to the heuristics. Examples of photometric transformation are intensity shift, colour, jitter, PCA, and more. Both transformations can be combined to form a new image. In this application, it implemented geometry transformation only. Figure 4-1-10 showed an example of the data augmentation, it is rotated and cropped, to produce more positive datasets, to be trained in the model.

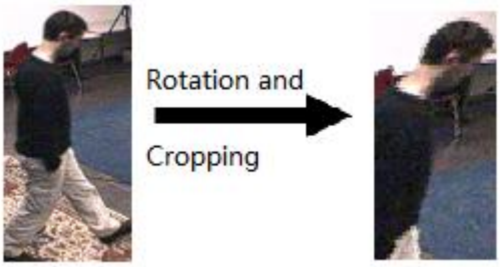


Figure 4-1-10: Examples of Data Augmentation

4.1.3.3 Convolutional Layers

Convolutional layers contain a set of filters that can be trained for different feature detection. Each filter will generate an output channel which is known as activation map or feature map, according to the filter. The following is the equation to calculate the output value from the input with the model parameter.

Z: output value (scalar)

W: filter value (vector)

X: input value (vector)

b: model parameter (scalar)

$$Z = W * X + b$$

The output value can be controlled by filter, stride, and padding. Filters controlled the output size. When the padding and stride remain the same, the greater the filter size, the smaller the activation map. Stride controls the number of steps that the filter moves. The filter slides across the input 2 steps when the stride is set as 2. The larger the stride, the smaller the activation map. When the filters move across the inputs, the spatial dimensions were reduced with layers, as the receptive field shrinks too fast, the model cannot be trained deeper or better results. Therefore, padding is the border added to the input. Commonly the padding value is zero. Figure 4-1-11shows how does the padding work. The input size is (3, 3). However, if the filter (3, 3) is applied to the input, the input will be washed away, and the output size is (1, 1). With the presence of padding=1, the input size is (5, 5). After the convolution with the (3, 3) filter, the output size remains the same as the input size.

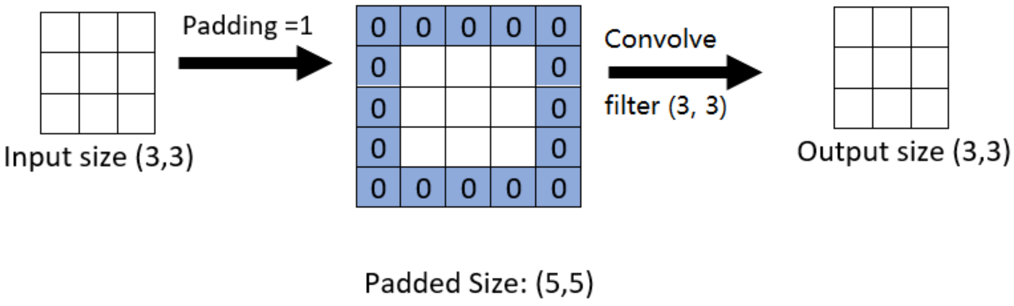


Figure 4-1-11: Padding

The output size is known as the size of the activation map. It can be predicted and calculated using the equation below. The reason that the padding is multiplied by, is that the input is padded for both sides.

o: output size

p: padding

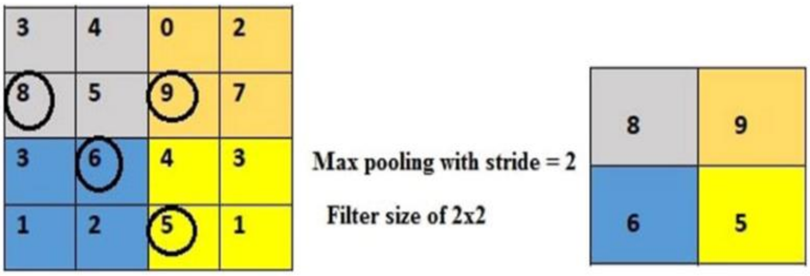
f: filter size

s: stride

$$o = \frac{i+2p-f}{s} +1$$

4.1.3.4 Max Pooling

Max pooling is the maximum element in the matrix from the input matrix. The input is filtered by a fixed filtering stride, which is the number of pixels that change their movement each time. Max pooling reduces the resolution of the feature, as shown in Figure 4-1-12.



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

8	9
6	5

Figure 4-1-12: Max Pooling [27]

4.1.3.5 ReLU Activation

After the summation phase, it is time for the activation phase. During the activation phase, there are choices of activation function such as ReLU, sigmoid, tanh, Leaky ReLU, Parametric ReLU and more. In CNN model, ReLU is chosen instead of the sigmoid function, which causes the poor local minima, because ReLU has less vanishing effect. Sigmoid function has a high risk of gradient varnishing, as saturated neurons vanish. As shown in the Figure 4-1-13, when σ (z=-10) =0 and σ (z=10) =1, the neurons are saturated. For derivatives of sigmoid, σ'(z=-10) =0 and σ'(z=10) =0, the neurons are saturated too. When the z=0, as its σ(z=0) =0.5, this is the point that exceptionally not saturated in the sigmoid activation function, as its derivative is σ'(z=0) =0.25. It is proven that sigmoid function suffers from the gradient varnishing. Hence, the ReLU is chosen as the activation function.

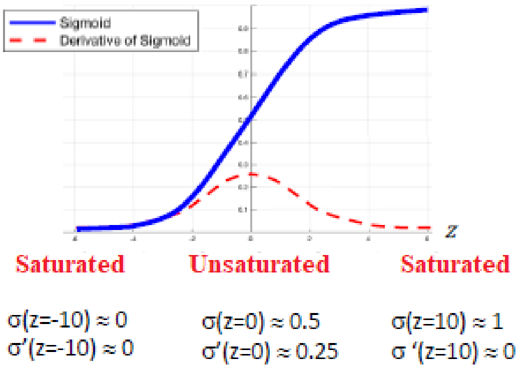


Figure 4-1-13: Sigmoid Function Graph

4.1.3.6 Dropout

Dropout is set, to maintain the simplicity of the network during training. It avoids co-adaptation of neurons, which compensates for the weakness of neurons in the model., In Figure 4-1-14, it showed the neural network with and without dropout. Due to dropout, batch normalization is not implemented, as batch normalization does not work well with the presence of the dropout. Dropout is applied when overfitting occurs. [29]

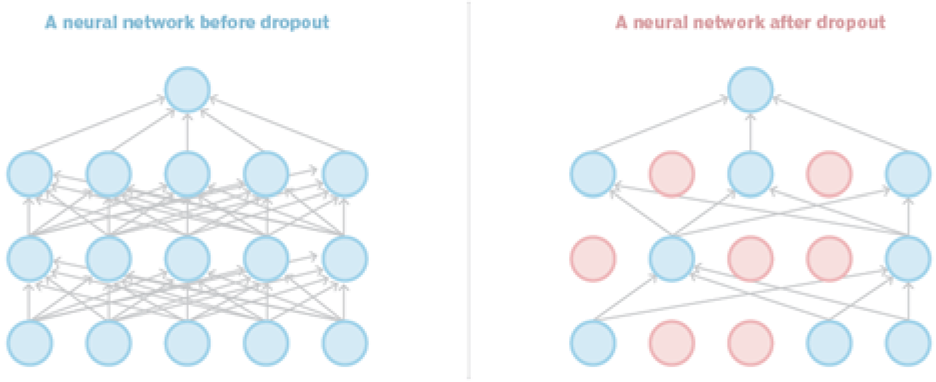

A neural network before dropout
A neural network after dropout

Figure 4-1-14: Neural Network with and without Dropout [29]

4.1.3.7 Flatten

The function of Flatten is to flatten the multi-dimension input tensor into a single dimension. With the presence of Flatten, the input layer can be passed through every neuron in the model efficiently. For instance, in Figure 4-1-15, the input size is (5, 3), the dense layer produces an output of 16 dimensions. To allow the input to be implemented in the dense layer, Flatten is necessary. Through the flattening process, the input size is (15,), it can be implemented in the dense layer of 16 dimensions, to produce an output with the size of (16,).

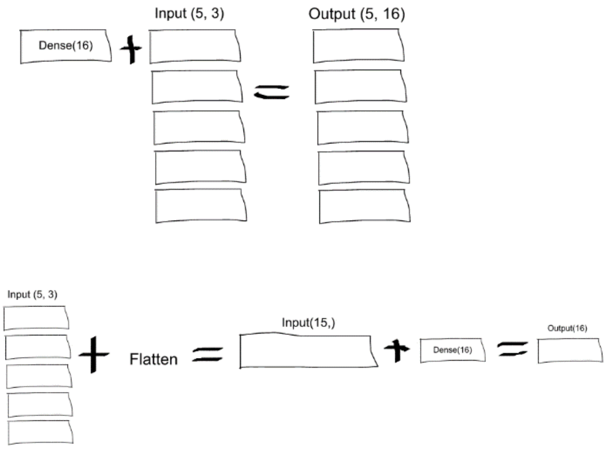


Figure 4-1-15: Examples of Flatten

4.1.3.8 Dense Layer

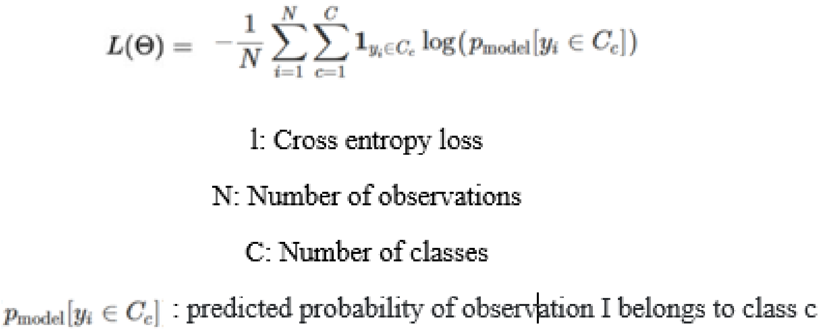
Dense layer is known as the fully connected layer. The fully Connected (FC) layer is defined as every node in the previous layer will be fully connected to all nodes into the next layers. It compiles and connects the output of the previous layer with every node (features), to form a final output. The main advantage of the fully connected layer is to remove the spatial dimension by the flattening technique. Both the convolutional layer and dense layer retrieve the output from the previous layers corresponding to all its neurons. The main difference between the convolution layer and the dense layer is that convolutional layer is non-linear operation, while the dense layer is a linear operation.

4.1.3.9 Adam Optimization

Adam optimization applies Stochastic Gradient Descent (SGD) as its optimization gradient algorithm. It joins both RMSProp and AdaGrad algorithms for optimization which handle the sparse gradient on the noises. RMSProp is an unpublished adaptive optimizer with the root mean square, to learn the learning rate for the gradient in different weights. On the other hand, AdaGrad is a stochastic optimization algorithm, which does not tune the learning rate manually. It is commonly applied on unequal weight scaling. Therefore, Adam is a better optimization, as it combines the advantages of AdaGrad and RMSProp algorithms. [30]

4.1.3.10 Sparse Categorical Cross Entropy

Sparse Categorical Cross Entropy is suitable for one-hot encoded labels only. Furthermore, the classifier applies the sparse categorical cross-entropy, to categorize the inputs. It will be faster as it speeds up the computational process, compared to the categorical cross-entropy.[31] The formula of sparse category cross-entropy loss, as shown in Figure 4-1-16.



$$L(\Theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbf{1}_{y_i \in C_c}\log(p_{\text{model}}[y_i \in C_c])$$

l: Cross entropy loss

N: Number of observations

C: Number of classes

$p_{\text{model}}[y_i \in C_c]$: predicted probability of observation I belongs to class c

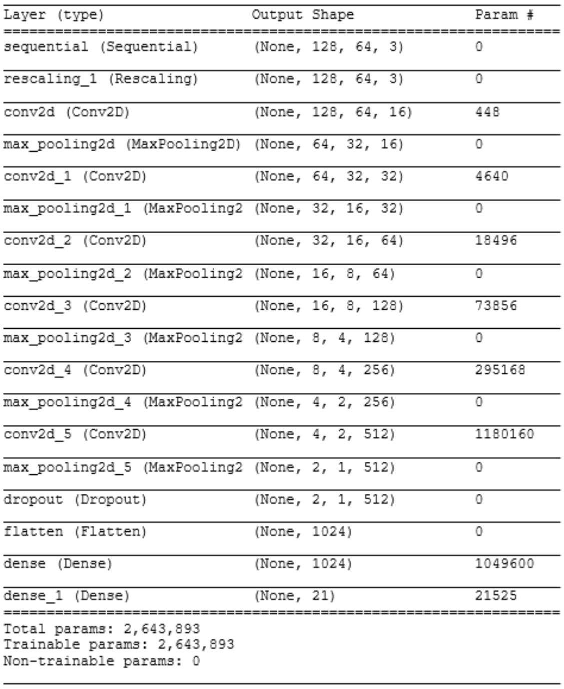
Figure 4-1-16: Sparse Category Cross Entropy Loss Function [29]

4.1.3.11 Softmax

Softmax converts the scores of each category into a probability distribution [32]. For example, there are 20 classes in the Training datasets, then there will be 20 Softmax classes with an individual probability. The total probability of all classes in Softmax will be '1'. It is suitable for classification with multiple labels such as person reidentification, as different pedestrians will be identified into different PID.

4.1.3.12 Summary of the Model

In Figure 4-1-17, it showed the summary of CNN model. The CNN model consisted of 6 convolutional layers and 2 dense layers with parameters. Before the inputs were passed to the convolutional layers, the inputs with an input size of (128, 64, 3), were grouped into a linear stack of layers by 'Sequential'. Then the linear stacks of layers are rescaled by 255, as the highest value of colour is 255. In total, there were 6 convolutional layers. The number of filters in the first layer was 16, 32 filters in the second convolutional layer, 64 filters in the third convolutional layer, 128 filters in the fourth convolutional layer, 256 filters in the fifth convolutional layer, and 512 in the last convolutional layers. Each convolutional layer had the same padding, their activation function was ReLU, followed by a 2D max pooling. After that, a dropout of 0.2 was implemented, to maintain the simplicity of the model. After dropping, the inputs were flattened, and to be passed to the dense layer with 1024 dimensions. The last step is to pass the 1024 dimensions inputs into the dimension, which is the same as the number of classes in the model. After that, the model was ready to be compiled with optimizer Adam and the sparse categorical Cross entropy.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential (Sequential)      (None, 128, 64, 3)        0
_________________________________________________________________
rescaling_1 (Rescaling)      (None, 128, 64, 3)        0
_________________________________________________________________
conv2d (Conv2D)              (None, 128, 64, 16)       448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 32, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 32, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 16, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 16, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 8, 64)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 8, 128)        73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 4, 128)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 4, 256)         295168
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 2, 256)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 4, 2, 512)         1180160
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 2, 1, 512)         0
_________________________________________________________________
dropout (Dropout)            (None, 2, 1, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              1049600
_________________________________________________________________
dense_1 (Dense)              (None, 21)                21525
=================================================================
Total params: 2,643,893
Trainable params: 2,643,893
Non-trainable params: 0
_________________________________________________________________
```

Figure 4-1-17: Summary of Model

However, CNN model experienced the gradient exploding and gradient varnishing. Gradient varnishing happens during the propagation when the gradient decreases dramatically during the backpropagation. In the end of backpropagation, the gradient remains unchanged or minuscule. Therefore, the model is unable to compute and update the model parameters (w and b) with the effective value, the model is stuck at the poor local minima, causing ineffective deep learning by the model. It leads to overfitting and leads to higher training errors. Gradient exploding happens when the model has large parameters updates, caused by the large gradient accumulation. This scenario causes the model to be unable to learn from the training data effectively. In Figure 4-1-18, the model suffers performance degradation especially in the lower layer, as the update of gradients in the lower layers are miniscule. On the other hand, the model which is shown in Figure 4-1-19, suffers from performance degradation, as the gradients in the lower layers are becoming larger, during back propagation. To solve the performance degradation in the model, ResNet should be implemented and applied as the training model. However, due the restriction of GPU memory, CNN model was implemented. Therefore, Resnet model will be the future implementation, when the GPU processing power is sufficient.

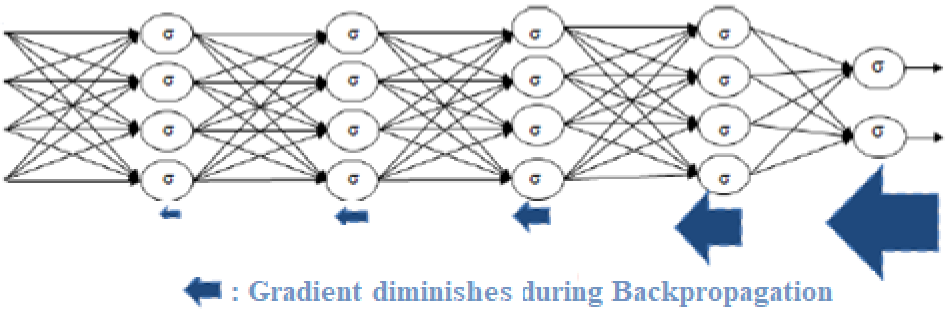


Figure 4-1-18:Gradient Varnishing

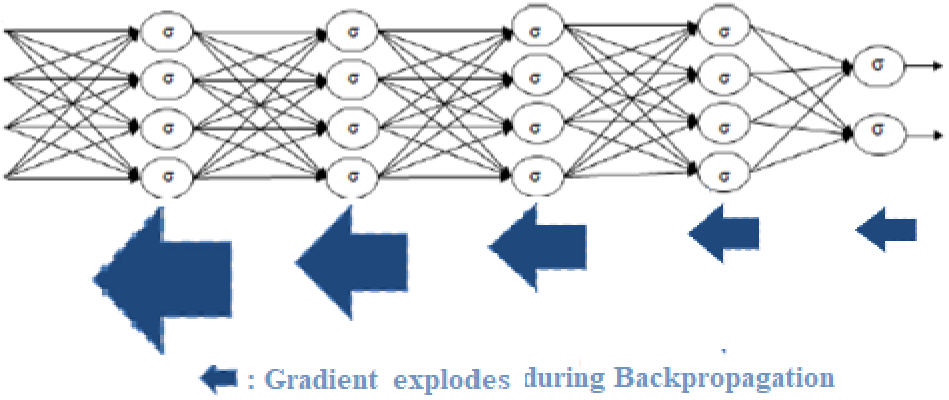


Figure 4-1-19: Gradient Exploding

4.2 System Design Procedures

4.2.1 Person Identification Library

In the first stage of the project, searching for a suitable person identification library for the application. Torchreid library was chosen as the person reidentification library. This person reidentification library, TorchReid was developed by [27] aiming to reidentify people in multiple videos with different angles. This library which was built on PyTorch allows model training and evaluation of the deep reidentification model. The greatest advantage of this library is that it supports 15 reidentification benchmark datasets for images or videos. In this person identification project, although the training, testing, and query datasets would be collected and retrieved on their own, it was better to train the common datasets with the models along with their architecture, to understand better on this Torchreid library [27]. However, it was not suitable for video person identification as it consumed a very huge amount of GPU memory, and it did not support video person identification but supported image person identification only.

4.2.2 Building Own Model for Person Identification

With the reference of the Torchreid library, a CNN model for person identification was built, it required a very low memory requirement. The number of convolutional layers of CNN model was not as much as Resnet-50 model in the Torchreid library. The more the convolutional layers, the more the number or parameters, the more the memory requirement needed. In Figure 4-2-1, the model worked well on the training set, with an accuracy of 0.9957 and a loss of 0.0165, and it worked well on the validation sets, with an accuracy of 0.9908 and a loss of 0.0379. However, other simpler model with slightly less accuracy required lesser time. When there were new videos input or target images, the model will be retrained. Hence, the model with lesser training time was preferable.

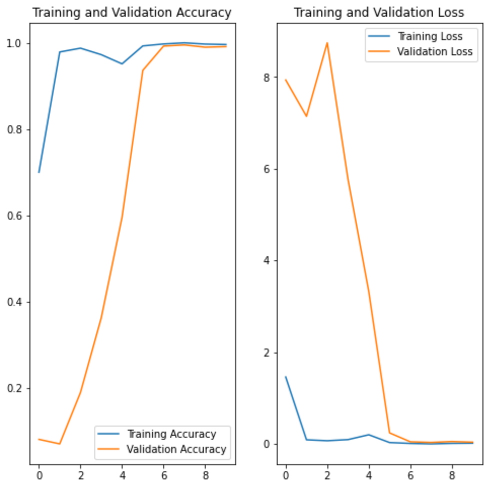


Figure 4-2-1: The Accuracy and Loss of Training and Validation for Resnet-50 Model

Before choosing the model with 6 convolutional layers, the model with 4 convolutional layers with a dense layer of 256 dimensions and the model with 5 convolutional layers with a dense layer of 512 dimensions were trained. However, the CNN model consisting of 6 convolutional layers with a dense layer of 1024 dimensions performed the best, the model had no overfitting or underfitting issues. Another reason that the model with 6 convolutional layers was chosen, was that the number of training datasets increased as new videos and images of the target were uploaded to this application. This avoided underfitting issues.

Table 4-2-1 showed the training accuracy, training loss, validation accuracy, and validation loss for all trained models for the best model selection. According to the Table 4-2-1, all the models did not experience overfitting issues and underfitting issues. Resnet-50 model performed the best in terms of training accuracy and validation accuracy. However, the training duration for Resnet-50 was too high. Compared with the 6 convolutional layers model, the duration ratio between both was 5.6, which 635 s was divided by 114s. Due to the high ratio (almost 6 times greater) of that, 6 convolutional layers model was chosen, as the performance in terms of training accuracy and validation accuracy was the best, excluding the Resnet-50 model.

Table 4-2-1: Summary of All Models

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Duration(s)
Resnet-50 (10 eps)	0.9957	0.0165	0.9908	0.0379	635
Model with 4 convolutional layers	0.9761	0.0749	0.9450	0.0488	66
Model with 5 convolutional layers	0.9794	0.0700	0.9738	0.1394	75

Model with 6 convolutional layers	0.9833	0.0580	0.9817	0.0532	114

Table 4-2-2 showed the summary of models without dropout and with dropout. In this application, a dropout of 0.2 was implemented in the CNN model, to maintain the simplicity of the model. The training accuracy of the model without dropout was the worst and the model training duration was the longest among all models implemented with dropout, as the model was not thinned or reduced during model training. The greater the dropout value, the faster the model training. Comparing the models with a dropout of 0.2 and dropout of 0.4, the difference of training accuracy and validation accuracy of the model with a dropout of 0.4 was slightly higher than 0.2, it proved that model with a dropout of 0.2 was more stable. It was the same as the training loss and the validation loss. Furthermore, the training accuracy of the model with a dropout of 0.6 was the worst among those models with dropout, as some useful neurons were dropped during model training. Dropout of 0.6 was too vigorous, and there were too much of inputs in the neurons layers were dropped. Therefore, a dropout of 0.2 was the optimum dropout to be implemented.

Table 4-2-2: Summary of Models with 6 convolutional layers Without Dropout and With Dropout

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Duration(s)
Model Without Dropout	0.9772	0.0615	0.9798	0.0679	123
Model With Dropout 0.2	0.9833	0.0580	0.9817	0.0532	113
Model With Dropout 0.4	0.9835	0.0470	0.9765	0.0868	108
Model With Dropout 0.6	0.9798	0.0679	0.9885	0.0534	106

The model of 6 convolutional layers with 0.2 dropout was chosen. In the first convolutional layers, it had 16 filters. In the second convolutional layer, it had 32 filters. In the third convolutional layer, it had 64 filters. In the fourth convolutional layer, it had 128 filters. In the fifth convolutional layer, it had 256 filters. In the sixth convolutional layer, it had 512 filters. Each layers have same padding, RELU activation function and 2D max pooling. After flattening, the dense layer had a dimension of 1024, and the output size will be according to the number of classes in the data directory.

4.2.3 Target Prediction

After receiving the inputs such as eight images of targets and four concurrent videos from the users, the system would predict those eight images, with the pre-trained model. If the person in the eights images was predicted as 'None', it would be proceeded to video tracking, for image retrieval from one of the videos only, and the model would be retrained according to the new dataset in the Training directory. After model retraining, the eight images were predicted again. It could be predicted correctly with a PID assigned for this target. On the other hand, if the person in the eights images was not predicted as 'None', it proceeded to person identification directly, without proceeding to the person tracking as previous. Figure 4-2-2 showed the flowchart of the process for initial person prediction.

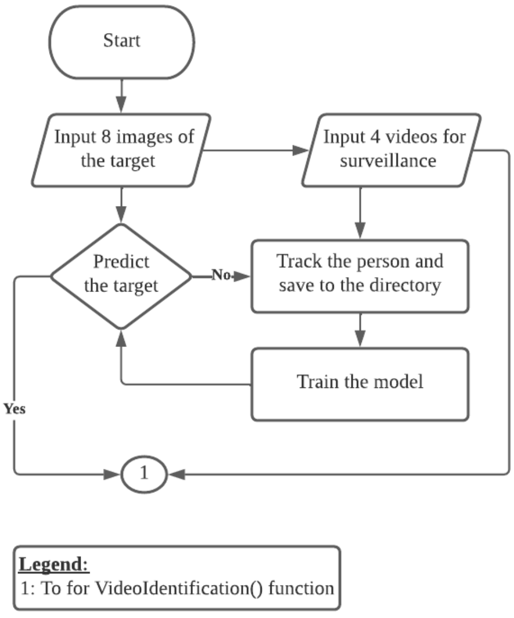

Start
Input 8 images of the target
Input 4 videos for surveillance
Predict the target
No
Track the person and save to the directory
Train the model
Yes
1
Legend:
1: To for VideoIdentification() function

Figure 4-2-2: Flowchart for Initial Person Prediction

To increase the reliability of the prediction, predict all images of the target. The images were loaded from image into array form. After that, predict the images using the model. Through Softmax calculation, the accuracy score was calculated. Target which acquired the accuracy score, which was more than 80, would be predicted as the PID. All the PIDs were appended into 'target_list'. The final PID was finalized by choosing the PID with the highest frequency in 'target_list'.

4.2.4 Person Detection

Person detection was implemented to retrieve the bounding box of the person to be trained for person reidentification in further. For person detection, implement the YOLOv3, as it worked the best among moving contour detection, HOG Descriptor in OpenCV, MobileNetSSD and Haar Cascade, as these methods were quite old and not suitable for this detection situation. Therefore, YOLOv3 was chosen to detect the person.

YOLOv3 is a very effective to detect object with the pretrained coco model. It can detect a total of 80 classes as listed in 'coco.names', such as person, bicycle, car, motorbike, aeroplane and more. In this project, only person class was the concise for detection. Therefore, create a condition to ensure that only person class was detected, instead of other objects.

Before using the YOLOv3, download YOLOv3 weight and convert the YOLOv3 model to create YOLOv3 model, for object detection in TensorFlow format. In the video processing, the name list of 'coco.names' was required for person detection. Therefore, read and load the class list in 'coco.names' into an array "class_names" using YOLOv3 library. After that, load the weights into the YOLOv3 model.

4.2.5 Person Tracking

After person detection setting, continue with the person tracking setting, as they were closely related. In this project, DeepSORT was chosen for the person tracking.

To define the similarity between objects, the maximum cosine distance was set as 0.5. If the maximum cosine distance was more than 0.5, the feature was similar between the object between previous frame and current frame. NMS maximum overlap was set to 0.8, to avoid to many detections on the same object. It can be set it as 1, but it was not advisable, as it would have a lot of similar detections. To load the pretrained model for pedestrian tracking, load the model 'mars-small128.pb'. Then, create a model encoder for tracking of bounding box, with batch size of 1. To classify the identities for each detect, tracking association metric was Nearest Neighbour Metric with the cosine distance. After that, parse the metric on the DeepSORT tracker. The tracker tracked each of the images that was detected in frames, with their bounding box values, ID, class, and more. At the initial of tracks, it predicted and updated the information of the images detected.

Nearest Neighbour was the most common classifier to classify unlabelled data such as video tracking, as it did not require a learning model. In the project, cosine distance was chosen as the distance metrics, instead of Manhattan distance metrics and Euclidean Distance metrics. Cosine distance metrics determined the distance metrics through the cosine similarity, which was the angle between vectors, then subtracted the cosine similarity from one, to obtain the cosine distance. Therefore, it had better performance than other distance metrics.

4.2.6 Video Dataset (Person Tracking)

After setting the person detection and person tracking, proceed on the video. After video uploading, all the videos were resized with width of 450 and height of 300 from the original videos. This controlled the size of video for video showing in the person identification. All the videos and resized videos were stored in the same video directory, as shown in Figure 4-2-3. The video type was either ‘mp4’ or ‘avi’, which was selected by the users. After video resizing, all videos were saved in ‘mp4’ format. Furthermore, the video after person identification were in ‘mp4’ format, same as after video resizing. ‘mp4’ video format is the most common video type, as long as the devices have the video player, the ‘mp4’ videos can be played.

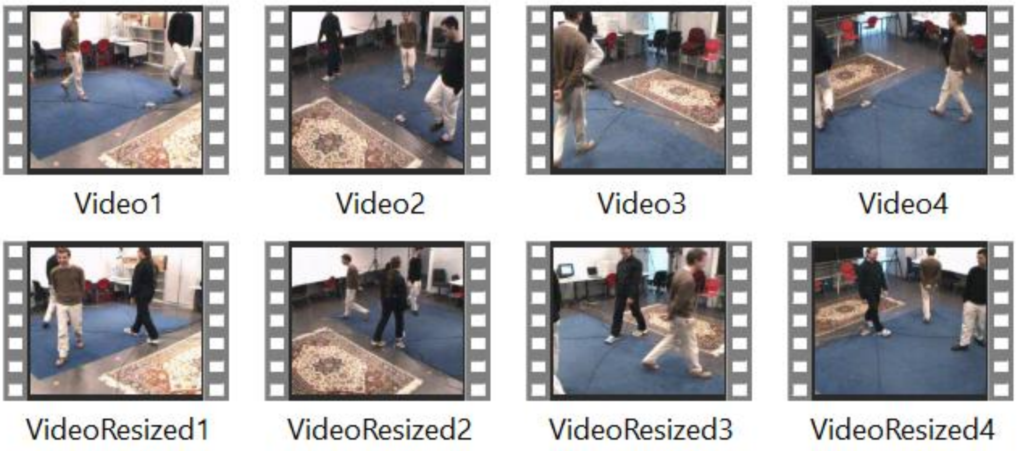


Figure 4-2-3: Videos and Resized Videos

First, load the video by using “cv2.VideoCapture”. To retrieve the frame per second (fps), height (h), and width (w) of the video, create a variable for each feature through the OpenCV library. After person detection and tracking, an ‘mp4’ file was created to save the processed video with its original speed, for backup purpose, the user could not watch these videos.

The video processing started here, by reading the image in videos. If there was no image in the video, there are 2 possibilities, the video was not loaded due to its path was not found, or it was the end of the video. For easy observation, if there was no image anymore, display “Completed”. Due to the colour code of OpenCV, TensorFlow in YOLOv3 differed, convert the colour code BGR in OpenCV into RGB, which suited the TensorFlow in YOLOv3. The dimension of the image now was in 3D, containing height, width, and channel only. To increase a dimension for batch size, expand an extra dimension. Now, our image was in 4D-dimension. After that, resize the image for the YOLOv3 into (416, 416).

After image processing, predict the image using YOLOv3.Information such as bounding box information, score information, class information, and a total number of the detected were retrieved. The boxes contained x-coordinate, y-coordinate, width, and height. The score contained the confidence score of the detected. The class contained the object class as listed in ‘coco. names’.

To detect objects in the video, create an array list ‘detections’ and call the detection function by parsing the bounding boxes, confidence scores, class names and features. Then, convert all of them into a NumPy array for non-maximum suppression. These NumPy arrays in ‘detections’ were used for non-maximum suppression to control and eliminate multiple detection frames on a target in the detection frame. In brief, it removed redundant images. To draw the rectangle frame for those people detected, create a colour list using the Matplotlib library. It allowed the assigning of colour for each frame using the tracker ID.

To track objects in the video, predict and update the DeepSORT tracker with the detections. Every new detection was recorded in the tracker. To loop the results in the tracker, create a FOR loop. In FOR loop, if the tracker had no update, the current track would be skipped. After that, retrieve all the features such as bounding box, class and frame colour for the respective track. For retrieving bounding box values, use ‘to_tlbr’. ‘to_tlbr’ format in the DeepSORT tracker allows the values in the x-coordinate and y-coordinate in bounding boxes to be allocated in OpenCV for video display.

Until the current state, all the objects in videos are detected. However, ‘person’ was the only class that was needed. Therefore, create an if-else condition to control the class for a person. In addition, create a list to store every person track ID and newly allocated

ID according to its detected sequence. The reason to create this list was that the original track ID list consists of other objects detected.

After that, create the variable for each coordinate of the bounding box, such as xA, xB, yA, and yB. xA is the x-coordinate of the left upper bounding box, xB is the sum of the width and x-coordinate. On the other hand, yA is the y-coordinate of the left upper bounding box, yB is the sum of the height and y-coordinate.

Subsequently, retrieve and save the images extracted from the video into the directory set. The directory was created as shown in the Figure 4-2-4 . The training folder consisted of person images in training and validation. In the input query set, the images of the targeted person to be identified in person identification were allocated here. As shown in Figure 4-2-4, each image name was set as '{directory}{PID}_s{frame_no}c{camera_no} _{num}.png'. The directory was the path of the image file for saving. PID as the ID of the person detected. Frame_no was the sequence of the frame in the video. Camera_no was the identifier of the camera, as there were numerous videos. Num is the number of images. PNG is the image format. For instance, '0000_s112c2_2.png' showed that this image had a PID of '0000' in the 112th frame of Camera 2, it is the second image that had been saved into the directory. However, before reaching the minimum requirement of 50 images, the images will be stored in temporary training (tmpTraining) directory, to prevent any corruption in the main training directory. Once the requirement achieved, the images in temporary training (tmpTraining) directory were moved to training directory for model training.

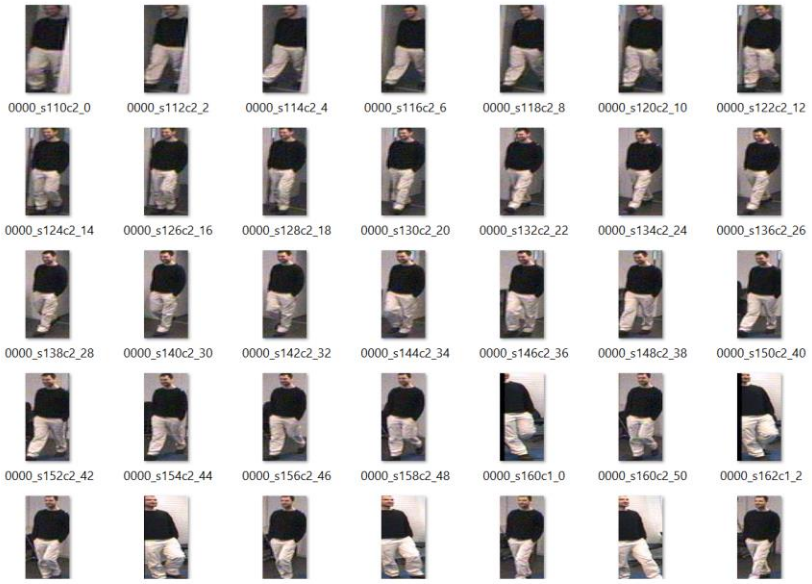


Figure 4-2-4: Images Extracted from Different Cameras in Training Set

After saving the images into specific directories, display the video with colour frames using OpenCV. Each person had a different PID. Hence, each of them had a different colour frame in the video, as set previously. Then, insert each coordinate of the bounding box, colour set and the border size of the frame. Furthermore, create a smaller rectangle frame above the person bounding box with his/her PID. For ease of observation, the current FPS of the video was shown on the top left of the video. For better visualisation, resize the window for video display and show the output video. When the video had fully processed, it was saved to the 'VideoTracking.mp4'. To exit or stop video processing, press key 'q'. Figure 4-2-5 and Figure 4-2-6 showed the image capture from different videos. Each of the persons detected is bounded in a different colour frame.

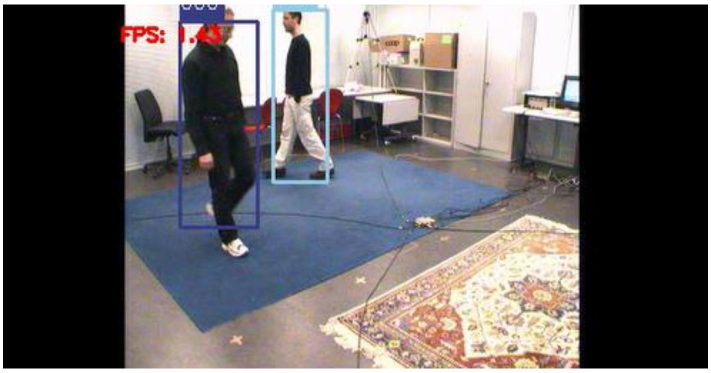


Figure 4-2-5: Person Tracking in Video 1

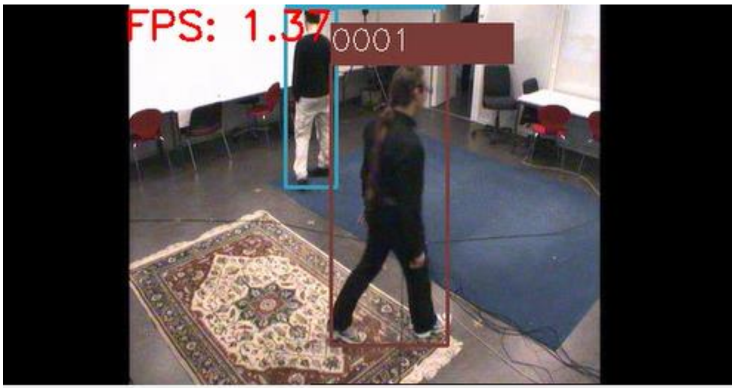
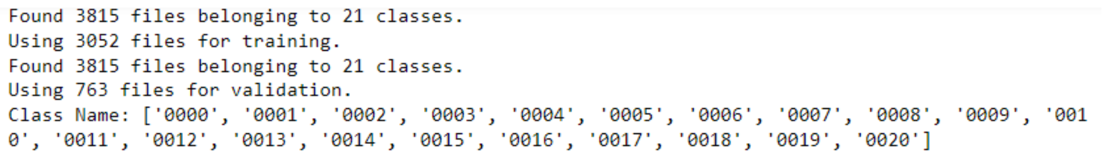


Figure 4-2-6: Person Tracking in Video 2

4.2.7 Model Training

Model training was the most important part of this application. Model training was divided into two in this application, such as pretrained model and retrained model. The pretrained model predicted the images of the target when the images were inserted by the users, while retrained happened in two conditions. In the first condition, when the target was predicted as None, which was unable to predict by model. The system would proceed to the video tracking, retrieve and save the images into the Training directory. After that, the model was retrained for model prediction again. In the second condition, some of the people might be missed out. Therefore, during person identification, the sets of images of the person would be cropped from the video and saved into the directories, and the model was retrained.

Before model training, the training dataset was split into the training dataset and validation dataset, with the ratio of 8:2. The seed allocated on the splitting was 123. The height of the images was 128, and the width of the images was 64. The batch size was set as 32. Class names list was derived from PID folders in the Training directory. In Figure 4-2-7 , there were 3815 images in the original Training directory. After data splitting, the training dataset had 3052 images, while the testing dataset had 763 images. There was a total of 21 classes in the training dataset and validation dataset.



```
Found 3815 files belonging to 21 classes.
Using 3052 files for training.
Found 3815 files belonging to 21 classes.
Using 763 files for validation.
Class Name: ['0000', '0001', '0002', '0003', '0004', '0005', '0006', '0007', '0008', '0009', '001
0', '0011', '0012', '0013', '0014', '0015', '0016', '0017', '0018', '0019', '0020']
```

Figure 4-2-7: Summary of Training Set, Validation Set and Class Names

The data augmentation of the model produced more positive datasets for model training, without adding new images into the datasets. In the model, data augmentation techniques such as horizontal random flip, random rotation, and random zoom were implemented. In the Figure 4-2-8, the image was flipped horizontally, rotated, zoomed, and cropped. Then, a new image was formed.

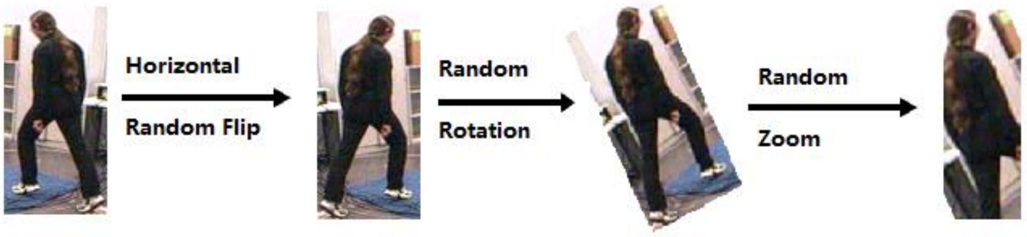


Figure 4-2-8: Data Augmentation of Model

The model consisted of 6 convolutional layers. In the first layer, the model had 16 filters. In the second layer, the model had 32 filters. In the third layer, the model had 64 filters. In the fourth layer, the model had 128 filters. In the fifth layer, the model had 256 filters. In the sixth layer, the model had 512 filters. Each layer had the same padding value, ReLU activation, and a 2D max pooling. After that, dropping out 20% of the training set. After dropping, the inputs were flattened for fully connected layers with 1024 dimensions. To predict the inputs using the model, the dimension of the last fully connected layers was the number of classes. The number of classes for the model was 21.

It was a model compilation stage. The model was complied with Adam optimizer, and the loss was counted by Sparse Categorical Cross Entropy. Before model fitting, the number of epochs was controlled by the number of classes that existed, to ensure that no overfitting and underfitting issues occurred. If the number of classes was more than 30, epochs were set as 20. Else, if the number of classes was more than 20 but less than 30, epochs were set as 15. Else, the epoch was set as 10. For instance, the number of classes was 21, the epochs were set as 15.

After setting the epochs, the model was fitted with a training dataset, validation dataset, and epochs. After training the data, save the model to the directories. For a better understanding of the model, the visualization such as graph was shown as in Figure 4-2-9. The figure showed the graph of training and validation accuracy and the graph of training and validation loss. The jittering formed was due to the dropout in the model. No underfitting or overfitting issue happened.

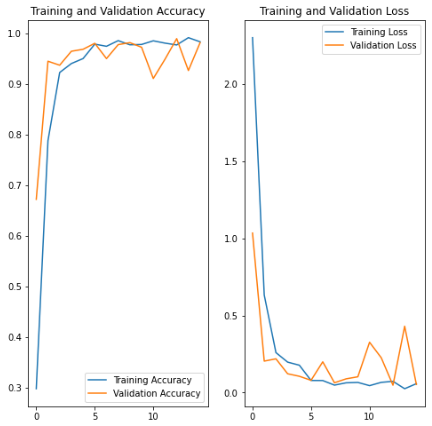


Figure 4-2-9: Visualisation of the model

4.2.8 Person Identification

Person identification was the main part of this system. It predicted every person in the videos and compared it with the PID-targeted person. If PID of person in the frame was the same as the PID of target, the targeted person in the videos would be bounded with a green bounding box. Oppositely, the others would not be bounded.

The model prediction was the most important part of the person identification application. Therefore, the model prediction must be accurate and fast. For the old way, the cropped person in the bounding box was predicted one by one. For example, a frame had 4 people, with a total of 100 frames. It meant that the model prediction was 4*100 = 400. Therefore, to increase the efficiency of the model prediction for person identification, the prediction list method was implemented. The person tracked by the tracker had its own 'id', and the system assigned a 'PID' for each tracked 'id'. The application predicted the 'PID' using this 'id'. For example, if 4 persons were in the frames, each of them was assigned with an 'id', by the DeepSORT tracker. The application predicted the PID 60 times (4 ids * 15 times) only, excluding the person who walked out of the frame and back to the frame. This algorithm reduced the burden on the application.

4.2.8.1 Initial Person Identification

The initial state of person identification was similar to the video tracking. If the video was opened, the person identification proceeded. If not, the person identification process was ended. When the video was opened, a 'frame' counter was initialized. Due to the slow video processing by the DeepSORT, the frame proceeded four by four, controlled by frame%4==0. This speeded up the person identification process slightly, as only even frames were processed in person identification. Once the condition was achieved, the 'ret' flag was set as True for further person identification process. For user view, the loading status was shown sixteen by sixteen, instead of four by four, to prevent bad user visualization.

If the ret was set as True, the image was converted from BGR into RGB, as cv2 runs with RGB only. Then, predict the images using the yolov3 for 'boxes', 'scores', 'classes', and 'nums'. All of these retrieved variables were processed to be compatible for tracking by DeepSORT tracker. After the detection, update the trackers with the detection and retrieve the information in the tracker in FOR loop. The information such as 'id', 'bbox', and 'class_name' could be retrieved from the tracker. 'id' was the id assigned by the trackers for each object detected. 'bbox' was the four points of the bounding box. ‘class_name’ was the label for an object such person, bicycle, pot and more. The following steps could refer to the flowchart for person identification, as shown in Figure 4-2-10.

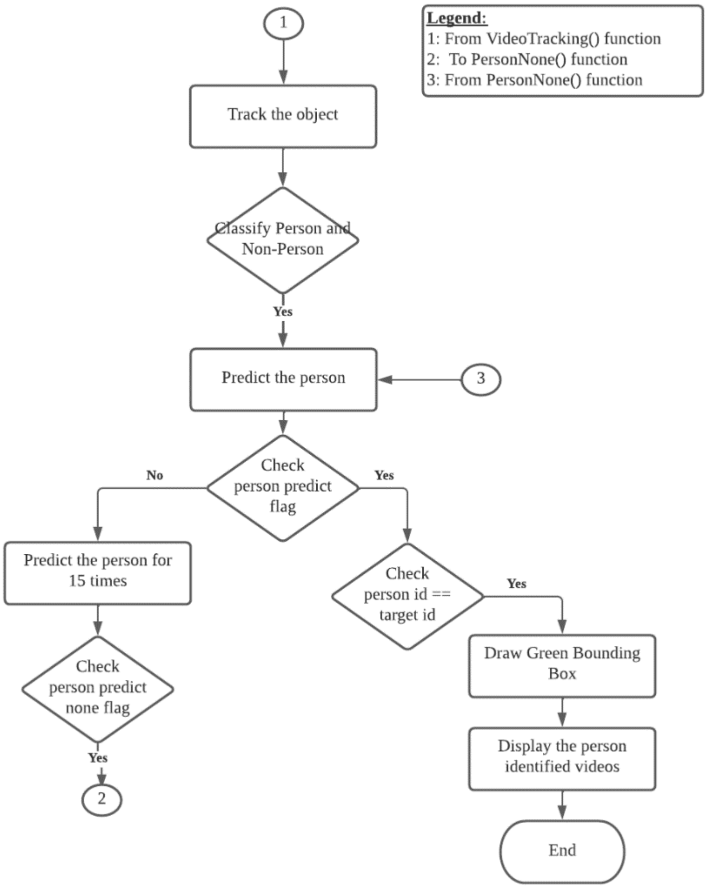

Legend:
1: From VideoTracking() function
2: To PersonNone() function
3: From PersonNone() function
1
Track the object
Classify Person and Non-Person
Yes
Predict the person
3
Check person predict flag
No
Yes
Predict the person for 15 times
Check person predict none flag
Yes
2
Check person id == target id
Yes
Draw Green Bounding Box
Display the person identified videos
End

Figure 4-2-10: Flowchart for Person Identification

Once the ‘class_name’ was person, allocate each point of the bounding box with four variables, such as xA, xB, yA, and yB, as shown in Figure 4-2-11. After allocating, calculate the area of the bounding box, it was useful later. By taking the advantage of the points of the bounding box, crop the image. If the size of the cropped image was not empty, proceed for person prediction.

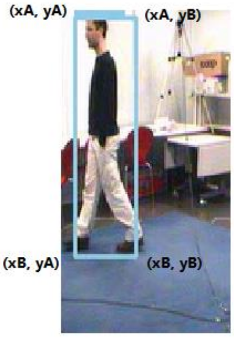
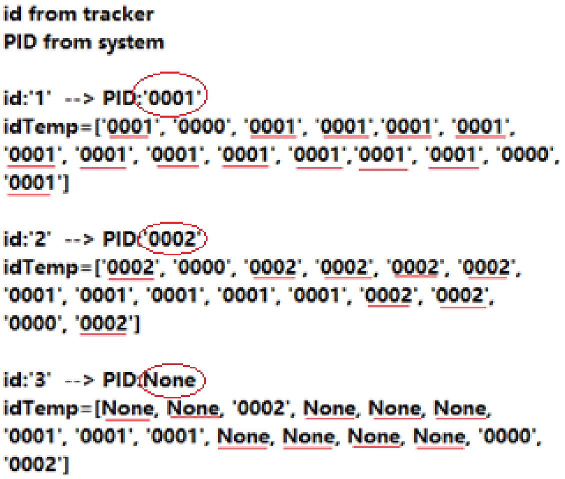


Figure 4-2-11: Coordinate of Bounding Box

In this application, it has two different IDs, ‘id’ was assigned by the tracker, while ‘PID’ was assigned by this system. The system assigned the ‘PID’ according to ‘id’. For example, the tracker assigned ‘1’ for person 1 and assigned ‘10’ for person 2. Those unassigned ‘id’ was non-person, they were useless in this appliction. ‘id’ with ‘1’ might appear in many frames. Therefore, to reduce the burden on the system, the system identified ‘PID’ according to ‘id’. It meant that once the tracker predicted ‘id’ as ‘1’, its ‘PID’ was predicted as ‘0001’ by the system. However, it was very critical, once the cropped image of an ‘id’ is predicted wrongly by the model, it would predict all wrong ‘PID’ for certain ‘id’. To reduce the criticality, the same ‘id’ from a different frame was predicted 15 times. Every time, the ‘id’ was predicted for ‘PID’, the PID was appended into the ‘idTemp’ list. The ‘PID’ will be finalized by selecting the most frequently appeared ‘PID’ in the ‘idTemp’ list, using the most_frequent(idTemp) function. The concepts and logic were shown in Figure 4-2-12. most_frequent(idTemp) function returned PID with the highest frequency.



```
id from tracker
PID from system

id:'1'  --> PID:'0001'
idTemp=['0001', '0000', '0001', '0001','0001', '0001',
'0001', '0001', '0001', '0001', '0001','0001', '0001', '0000',
'0001']

id:'2'  --> PID:'0002'
idTemp=['0002', '0000', '0002', '0002', '0002', '0002',
'0001', '0001', '0001', '0001', '0001', '0002', '0002',
'0000', '0002']

id:'3'  --> PID:None
idTemp=[None, None, '0002', None, None, None,
'0001', '0001', '0001', None, None, None, None, '0000',
'0002']
```

Figure 4-2-12: Relationship between ‘id’ and ‘PID’

Before that, create the id_list to store the ‘id’ assigned by DeepSORT tracker, PID, ‘icheck’, and ‘idTemp’. PID is the ID that is predicted from the trained model. ‘icheck’ is the frequency of appending the ‘PID’ into ‘idTemp’, similar to the length of the ‘idTemp’ list. If the PID was predicted as None, proceed to predictNone() function, as shown in Figure 4-2-10. Oppositely, If the PID was not predicted as None, compare PID with target_predicted in the initial person prediction stage. If PID and target_predicted were the same, draw a green bounding box for that cropped image. However, before drawing a green bounding box for the cropped image, check if the area of cropped image is more than 2000, if it was True, draw the green bounding box in that frame of the video. After that, the complete person identification videos were stored in the same directory, as shown in Figure 4-2-13.

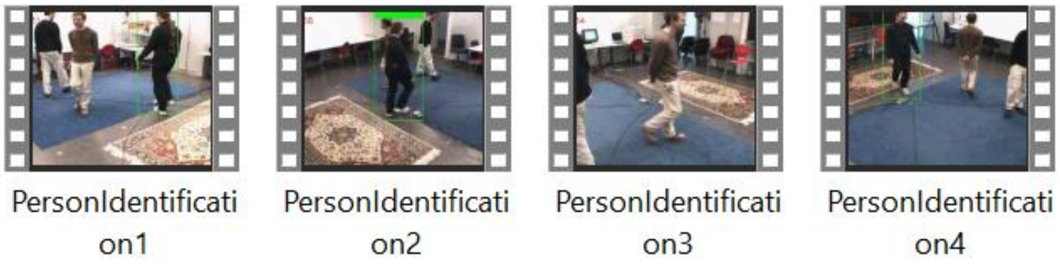


Figure 4-2-13: Person Identification Videos

4.2.8.2 Predict None

PredictNone() function was to create a directory and save the cropped image into the newly created PID directory. Furthermore, PredictNone() function allowed model retraining. When the requirements for the retraining model were achieved, the model would be retrained. The procedure for PredictNone() was shown as in Figure 4-2-14.

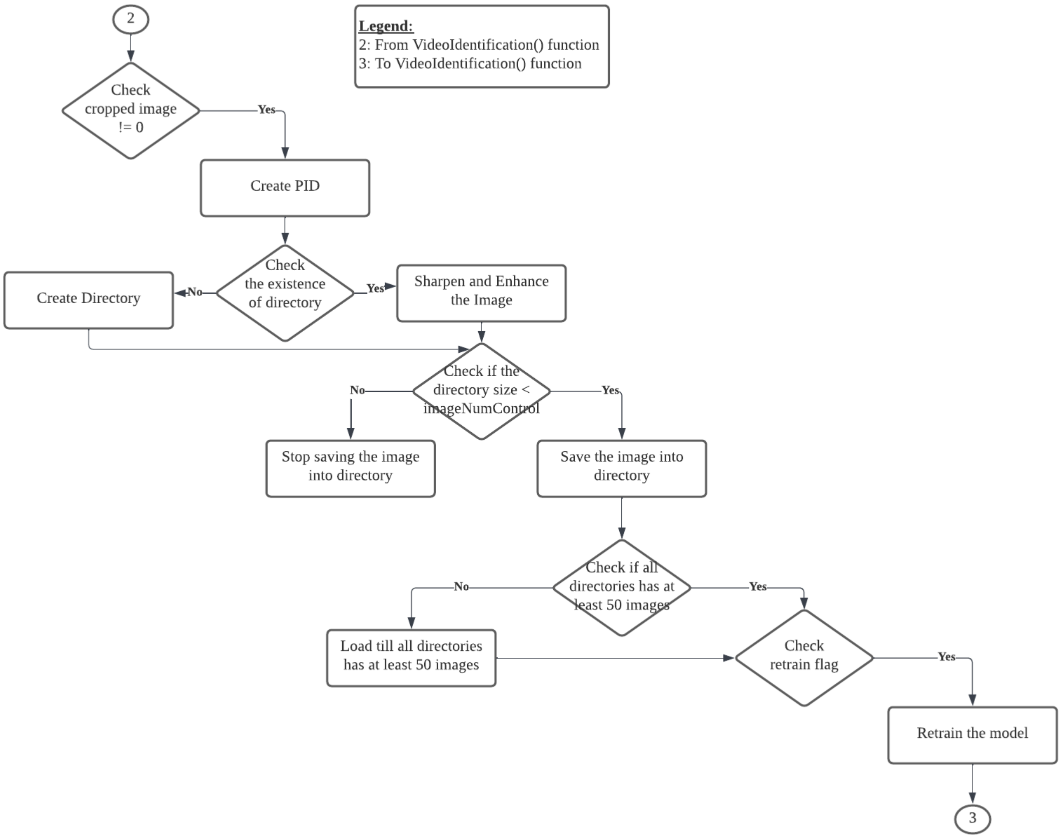

Legend:
2: From VideoIdentification() function
3: To VideoIdentification() function
2
Check cropped image != 0
Yes
Create PID
Check the existence of directory
No
Create Directory
Yes
Sharpen and Enhance the Image
Check if the directory size < imageNumControl
No
Stop saving the image into directory
Yes
Save the image into directory
Check if all directories has at least 50 images
No
Load till all directories has at least 50 images
Yes
Check retrain flag
Yes
Retrain the model
3

Figure 4-2-14: Flowchart of Predict None

‘nonePredictList’ was passed from the initial person identification stage. It consisted of ‘id’, ‘PID’, ‘image_num’, ‘trainedFlag’, ‘iCheck’, and ‘pathFlag’. Similar as the initial person identification stage, ‘id’ was assigned by the tracker, while ‘PID’ was assigned by this system. ‘image_num’ was the number assigned for image file naming. ‘trainedFlag’ was the flag to allow model retraining. If ‘trainedFlag’ is True, model retrained, while ‘trainedFlag’ is False, model was not retrained. ‘icheck’ is the frequency of appending the ‘PID’ into ‘idTemp’. ‘pathFlag’ was the flag to check the existence of the directory.

First, ‘predictID’ was passed from the initial person identification stage, checking if it was none. If it was none, create a PID for it, using the current length of the class name. Class name was the label of the PID directory in the Training directory, as shown in Figure 4-2-15. After that, check if the ‘nonePredictList’ was blank, if it was blank, append all the five variables into it.

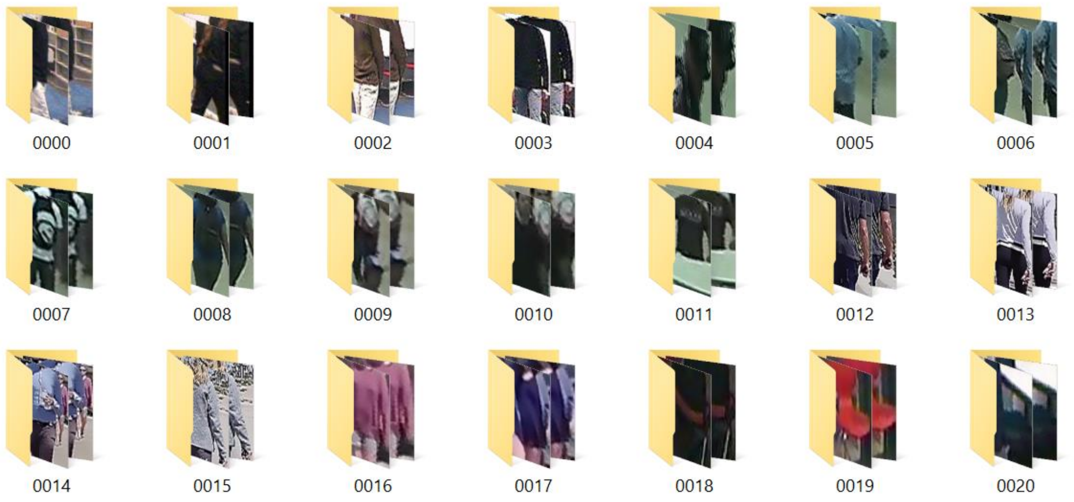
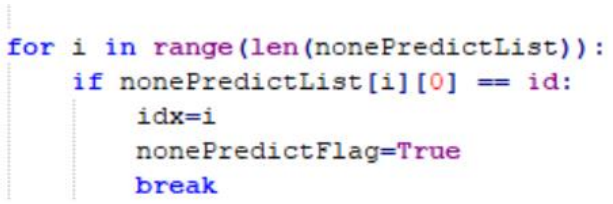


Figure 4-2-15: PID directories in Training Directory

To allow image file saving according to the 'id' assigned by the tracker, use the for loop to find out the 'id' in 'nonePredictList', and assigned 'idx = i', as shown in Figure 4-2-16. After that, set 'nonePredictFlag' as True. If 'nonePredictFlag' is False, append all the five variables into it. It meant that the 'id' is newly assigned by the tracker.



```
for i in range(len(nonePredictList)):
    if nonePredictList[i][0] == id:
        idx=i
        nonePredictFlag=True
        break
```

Figure 4-2-16: For Loop in PredictNone()

'nonePredictList[idx]' was an array list, 'idx' was compulsory to retrieve the particular array element from 'nonePredictList'. After checking the 'iCheck' in the 'nonePredictList[idx]' is larger than 15, check if the existence cropped image and an area of 2000, which was passed from the initial person identification stage. If both requirements were achieved and the pathFlag in 'nonePredictList[idx]' is True, create the directory for this PID, once the PID directory was created, the 'pathFlag' was changed into False. Initially, 'pathFlag' in all 'nonePredictList[idx]' is True, when newly appending into the ‘nonePredictList’. When the ‘pathFlag’ was false, join the temporary directory and PID as the path, the path exists in the previous.

After that, it was image enhancement. The enhancement stage was divided into three parts. First, the cropped image was sharpened, with the NumPy array implementation of cv2.filter2D. The array was [[0, -1, 0], [-1, 5, -1], [0, -1, 0]] for image sharpening. The reason to sharpen the image was that the person in the video might be blurred, as the resolution of the CCTV camera was not high as the digital camera. Second, the sharpened image was blurred with bilateral filter, to focus on the image and blur the background. Third, the last enhancement with the detail enhancer, to sharpen the detail, especially the person cropped. After the enhancement of the cropped image, saved it into the temporary training directory.

4.2.8.3 Model Retraining

Model retraining allowed the Training directory to be trained and built a new model once the model retraining requirement was achieved. Model retraining was due to the newly created PID directory, which was missed out on in the previous model training.

To maintain the quality of the model, for the first requirement, the minimum number of images in all PID directories was at least 50 before model training. The maximum images that could be saved were dependent on the duration of the videos. When the duration of the video did not exceed the 30s, the maximum number of saved images was 100. On the other hand, when the duration of the video exceeded the 30s, the maximum number of saved images was 300.

The second requirement was that ‘trainedFlag’ must be True. Once both requirements were achieved, the model would be retrained. After model retraining, return ‘nonePredictList’ to the initial person identification stage.

4.2.9 Remove Directory

After person identification for each video, those directories which had lesser than 50 images would be removed, to prevent any false prediction when the model was retrained during the person identification progress.

First, join the temporary training (tmpTraining) directory and PID to check whether the existence of the path. If the path existed, set ‘imageNum’ as the number of directories in the temporary training (tmpTraining) directory. When the ‘imageNum’ was smaller than 50, showing that the directory had less than 50 images.

However, before reaching the minimum requirement of 50 images, the images were being stored in the temporary training (tmpTraining) directory, to prevent any corruption in the main training directory. The requirement to move the images in the temporary training (tmpTraining) directory into the training directory is that the images in the temporary training (tmpTraining) directory must contain at least 50 images. Once the requirement was achieved, the images in the temporary training (tmpTraining) directory were moved to the training directory for model training, with newly created PID, according to the number of classes in Training directory. Before finishing the person identification progress, all images and folders in the temporary training (tmpTraining) directory were cleared, before the next person identification. To prevent some of the original pretrained datasets were deleted accidentally, the temporary training (tmpTraining) directory played an important role.

4.2.10 User Interface Design

The UI design was designed under PyWebIO library. PyWebIO provides an interactive platform allowing users to input and receive output on the browser. It is simple for the users to interact with the system [34]. It allows the developer to retrieve the input from the users and display the output results for the users.

The inputs of this application were the images of the targeted person and camera videos. Therefore, users are required to insert eight images of the target and provided multiple concurrent CCTV videos as the input for the person identification application. Also, the user interface should be simple and easy to interact with the users. Furthermore, the results of person identification will be shown in the output. The person in the green box is the targeted person, while the pedestrian will be ignored. Also, the images of the targeted person in multiple videos will be shown as the output for users.

To receive input from the users, apply ‘file_upload’ in a FOR loop and save the file uploaded into the QueryInput directory. It was as shown in Figure 4-2-17. To simplify the images retrieval, the file was named according to the sequence in the loop. At the same time, the image was appended into 'imageList'. For example, if the image is first uploaded, the naming was ‘1.png’. The FOR loop was set for eight images. Once the FOR loop ended, the images were displayed in a row from the 'imageList', as shown in Figure 4-2-18. This showed that all the images were successfully saved in the directories.

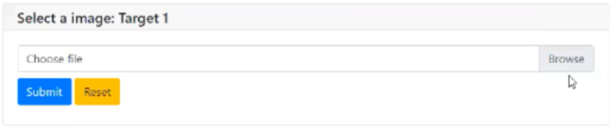

Select a image: Target 1

Figure 4-2-17: Image Uploading UI

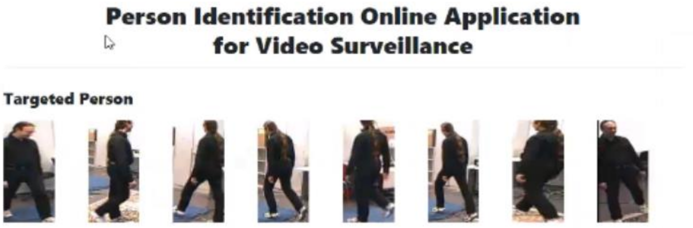


Figure 4-2-18: Images Uploaded Display

After image uploading, it was video uploading. The video uploading UI was as shown in Figure 4-2-19. Due to the file size of videos were slightly larger, it might take time to upload and save into the Video directory. When the videos were uploaded one by one, a loading bar was shown, to allow the users to trace the progress of video uploading, as shown in Figure 4-2-20. Once the video was uploaded successfully, it showed ‘Video.avi is uploaded’, as shown in Figure 4-2-21.

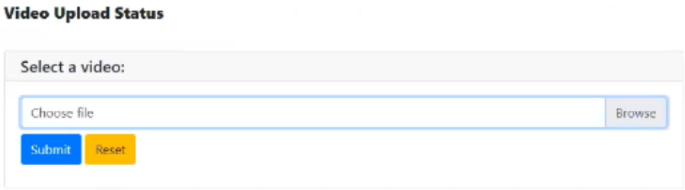


Figure 4-2-19: Video Uploading UI

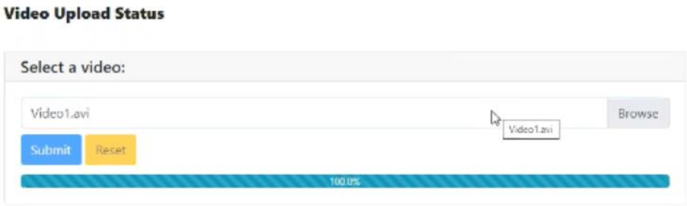
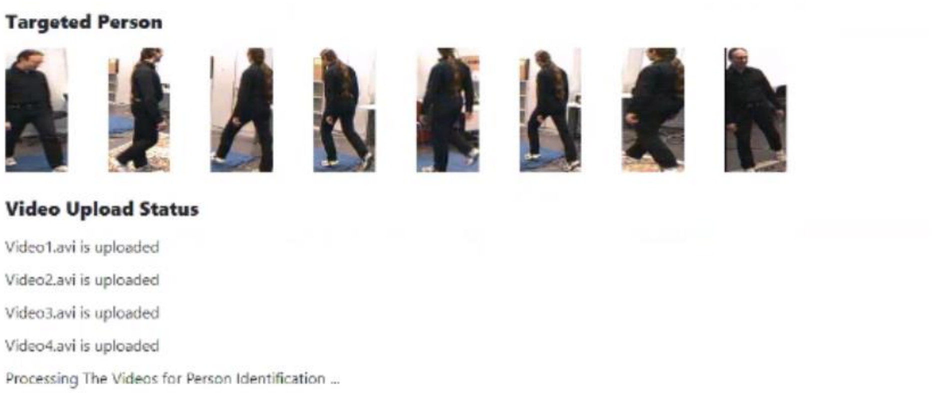


Figure 4-2-20: Video Uploading with Loading Bar



Targeted Person

Video Upload Status

Video1.avi is uploaded

Video2.avi is uploaded

Video3.avi is uploaded

Video4.avi is uploaded

Processing The Videos for Person Identification ...

Figure 4-2-21: Video Uploaded Successfully

After all images and videos were uploaded successfully into directories, the person identification process was started. If the images of targets were not predicted successfully, it would start the person tracking, to save all the person cropped images into the Temporary Training (tmpTraining) directory from Video 1 only. After that, train the model and predict the images of targets again. It should be predicted successfully, as the targeted person was saved into the Temporary Training (tmpTraining) directory and retrained the model. Next, it was the person identification process. Once four of the videos were done with the person identification process. The ‘doneFlag’ would be changed from False into True. Thus, the output, which is the person identification videos would be displayed as shown in Figure 4-2-22.

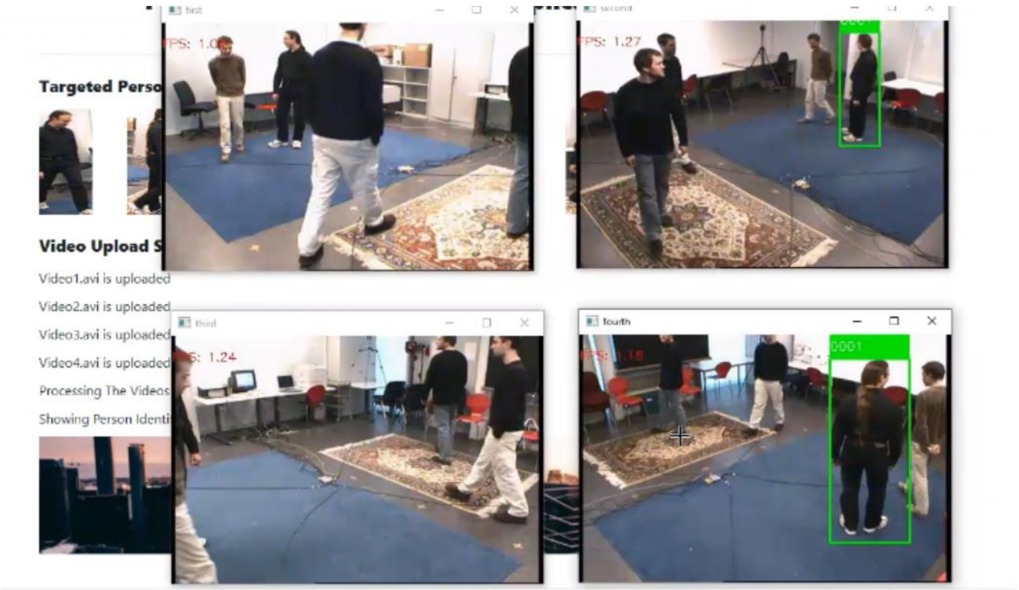


Figure 4-2-22: Display Person Identification Videos

To observe and watch the person identification again and again videos, the user clicked on the Show Person Identification Videos, to watch the 4-person identification concurrently repeatedly as shown in Figure 4-2-23. Furthermore, the user downloaded the person identification videos for better experience of observance and watching, as shown in Figure 4-2-23. As watching in the user's video player, they had better control on the videos, such as play, pause, fast forward and skip.

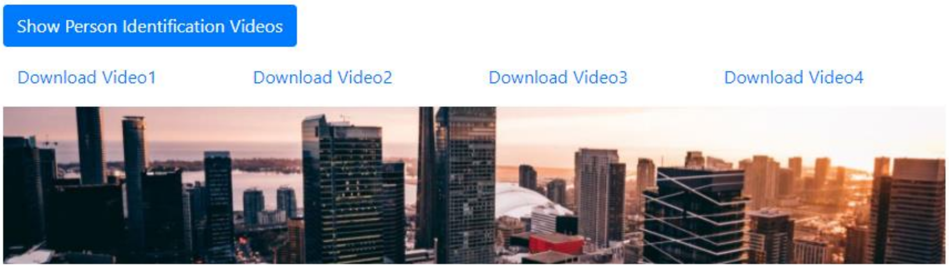


Figure 4-2-23: Download Person Identification Videos

CHAPTER 5
SYSTEM IMPLEMENTATION

5.1 Methodologies and General Work Procedures

In this development of person reidentification application, there were six main steps in total. Initially, users were requested to insert four videos. Simultaneously, the users inserted the images of the target, eight images from various distinct angles. After that, it was backend processing. Before any person tracking and person identification, the target images that were inserted by the users, would be predicted with the trained model. If the model predicted the target, then it proceeded to the person identification stage. Oppositely, the target images are not predicted, it would be proceeded to person tracking, for images extraction into the directories. After image extraction, the model would be trained, and the images of the targeted should be predicted, by implementation of the newly trained model. After that, it proceeded to the person identification stage. Last, display videos with the target detected in videos. The target was bounded with green frames, which meant that it was successfully identified, while the pedestrians were ignored. Figure 5-1-1 showed the flowchart of system overview.

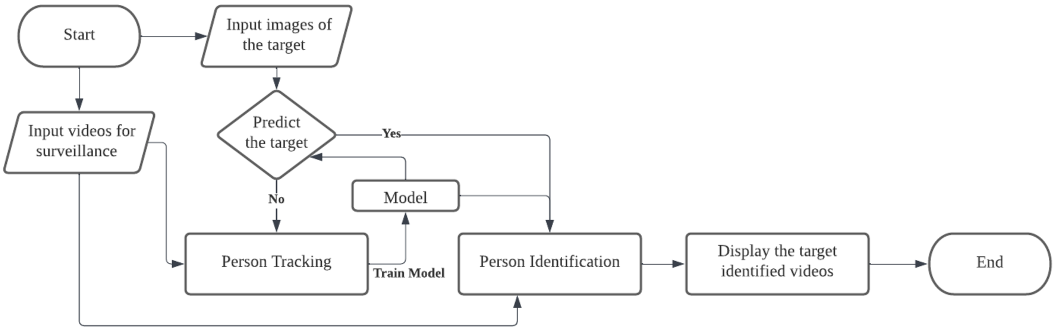

Start
Input images of the target
Input videos for surveillance
Predict the target
Yes
No
Model
Person Tracking
Train Model
Person Identification
Display the target identified videos
End

Figure 5-1-1: Flowchart of System Overview

5.2 Tools to Use

5.2.1 Hardware Setup

Without achieving the hardware requirement, it may not function smoothly when the installation of this person identification application for video surveillance. Therefore, it was important to reach the minimum hardware requirement. The most important in this person identification application was to require at least 6GB of GPU memory, as CNN model training required more memory for training the datasets. Therefore, an additional GPU requested was allocated for model training in MobaXterm, but later the software platform was changed into X2GoClient. The hardware requirements for the person identification application for video surveillance are stated in Table 5-2-1.

Table 5-2-1: Hardware Requirements

GPU	Intel (R) UHD Graphics 620
GPU Memory	At least 6 GB
CPU	Intel (R) Core (TM) i5-8250 CPU @ 1.60GHz
Processor	x64-based Processor
Operating System	Microsoft Windows 10
RAM	12 GB
System Type	64-bit Operating System

5.2.2 Software Setup

The python environment in Jupyter Notebook was be chosen for the development of person identification application for video surveillance, as python is a programming language which is powerful for image processing and deep learning [33]. Furthermore, python consists of many useful and implementable libraries for this application, such as DeepSORT, TensorFlow, cv2, NumPy, PyWebIO, yolov3 and more. Therefore, it is the most suitable language to implement the person identification application for video surveillance. Both backend and front-end development environment were in python. However, the GPU memory was not enough for the person reidentification training. Therefore, instead of Jupyter notebook, MobaXterm and X2Go Client were installed and set up for model training to prevent memory exhaustion during model training, for the Torchreid library.

In the final stage, the model that implemented for training did not require that huge amount of memory. Therefore, the NVIDIA GPU in the laptop was enough. However, before using the GPU, set up the CUDA toolkit and cuDNN. Before installing the cuDNN, check the version of CUDA and python, whether they were compatible with cuDNN. It was important to have cuDNN, to allow loading of dynamic library 'cudart64_110.dll'. After setting up the GPU, install the required python libraries in the environment. In Table 5-2-2, it displayed the Python libraries that are installed for this application.

Table 5-2-2: Libraries in Python

Libraries in Python	**Function**
cv2	Provides real time computer vision function [35]
DeepSORT	Track the object detected [36]
io	Deal with various type of inputs and outputs. [37]
matplotlib	Create static or dynamic interactive visualisations [38]
NumPy	Fundamental package in Python which provides multidimensional array object and more. [39]
os	Interact with operating system [40]
pathlib	Deal with the directories and files [41]
PIL	Interpret and edit images [42]
PyWebIO	Provide interactive platform to be launch on the browser. [34]

shutil	Copy and remove files in the directories. [43]
socket	Transliterate the Unix system call [44]
TensorFlow	Build and train the model. [45]
Yolov3	Detect object using CNN model [21]

In this project, the person identification learnt from input datasets for PID classification. The input was the images of the targeted person and the surveillance videos, while the output was the result of the person identification with the target identified in the green frames. In this person identification model, the CNN model performance was evaluated based on accuracy in percentage. In the end, the target was detected with green person recognition frames, while other pedestrians were not bounded with any frame. When the green person recognitions are shown, it means that the targeted person had been detected and identified successfully.

5.3 User Requirement

1. User shall input four videos from multiple angles for person identification.
2. User shall input eight images of target from different angles for person identification.
3. User shall watch the person identification videos after person identification process.
4. User shall replay the concurrent videos, by clicking ‘Show Person Identification Videos’.
5. User shall download the person identification videos, for better observation.
6. User shall identify the target in the green frames in the videos.

5.4 System Performance

Person detection which applied the YOLOv3 detected 80 classes of object accurately using the confidence during detection. As in this project, only person was required for identification. Therefore, set the class as person to be detected only, and the person will be detected in the videos accurately, without mixing with the other 79 classes. In person tracking, each identified person was be provided with an identifier to track the person correctly. In the DeepSORT, Nearest Neighbour filter with cosine distance metric algorithm solved the detection and association issues to reduce and avoid noises when tracking the person in the videos. After person detection and person tracking, to increase the performance of person identification system, the resolution of the images within the bounding box was enhanced, to reduce the blurriness and increases the sharpness of the image.

The previous person identification method was done using TorchReid library for person identification. However, the library was suitable for image dataset person identification only, it was not suitable for video person identification implementation. Furthermore, it required a huge amount of GPU for person identification of images only, it was not efficient for a video application. Therefore, the CNN model was implemented and built with 6 convolutional layers with 0.2 dropout. The model implements the Sparse Categorical Cross Entropy in classification layer and Adam optimizer.

5.5 Verification Plan

Although the accuracy of this person identification had high accuracy and consistency. However, different situation may result in false identification. In this application, there were three main process such as person detection, person tracking and person identification. To retrieve training and testing data for the person identification, the detected images of person with its own identifier were saved into training set and test set separately. However, there were some issues during when performing these three steps. There were few critical situations are listed as below:

1. Person occlusion. During person detection and tracking, the person might be blocked by some other object, causing an occlusion.
2. Interchange of person identifier. During person detection, if the distance between objects was too short, their identifier would be interchange automatically.
3. Person image retrieval with low resolution. During image retrieval into directories for training, the images extracted directly from the videos have low resolution.
4. Person Identification Camera Issues. During the person reidentification, the CNN model worked when there are images under more than 1 cameras only.

1. Person occlusion

Table 5-5-1: Validation of Person Occlusion

Procedure Number	P1
Method	Testing
Applicable Requirements	Retrieve bounding box of person even there is an occlusion
Purpose / Scope	To retrieve the bounding box of person
Items Under Test	Bounding box of person in the video
Precautions	The occlusion will reduce the person threshold during person detection.

Special Conditions / Limitations	Bounding box with low threshold may cause the false negative of person detection, it was not identified for PID correctly.
Equipment / Facilities	Laptop
Data Recording	None
Acceptance Criteria	System successfully retrieves the bounding box of person from the video.
Procedures	1. Insert video 2. Detect the object in the video and choose the person from the detected object, using YOLOv3 and DeepSORT tracker. 3. Assign a PID for each person identified, according to ID assigned by the DeepSORT tracker.
Troubleshooting	Repeat the procedure
Post-test Activities	None

2. Interchange of Person Identifier

Table 5-5-2: Validation of Interchange of Person Identifier

Procedure Number	P2

Method	Testing
Applicable Requirements	Identify each person with an identifier.
Purpose / Scope	To provide an identity for each person tracked
Items Under Test	Bounding box of person in the video
Precautions	The distance between persons should be greater.
Special Conditions / Limitations	When the persons are too close to each other, their identifier will interchange.
Equipment / Facilities	Laptop
Data Recording	None
Acceptance Criteria	System successfully identifies each person with an identifier.
Procedures	1. Insert video. 2. Detect the person and track the person using the identifier. 3. Assign a PID for each person identified, according to ID assigned by the DeepSORT tracker.
Troubleshooting	Repeat the procedure

Post-test Activities	None

3. Person Image Retrieval with Low Resolution

Table 5-5-3: Validation of Person Image Retrieval with Low Resolution

Procedure Number	P3
Method	Testing
Applicable Requirements	Increase the resolution of the images
Purpose / Scope	To retrieve the images in the bounding box with a better resolution
Items Under Test	Image within the bounding box in the surveillance video
Precautions	Image should have better resolution for better person identification in future
Special Conditions / Limitations	Image with low resolution may decrease system accuracy.
Equipment / Facilities	Laptop
Data Recording	None

Acceptance Criteria	System successfully increases the image resolution in the bounding box.
Procedures	1. Insert video 2. Detect the person and track the person. 3. Increase the resolution of the image retrieved from the bounding box with image sharpening, Bilateral filter, and image enhancement.
Troubleshooting	Repeat the procedure
Post-test Activities	None

4. Person Identification Camera Issues

Table 5-5-4: Validation of Person Identification Camera Issues

Procedure Number	P4
Method	Testing
Applicable Requirements	Identify person from multiple cameras
Purpose / Scope	To identify the target from the image datasets
Items Under Test	Person image datasets
Precautions	The person images should be retrieved from multiple videos

Special Conditions / Limitations	Image datasets with single camera will cause error and false prediction, due to insufficient of Training data.
Equipment / Facilities	Laptop
Data Recording	None
Acceptance Criteria	System successfully identifies the person from multiple videos.
Procedures	1. Insert image datasets from multiple videos with proper labels such as person ID and camera ID 2. Person identification will be done with the CNN model from the training dataset and query dataset.
Troubleshooting	Retrieve the images of person from multiple surveillance videos
Post-test Activities	None

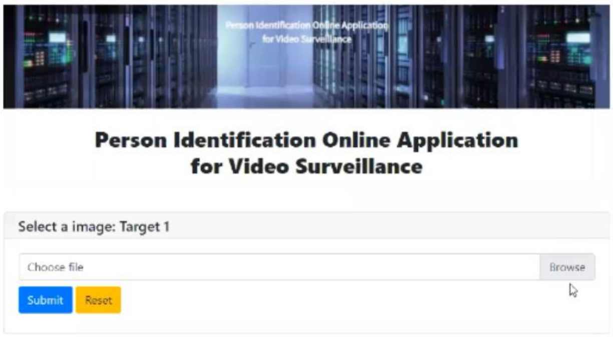
5.6 System Operation (with Screenshot)

Procedures to use the Person Identification Application:

1. User uploads image by clicking ‘Browse’ button, as shown in Figure 5-6-1.

2. User clicks ‘Submit’ button, as shown in Figure 5-6-1.

3. Repeat until all 8 images of target are uploaded.

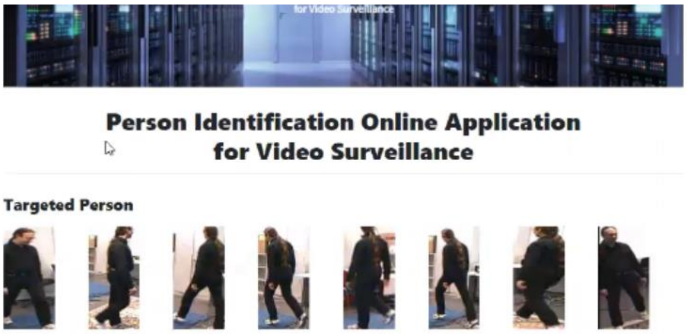


Person Identification Online Application for Video Surveillance

Select a image: Target 1

Figure 5-6-1: Uploading Images

4. User sees a row of uploaded images, as shown in Figure 5-6-2.



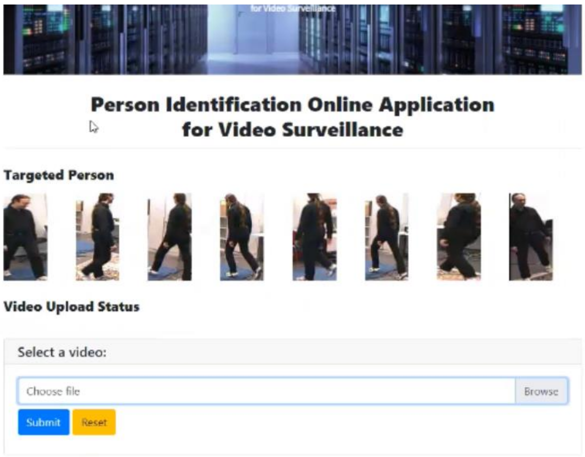
Person Identification Online Application for Video Surveillance

Targeted Person

Figure 5-6-2: Showing A Row of Uploaded Images

5. User uploads video by clicking ‘Browse’ button, as shown in Figure 5-6-3.

6. User clicks ‘Submit’ button, as shown in Figure 5-6-3.



Person Identification Online Application for Video Surveillance

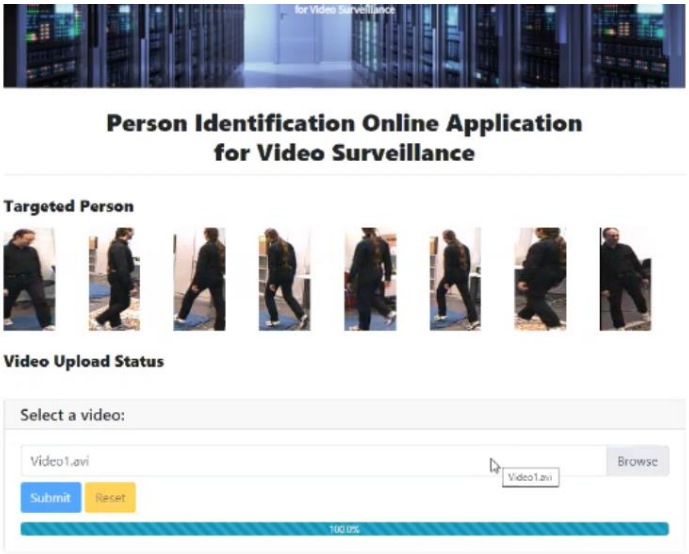
Targeted Person

Video Upload Status

Select a video:

Figure 5-6-3: Uploading Videos

7. User sees the loading bar, as shown in Figure 5-6-4.



Person Identification Online Application
for Video Surveillance

Targeted Person

Video Upload Status

Select a video:

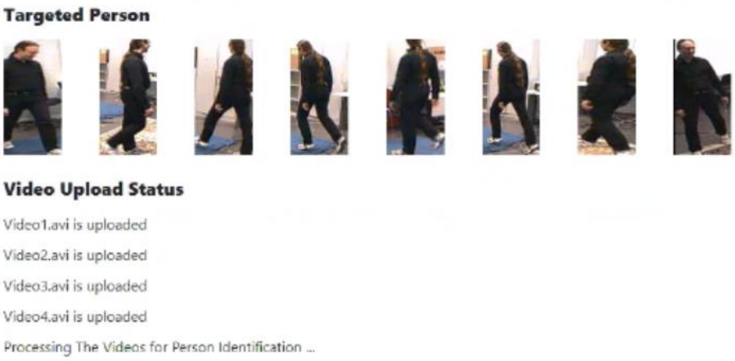
Video1.avi

Figure 5-6-4: Processing the Video for Upload

8. Repeat until all 4 videos for person identification are uploaded.

9. The user sees the Video Upload Status, once a video is upload and saved into directory successfully, as shown in Figure 5-6-5.

10. User sees the text that the system proceeds on person identification, as shown in Figure 5-6-5.



Targeted Person

Video Upload Status

Video1.avi is uploaded

Video2.avi is uploaded

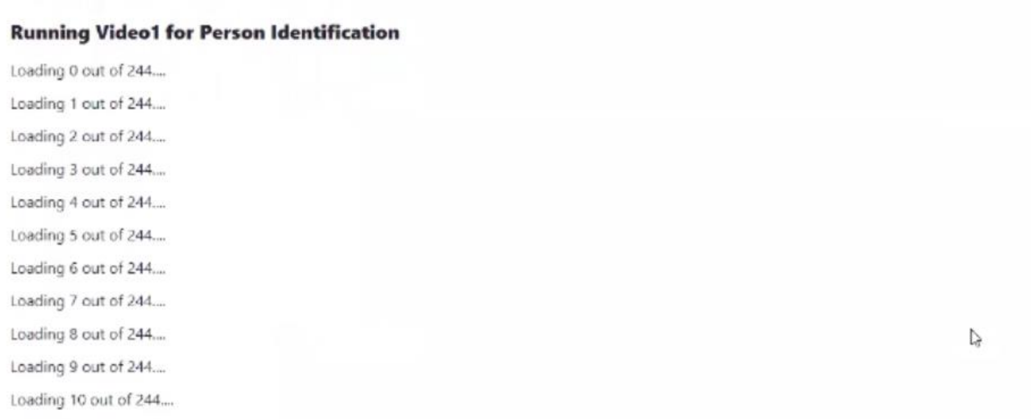
Video3.avi is uploaded

Video4.avi is uploaded

Processing The Videos for Person Identification ...

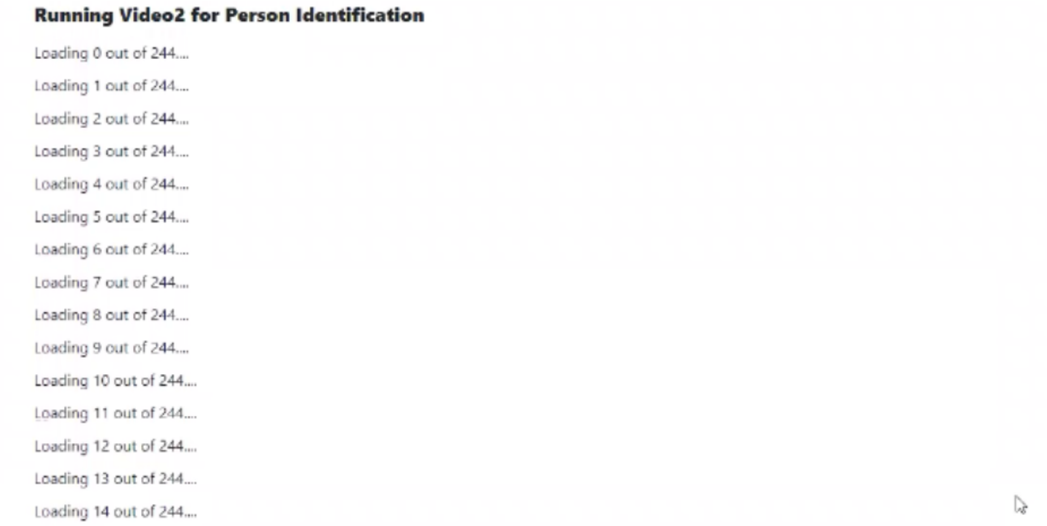
Figure 5-6-5: Video Uploading Status

11. User sees the person identification status of each video, as shown in Figure 5-6-6, Figure 5-6-7, Figure 5-6-8 and Figure 5-6-9.



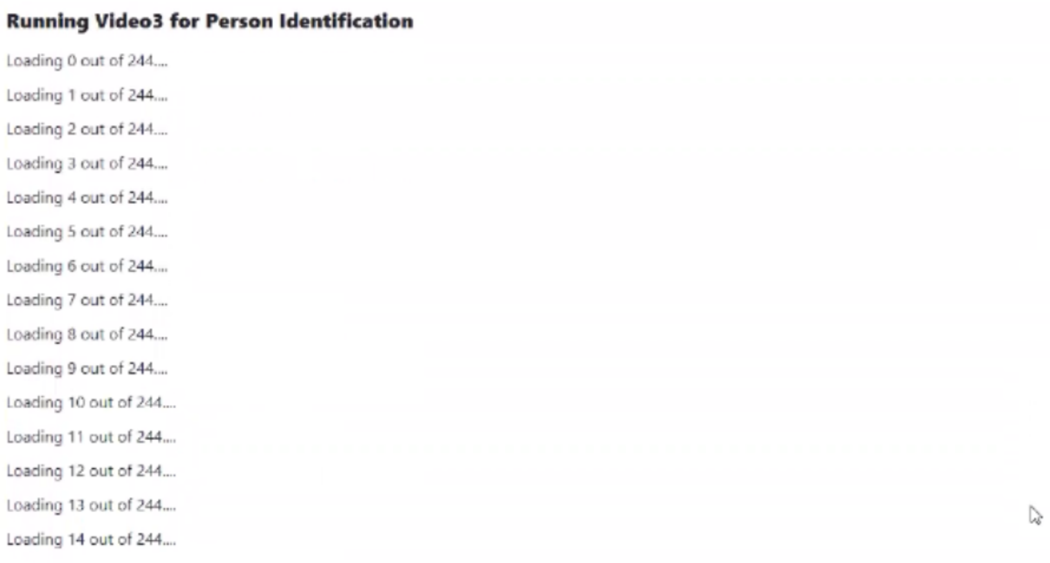
```
Running Video1 for Person Identification
Loading 0 out of 244....
Loading 1 out of 244....
Loading 2 out of 244....
Loading 3 out of 244....
Loading 4 out of 244....
Loading 5 out of 244....
Loading 6 out of 244....
Loading 7 out of 244....
Loading 8 out of 244....
Loading 9 out of 244....
Loading 10 out of 244....
```

Figure 5-6-6: Person Identification Status of Video 1



```
Running Video2 for Person Identification
Loading 0 out of 244....
Loading 1 out of 244....
Loading 2 out of 244....
Loading 3 out of 244....
Loading 4 out of 244....
Loading 5 out of 244....
Loading 6 out of 244....
Loading 7 out of 244....
Loading 8 out of 244....
Loading 9 out of 244....
Loading 10 out of 244....
Loading 11 out of 244....
Loading 12 out of 244....
Loading 13 out of 244....
Loading 14 out of 244....
```

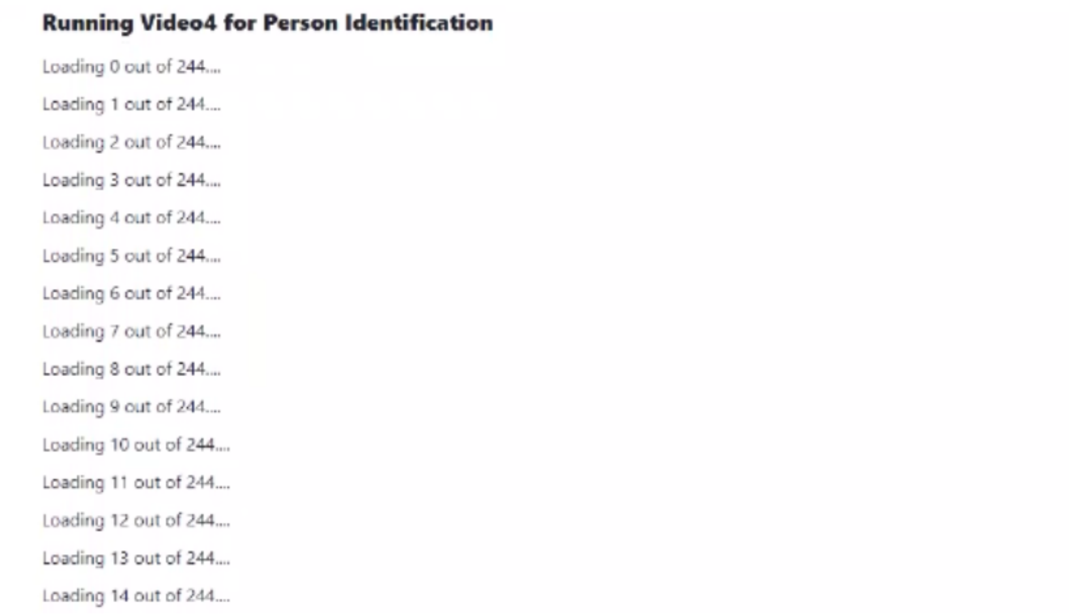
Figure 5-6-7: Person Identification Status of Video 2



Running Video3 for Person Identification

```
Loading 0 out of 244....
Loading 1 out of 244....
Loading 2 out of 244....
Loading 3 out of 244....
Loading 4 out of 244....
Loading 5 out of 244....
Loading 6 out of 244....
Loading 7 out of 244....
Loading 8 out of 244....
Loading 9 out of 244....
Loading 10 out of 244....
Loading 11 out of 244....
Loading 12 out of 244....
Loading 13 out of 244....
Loading 14 out of 244....
```

Figure 5-6-8: Person Identification Status of Video 3

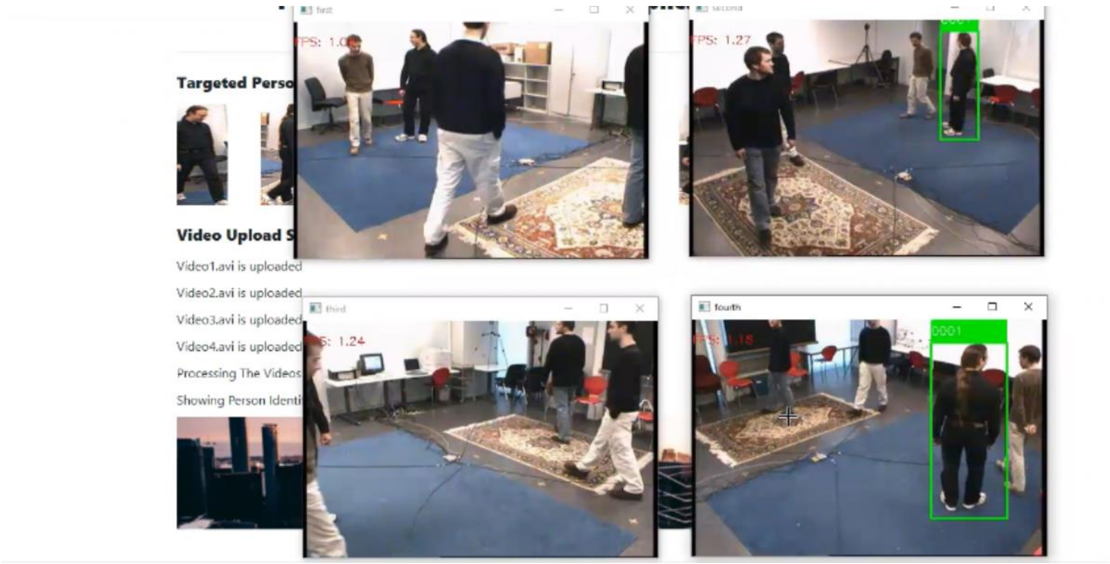


Running Video4 for Person Identification

```
Loading 0 out of 244....
Loading 1 out of 244....
Loading 2 out of 244....
Loading 3 out of 244....
Loading 4 out of 244....
Loading 5 out of 244....
Loading 6 out of 244....
Loading 7 out of 244....
Loading 8 out of 244....
Loading 9 out of 244....
Loading 10 out of 244....
Loading 11 out of 244....
Loading 12 out of 244....
Loading 13 out of 244....
Loading 14 out of 244....
```

Figure 5-6-9: Person Identification Status of Video 4

12. User watches the person identification videos concurrently, after person identification, as shown in Figure 5-6-10.



Video1.avi is uploaded
Video2.avi is uploaded
Video3.avi is uploaded
Video4.avi is uploaded
Processing The Videos
Showing Person Identi

Figure 5-6-10: Showing Person Identification Videos Concurrently

13. If user wishes to watch the 4 concurrent videos display again, click ‘Show Person Identification Videos’, as in Figure 5-6-11. User can watch the concurrent videos, as in Figure 5-6-10.

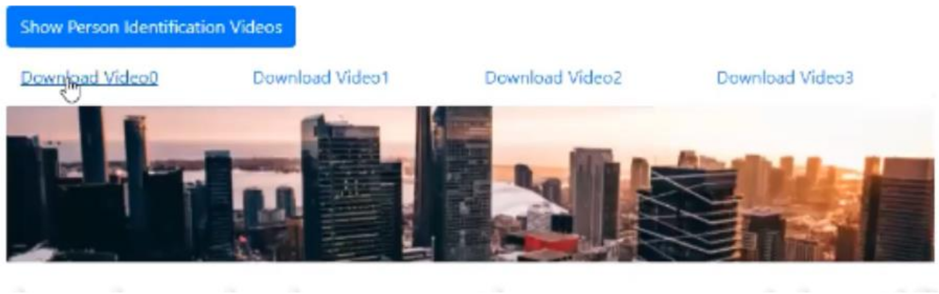
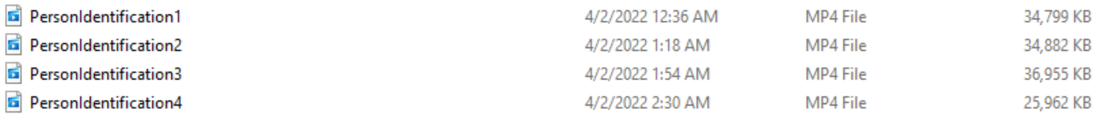


Figure 5-6-11: Download Person Identification Videos

14. If user wishes to download, click on the specific ‘Download Video’ hyperlink, for download, as in Figure 5-6-11. The downloaded videos will be in the ‘Downlaod’ directory of the user.



PersonIdentification1	4/2/2022 12:36 AM	MP4 File	34,799 KB
PersonIdentification2	4/2/2022 1:18 AM	MP4 File	34,882 KB
PersonIdentification3	4/2/2022 1:54 AM	MP4 File	36,955 KB
PersonIdentification4	4/2/2022 2:30 AM	MP4 File	25,962 KB

Figure 5-6-12: Downloaded Person Identification Videos in Download Directory

15. User watches the downloaded person identification videos, one by one, on the video player, as shown in Figure 5-6-13.

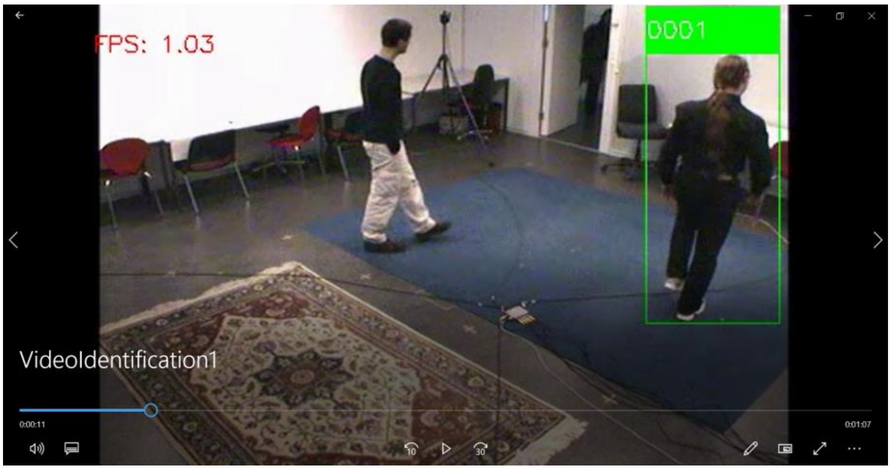


Figure 5-6-13: Downloaded VideoIdentification1

CHAPTER 6
SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing

When the images of the target were inserted, as shown in Figure 6-1-1, the target was predicted with a PID. Before target prediction or person prediction, as shown in Table 6-1-1, the model with 6 convolutional layers with dropout 0.2 was chosen, as it has the highest training accuracy of 0.9833 and validation accuracy of 0.9817. Furthermore, the duration of the model training was the shortest.

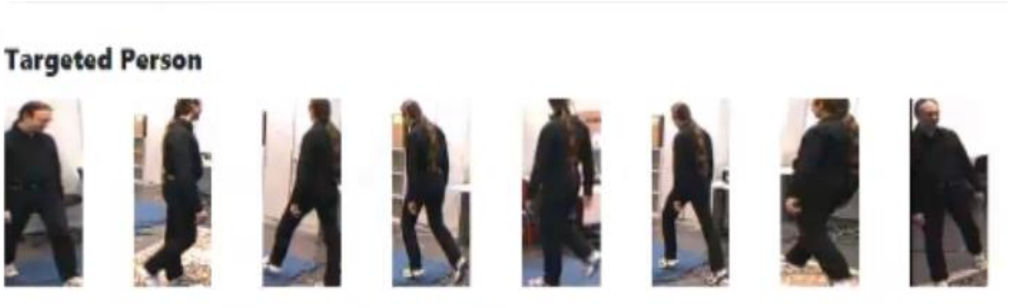


Figure 6-1-1: Target for Person Identification

Table 6-1-1: Training Accuracy, Training Loss, Validation Accuracy and Validation Loss

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Duration(s)
Model with 6 convolutional layers With Dropout 0.2	0.9833	0.0580	0.9817	0.0532	113

Both target identification and person identification worked with the trained model for prediction. The images of either target or cropped images from the videos were loaded from image into array form. Then, the image would be predicted using the trained CNN model. Through Softmax calculation, the highest score in the Softmax list was chosen with multiple of 100, accuracy score was calculated. Target which acquired the accuracy score, which was more than 80, would be assigned with the PID. The performance metrics in this application was accuracy, PID was assigned according to the accuracy score of the model prediction.

The person identification was done frame by frame in the video. In this frame, three persons were tracked and cropped for model prediction, as shown in Figure 6-1-2. Each of the cropped images was assigned with a PID. The PID of cropped and the PID of the target was compared, once the PID was the same. The green bounding box would be added to the predicted person, as shown in Figure 6-1-3.

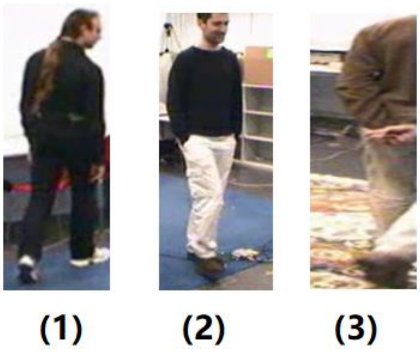


Figure 6-1-2: Persons Cropped

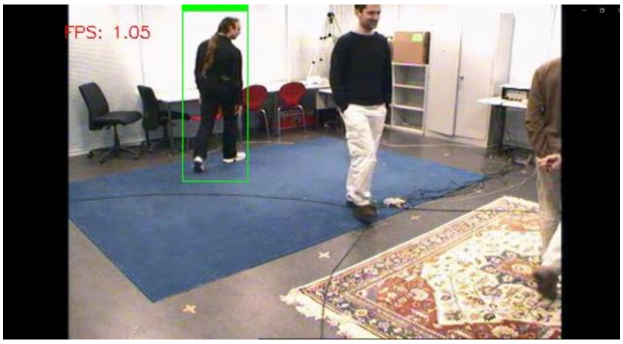


Figure 6-1-3: Person Identified in The Video

However, if the images of the target were predicted as None. It meant that the person dataset did not exist in the Training directory. Therefore, proceed for video tracking, to retrieve and save the cropped images into the Temporary Training (tmpTraining) directory, as shown in Figure 6-1-4. After that, train the model and predict the images of the target again. Once the target was assigned for PID, it proceeded to person identification.

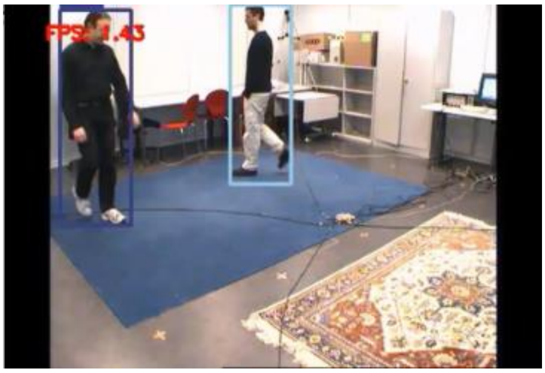


Figure 6-1-4: Person Tracking for Image Retrieval

Some of the frames in person identification videos, were as shown in Figure 6-1-5, Figure 6-1-6, Figure 6-1-7, and Figure 6-1-8. In the top left and top right of Figure 6-1-5, the person predicted was alone in the corner. However, in the bottom left and bottom right of Figure 6-1-5, the person predicted was near others. If they were to be closer or overlapped, it may lead to person occlusion, due to the weaknesses of DeepSORT tracker. However, in video 1, person occlusion did not happen. Therefore, the targeted person was predicted accurately. In Figure 6-1-6, the target was predicted correctly and there was no person occlusion issue, as he was alone and apart from the crowd in the frame.

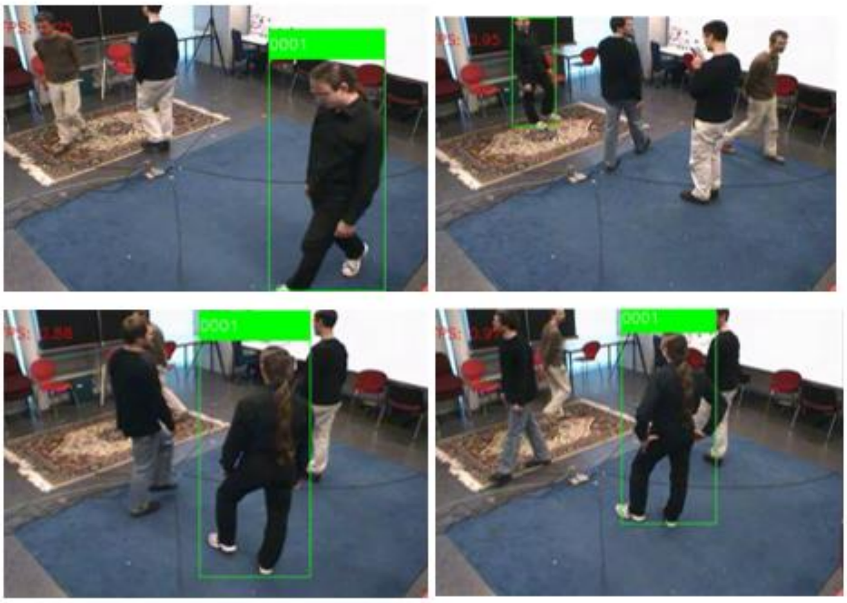


Figure 6-1-5: Frames in the Video Identification 1

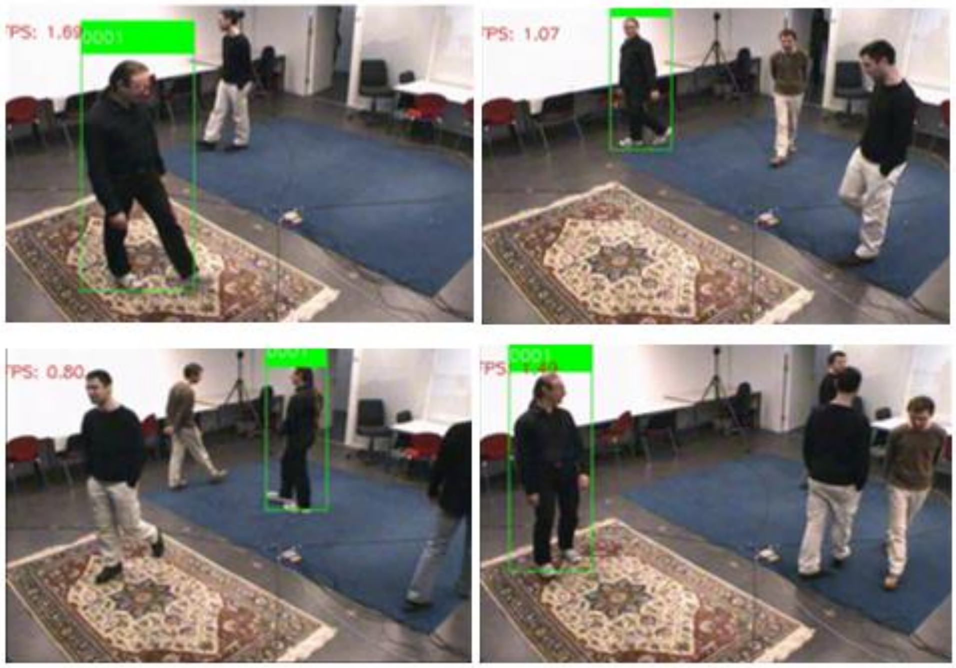


Figure 6-1-6: Frames in the Video Identification 2

In the top left and bottom left of Figure 6-1-7, the target person was away from crowd, so it was predicted correctly. However, in the top right, the person occlusion was nearly to happen, if each person was closer to the other. In the left bottom, the target showed his leg only, but it was still predicted correctly, as the list prediction method was implemented. List prediction method was that the PID was assigned to the person, according to the id assigned by the DeepSORT tracker. If the tracker id remained, the person would be predicted and tracked with the DeepSORT tracker id. In the top left, bottom left and top right of Figure 6-1-8, the target was predicted correctly, as he was alone and away from the crowd. In the top right, the target was still predicted correctly, although he was very close to others, as the list prediction method worked here. As long as the id assigned by the DeepSORT tracker remained, the PID remained the same, and the green bounding box would be bounded to the person.

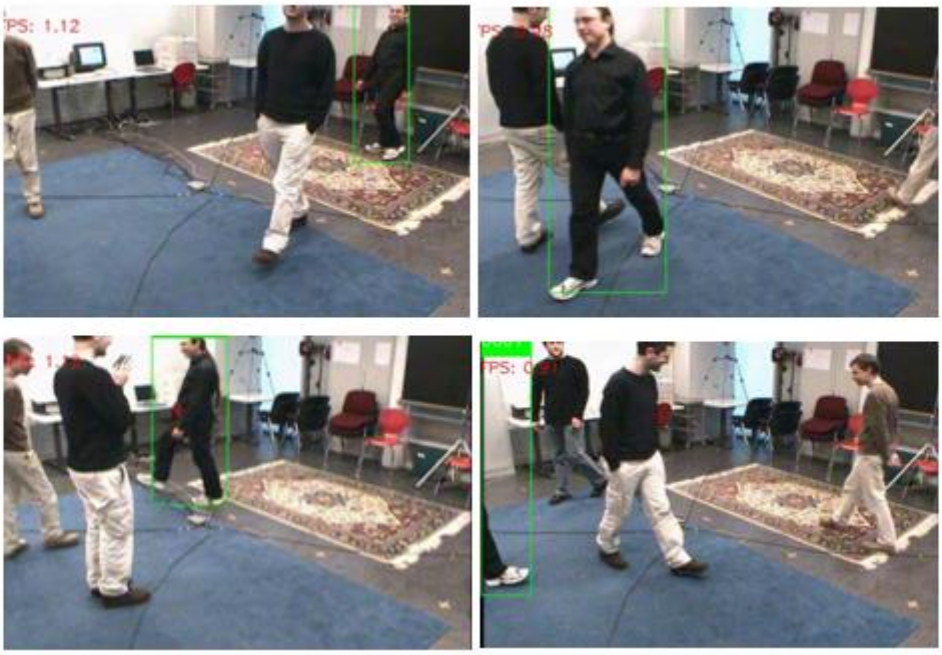


Figure 6-1-7: Frames in the Video Identification 3

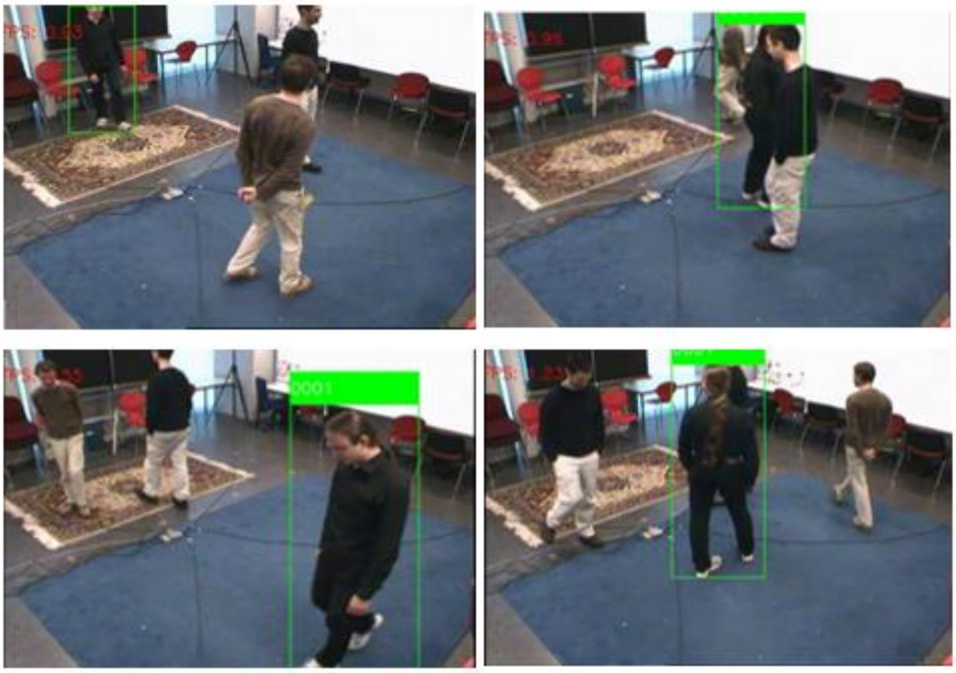


Figure 6-1-8: Frames in the Video Identification 4

6.2 Use Case Testing

Table 6-3-1: Use Case Testing

Use Case ID	**Flow**	**Inputs**	**Expected Outputs**	**Actual Outputs**	**Result (Pass /Fail)**
UC01	Main	Inserting 8 images of target	System displays row of 8 images of target	System displayed row of 8 images of target	Pass
UC02	Main	Inserting 4 videos	System displays the video uploading status	System displayed the video uploading status	Pass
UC03	Main	Clicking the 'Show Person Identification Videos' button	System displays the four-person identification videos concurrently	System displayed the four-person identification videos concurrently	Pass
UC04	Main	Clicking the 'Download Video' hyperlink	System allows user to download the video separately.	System allowed user to download the video separately. The download videos were in the 'Download' directory.	Pass

6.3 Objectives Evaluation

To develop a person identification application for video surveillance for monitoring suspicious people. The images of the target were inserted and predicted for PID. Through the CNN model, the person would be identified by using the embedding of features. CNN model acted an important role. Although there was no control on the feature to be trained in the CNN model, it learned itself to identify in model training. Once the PID was predicted, the application proceeded to the person identification person. When the person's ID and target ID were the same, the person in the video was cropped with a green bounding box.

To develop a model to perform person identification and tracking in surveillance video continuously, with the help of the DeepSORT tracker, the application could track the person using the id assigned by the tracker. With this advantage of DeepSORT, once the person in the videos was predicted, the person will be always assigned with the PID, according to this tracker ID. This reduced the frequency of model prediction and increased the effectiveness of the tracker. When the person was identified as the target, the id assigned by the tracker was highlighted, and the tracker detected this id, the green bounding box would be formed on the person with this id in the video.

To develop a model that performs person identification over multiple videos at the same moment, the trained CNN model allowed to predict and identify the person in different cameras with different angles accurately. The more the videos, the lesser the blind spots in a certain area. In addition, the environment in the surveillance videos must be similar, with less distinction. For example, the targeted person to be tracked must wear the same clothing in the videos, as clothing is one of the features for model training. After person identification in the videos, the person identification videos were displayed concurrently, as output.

6.4 Project Challenges

Challenges of new application development is evitable. The quality of person identification application will be affected by technical and environmental problems. In common, the challenges faced during person identification development were categorized into feature representation and feature matching. Feature representations are environmental factors such as the complex background, sunlight exposure, illumination, human posture, human position, shooting angle, crowd overlapping, etc. These may reduce the accuracy of person identification.

This person identification application required three main steps, which were person detection using YoloV3, person tracking using DeepSORT, and person identification using the CNN model. The challenge in DeepSORT was that the persons' identity would be swapped between each other when the persons are too close to each other. The DeepSORT tracker may change the bounding box and the id when the person occlusion happened. Before the persons were close to each other, the targeted person was bounded with green bounding box, as shown in Figure 6-4-1. However, when two persons were too close to each other, the green bounding box and id were transferred to another person, although initially the target whose arms with akimbo and wearing a pair of spectacles, was predicted, as shown in Figure 6-4-2. To reduce this issue happened, the application assigned the PID according to the id assigned by the DeepSORT tracker. As the PID was predicted at the first 15 images of that id assigned by the tracker, it reduced wrong person prediction issue to be happened. This list method reduced a huge amount of model prediction during person identification and the duration of person identification. At the same time, it reduced the swapping of person identity issues.

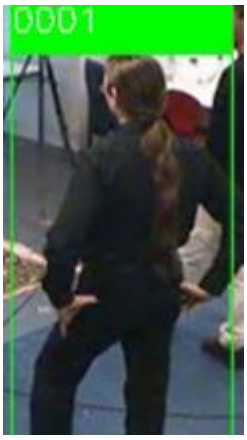


Figure 6-4-1: Predicted Accurately Before the Persons were Close

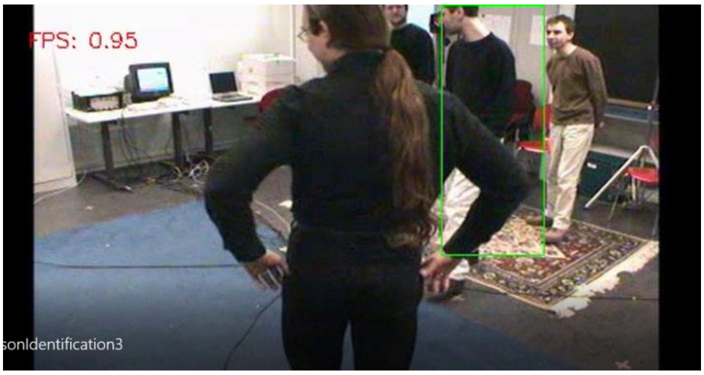


Figure 6-4-2: DeepSORT Tracker issue

After person prediction as None, saving the images into the Training directory was dangerous. The original data in the Training directory might be removed or corrupted accidentally if the application malfunctioned or was corrupted. Therefore, when the person was predicted as None, it would be saved in the PID directory of tmpTraining directory. Once the PID directory reached the requirement, such as reaching 50 images, this directory would be moved to the real Training directory, for model retraining. This reduced the in and out of the images or direct images removal in the main Training directory, which may lead to the risk of removal of wrong directories in the main Training directories. After the full person identification process, the directories in tmpTraining directories were removed.

The system suffered errors when the file type of the videos was inconsistent. When the user inserted the videos, the user may insert different video files type into the application. All the video files type must be the same. Therefore, the application allowed the user to choose the video file type before inserting the videos and saved into the directories. Although the response of users was collected, the application would recheck whether the videos inserted were .mp4 file type or .avi file type. If it was a .avi file type, it would be converted into a .mp4 file type, as.mp4 file type was the most common video file type. In the end, the file type of person identification videos was .mp4 file type.

The video which was inserted by the users was inconsistent. This challenge was divided into 2 parts, video resolution, and environment in the videos. The users were allowed to insert videos either high-quality videos or low-quality videos. Blurred videos, high light exposure videos, low light exposure videos and etc are examples of low-resolution videos. It may cause wrong identification or misidentification, as same person in different light exposure may look different. Therefore, the system was responsible to upscale low-resolution videos to high-resolution videos, with image enhancement frame by frame. However, if the changes in the environment between videos were distinct, background removal is necessary, to identify the person correctly.

In addition, this application cannot detect the person who was too small, as it was unable to be tracked as a person by the person detector, as they lost the features of a person. Therefore, the person might be missed out by the detector, as shown in the red circles in Figure 6-4-3. Another restriction is that, when the person was moving too fast, the detector was unable to detect the person clearly, as the fps of the video was low. The person might or might not be detected as a person, but it might be cropped and detected as something black. For example, the person who was circled with blue in Figure 6-4-3, he was with his skateboard. Due to the low resolution of the video, it was not predicted as a person.

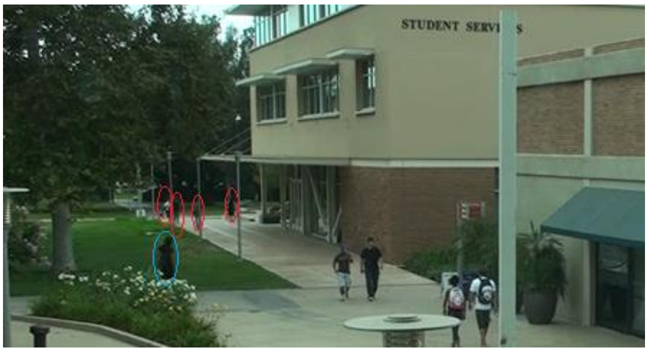


Figure 6-4-3: Undetected Person in The Frame.

Due to the availability of the resources on training datasets, it was a challenge to search for sufficient useful datasets for the development of person application identification. Most of the datasets such as Market-1501, MLR-VIPeR dataset, and CUHK03 dataset were from different countries, some countries like Saudi Arabia may have different environmental factors that other countries will not have. The face occlusion and person detection issue in Saudi Arabia may be highest, as the women in Saudi Arabia wear turbans which may cover parts of the face and only a pair of eyes are available to be seen. It was one of the challenges to face occlusion that needs to be taken into consideration. In this application, YoloV3 was applied for person detection, the person might not be identified as a person, as the dataset for model training did not include this issue in the training data. This was one of the issues of YoloV3 for person detection.

CHAPTER 7
CONCLUSION AND RECOMMENDATION

7.1 Project Review and Discussion

In this project 'Development of Person Identification Application for Video Surveillance', the inputs inserted by user are multiple surveillance videos and eight images of the target. After inserting the videos and images of the target, the videos were processed for person detection, person tracking, and person identification. In the end, user observed the videos with bounding boxes. The person in green bounding boxes was target, while the person which was not bounded with any frame was considered as pedestrians.

After inserting the videos, the system detected person using YOLOv3, which classifies object into 80 classes. As the main purpose of this project was to identify the person as target, the person class was detected only. In YOLOv3, NMS prevented the multiple detections, only those that had greater or equal value than NMS was considered as positive detections. Confidence was an important measure for the positive detection in the videos. The greater the confidence, the greater the accuracy of detections.

Person tracking implemented DeepSORT to track the person, with the nearest neighbour algorithm with the cosine distance metric. During person tracking, the images of tracked person were stored in training directories. When retrieving those images in the directories, the images were named properly with the person ID, camera ID, frame captured, image number, and 'PNG', which is the bitmap image format. After person tracking, the images of the person tracked in the DeepSORT tracker were retrieved and stored in the directories for person identification.

For person identification, CNN model was built for training. The images in Training directory were spilt into training dataset and validation dataset, with a ratio of

8:2. The image size was (64, 128). The model was built with 6 convolutional layers, with ReLU activation and followed by 2D max pooling. After that, flatten the input for dense layer with dimension of 1024. The dimension of last dense layer was the number of classes.

Before model training, data augmentation such as random flip and random crop was done to increase the positive variability and noises of the datasets without any additional cost. In addition, the classification layer of CNN was Sparse Categorical Cross Entropy, and the model optimizer was Adam. After setting up the model, the training of person identification was started. It would take some time for model training.

Through the CNN model, the person could be predicted correctly. By comparing the predicted target ID and the person PID in the video, draw the green bounding box for the person in the frame, when the target PID and person PID was same. Therefore, the person identification videos were done

7.2 Novelties and Contributions

1. The person identification application tracks the target in multiple videos in the public environment or in private environment.

In person tracking, the person can be tracked by the person detection tracker. It works well in a video. However, when the person was out of the video for few minutes and came back into the video recording area. The person was assigned a new id by the DeepSORT tracker [48]. In brief, the person was tracked as another person, although both are the same person. If this issue happens in a video, it will not work well in multiple videos. As the same person in multiple videos will be assigned with a different person ID in the person identification videos. In this project, this application solved this issue with the person identification, with the CNN model and the list prediction method. The CNN model was trained to identify the target in the videos. At the same moment, those persons with different person IDs were trained and classify into the same person

ID. To reduce this issue happened, the application implemented list prediction method. The list prediction method assigned the PID according to the id assigned by the DeepSORT tracker. Therefore, the target can be identified between target and non-target from pedestrians, with the implementation of person identification.

2. CNN model was implemented in video person identification.

Instead of Convolutional Neural Network (CNN) was proposed in [3], CNN was implemented for person identification. ResNet is an advanced version of CNN, to solve the vanishing gradient problems during backpropagation. The vanishing gradient diminishes the gradient of the model in each layer. However, Resnet required more time for model training, as Resnet has a greater number of parameters. Although the CNN does not have large number of parameters, it performed well during model prediction for person dentification. Therefore, CNN required less GPU memory. Furthermore, the duration to train CNN model is 6 times lesser than the Resnet. Therefore, CNN model works better compared to Resnet, in this application.

7.3 Conclusion with Supportive Remarks

The person identification for video surveillance application reduces the manpower on video surveillance, reduces human error and improves the accuracy of person identification in video. From the receiving input from users until presenting outputs for user, the input such as videos were processed in several stages such as person detection, person tracking, and person identification. First, users are requested to videos and images of target for person identification in multiple videos. Second, the videos were processed with YOLOv3 algorithm for person detection. Third, the persons detected in person detection were tracked in person tracking. Person tracking implemented DeepSORT with the Nearest Neighbour algorithm. The tracker in DeepSORT tracked and stored the information of the track such as track ID, class type, and bounding box information. Forth, person identification was implemented with the self-implemented CNN model. The CNN model was built and trained with the Sparse Categorical Cross Entropy classification layer and Adam optimizer. For the target prediction, it implements Softmax. In the end, as shown in Figure 7-3-1, the output is the videos with green. The green bounding box indicates that the person in the bounding box is the target.

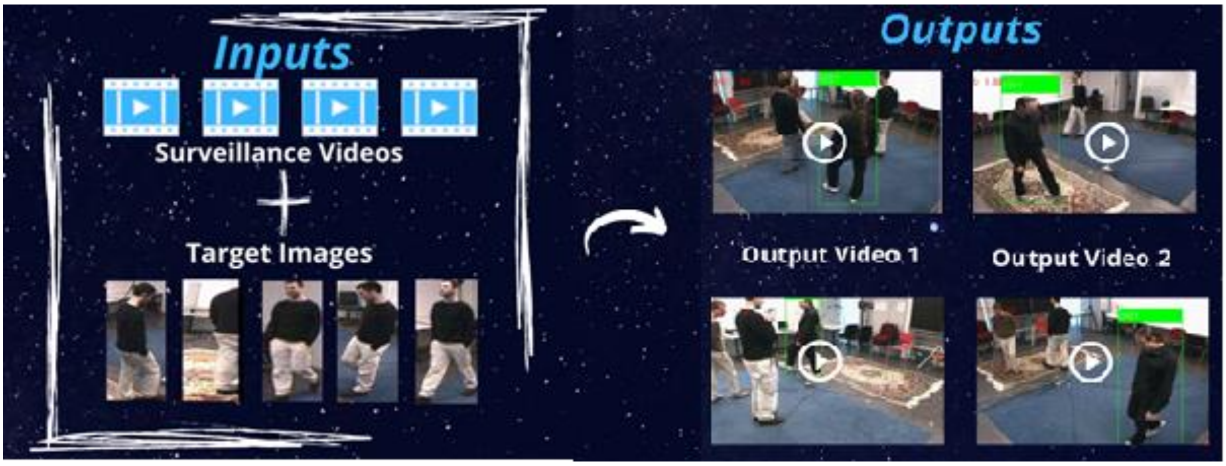


Figure 7-3-1: Person identification in Video Surveillance Application

7.4 Future Work

Due to time restrictions, some improvement which requires further research will be done in the future. First, the person identification process required long time to perform completely, as the DeepSORT tracker may requires a better GPU to run faster. Although the trained model works well to predict the target images and cropped person in the videos, the processing time required by the DeepSORT in both person tracking and person identification is slightly longer. Second, another improvement can be done in the future is that the ID between person will be interchanged unconsciously. This may lead to some false predictions by the model, as a PID directory might have cropped images of more than 1 person. To enhance this application, replace DeepSORT with a better person tracker.

All the images cropped and saved into the Training directory were with the background. Hence the background may reduce the accuracy during prediction, as different background has different features such as color and texture, it is noisy during model training and model prediction. When the background is removed, the images are left with person only. Therefore, if the time was sufficient, background removal may increase the accuracy of the person identification application.

REFERENCES

[1] S.-I. Yu, D. Meng, W. Zuo, and A. Hauptmann, “The solution path algorithm for identity-aware multi-object tracking,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] Y. Balasubramanian, N. Chandrasekaran, S. Asokan, and S. Sri Subramanian, “Deep-facial feature-based person reidentification for authentication in surveillance applications,” in Visual Object Tracking with Deep Neural Networks, IntechOpen, 2019.

[3] W. Song, Y. Wu, J. Zheng, C. Chen, and F. Liu, “Extended global–local representation learning for video person re-identification,” IEEE Access, vol. 7, pp. 122684–122696, 2019.

[4] “100,000 years of identity verification: an infographic history,” https://www.trulioo.com. [Online]. Available: https://www.trulioo.com/blog/infographic-the-history-of-id-verification. [Accessed: 23-Mar-2022].

[5] Ö. Toygar, E. Alqaralleh, and A. Afaneh, “Person identification using multimodal biometrics under different challenges,” in Human-Robot Interaction - Theory and Application, InTech, 2018.

[6] ThemeGrill, “NEC Person Re-Identification Technology can recognize by body shape,” Robot News, 11-Feb-2019. [Online]. Available: https://yellrobot.com/nec-person-reidentification-technology-can-identify-body-shape. [Accessed: 23-Mar-2022].

[7] L. Zhang, K. Li, Y. Qi, and F. Wang, “Local feature extracted by the improved bag of features method for person re-identification,” Neurocomputing, vol. 458, pp. 690–700, 2021.

[8] E. Yaghoubi, A. Kumar, and H. Proença, “SSS-PR: A short survey of surveys in person re-identification,” Pattern Recognit. Lett., vol. 143, pp. 50–57, 2021.

[9] H.-Q. Nguyen, T.-B. Nguyen, and T.-L. Le, “Robust person re-identification through the combination of metric learning and late fusion techniques,” Vietnam J. Comput. Sci., vol. 08, no. 03, pp. 397–41

[10] M. Li, Z. Zhou, and X. Liu, “3D hypothesis clustering for cross-view matching in multi-person motion capture,” Comput. Vis. Media (Beijing), vol. 6, no. 2, pp. 147–156, 2020.5, 2021.

[11] K. Nandakumar, S. Blandin, and L. Wynter, “High-frequency crowd insights for public safety and congestion control,” arXiv [cs.CV], 2019.

[12] M. Adil, S. Mamoon, A. Zakir, M. A. Manzoor, and Z. Lian, “Multi scale-adaptive super-resolution person re-identification using GAN,” IEEE Access, vol. 8, pp. 177351–177362, 2020.

[13] Yearbook Machine Ltd, “Image resolution explained,” Yearbook.com. [Online]. Available: https://yearbook.com/support/article/image-resolution-explained. [Accessed: 23-Mar-2022].

[14] Y. Sun, Z. Dou, Y. Li, and S. Wang, “Improving semantic part features for person re-identification with supervised non-local similarity,” Tsinghua Sci. Technol., vol. 25, no. 5, pp. 636–646, 2020.

[15] Y. Zhang, X. Gu, J. Tang, K. Cheng, and S. Tan, “Part-based attribute-aware network for person re-identification,” IEEE Access, vol. 7, pp. 53585–53595, 2019.

[16] H. Sheng et al., “Mining hard samples globally and efficiently for person reidentification,” IEEE Internet Things J., vol. 7, no. 10, pp. 9611–9622, 2020.

[17] M. Cao, C. Chen, X. Hu, and S. Peng, “Towards fast and kernelized orthogonal discriminant analysis on person re-identification,” Pattern Recognit., vol. 94, pp. 218–229, 2019.

[18] "Eigenvector and eigenvalue," Mathsisfun.com. [Online]. Available: https://www.mathsisfun.com/algebra/eigenvalue.html. [Accessed: 23-Mar-2022].

[19] A. J. Ma, J. Li, P. C. Yuen, and P. Li, "Cross-domain person reidentification using domain adaptation ranking SVMs," IEEE Trans. Image Process., vol. 24, no. 5, pp. 1599–1613, 2015.

[20] R. Guo, C. Lin, C.-G. Li, and J. Lin, "Deep group-shuffling dual random walks with label smoothing for person reidentification," IEEE Access, vol. 8, pp. 40018–40028, 2020.

[21] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv [cs.CV], 2018.

[22] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Mini-YOLOv3: Real-time object detector for embedded applications," IEEE Access, vol. 7, pp. 133529–133538, 2019.

[23] Manishgupta, "YOLO — you only look once," Towards Data Science, 30-May-2020. [Online]. Available: https://towardsdatascience.com/yolo-you-only-look-once-3dbdbb608ec4. [Accessed: 23-Mar-2022].

[24] S. R. Maiya, "DeepSORT: Deep Learning to track custom objects in a video," AI & Machine Learning Blog, 19-Jul-2019. [Online]. Available: https://nanonets.com/blog/object-tracking-deepsort/. [Accessed: 23-Mar-2022].

[25] H. A. Abu Alfeilat et al., "Effects of distance measure choice on K-nearest neighbor classifier performance: A review," Big Data, vol. 7, no. 4, pp. 221–248, 2019.

[26] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in 2017 IEEE International Conference on Image Processing (ICIP), 2017.

[27] K. Zhou and T. Xiang, “Torchreid: A library for deep learning person re-identification in PyTorch,” arXiv [cs.CV], 2019.

[28] M. Bedeli, Z. Geradts, and E. van Eijk, “Clothing identification via deep learning: forensic applications,” Forensic Sci. Res., vol. 3, no. 3, pp. 219–229, 2018.

[29] M. E. Shacklett, “What is Dropout? Understanding Dropout in Neural Networks,” SearchEnterpriseAI, 05-Mar-2021. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/dropout. [Accessed: 23-Mar-2022].

[30] “Optimization with ADAM and RMSprop in Convolution neural Network (CNN): A Case study for Telugu Handwritten Characters,” Int. j. emerg. trends eng. res., vol. 8, no. 9, pp. 5116–5121, 2020.

[31] Leakyrelu.com. [Online]. Available: http://leakyrelu.com/2020/01/01/difference-between-categorical-and-sparse-categorical-cross-entropy-loss-function. [Accessed: 23-Mar-2022].

[32] X. Ye and Q. Zhu, “Class-incremental learning based on feature extraction of CNN with optimized softmax and one-class classifiers,” IEEE Access, vol. 7, pp. 42024–42031, 2019.

[33] “Managing Data,” in Python® Projects, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 103–160.

[34] “PyWebIO — PyWebIO 1.5.2 documentation,” Readthedocs.io. [Online]. Available: https://pywebio.readthedocs.io/en/latest/. [Accessed: 23-Mar-2022].

[35] “OpenCV - overview,” GeeksforGeeks, 23-Sep-2019. [Online]. Available: https://www.geeksforgeeks.org/opencv-overview/. [Accessed: 24-Mar-2022].

[36] S. R. Maiya, “DeepSORT: Deep Learning to track custom objects in a video,” AI & Machine Learning Blog, 19-Jul-2019. [Online]. Available: https://nanonets.com/blog/object-tracking-deepsort/. [Accessed: 23-Mar-2022].

[37] “io — Core tools for working with streams — Python 3.10.3 documentation,” Python.org. [Online]. Available: https://docs.python.org/3/library/io.html. [Accessed: 24-Mar-2022].

[38] “Matplotlib — Visualization with Python,” Matplotlib.org. [Online]. Available: https://matplotlib.org/. [Accessed: 24-Mar-2022].

[39] “What is NumPy? — NumPy v1.22 Manual,” Numpy.org. [Online]. Available: https://numpy.org/doc/stable/user/whatisnumpy.html. [Accessed: 24-Mar-2022].

[40] “socket — Low-level networking interface — Python 3.10.3 documentation,” Python.org. [Online]. Available: https://docs.python.org/3/library/socket.html. [Accessed: 24-Mar-2022].

[41] R. Python, “Python 3’s pathlib Module: Taming the File System,” Realpython.com, 23-Apr-2018. [Online]. Available: https://realpython.com/python-pathlib/. [Accessed: 24-Mar-2022].

[42] “Python PIL,” GeeksforGeeks, 15-Jul-2019. [Online]. Available: https://www.geeksforgeeks.org/python-pil-image-open-method/. [Accessed: 24-Mar-2022].

[43] “shutil — High-level file operations — Python 3.10.3 documentation,” Python.org. [Online]. Available: https://docs.python.org/3/library/shutil.html. [Accessed: 24-Mar-2022].

[44] “OS module in python with examples,” GeeksforGeeks, 21-Nov-2016. [Online]. Available: https://www.geeksforgeeks.org/os-module-python-examples/. [Accessed: 24-Mar-2022].

[45] “TensorFlow core,” TensorFlow. [Online]. Available: https://www.tensorflow.org/overview. [Accessed: 24-Mar-2022].

[46] A. Geitgey, “Machine learning is fun! Part 4: Modern face recognition with deep learning,” Medium, 24-Jul-2016. [Online]. Available:

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78. [Accessed: 23-Mar-2022].

[47] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[48] M. Ahmad, I. Ahmed, F. A. Khan, F. Qayum, and H. Aljuaid, "Convolutional neural network–based person tracking using overhead views," Int. J. Distrib. Sens. Netw., vol. 16, no. 6, p. 155014772093473, 2020.

APPENDIX

A-1 Weekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 3
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Remove the TorchREID library and implement with customized model.

2. WORK TO BE DONE

Complete the front-end development, overall development and the testing

3. PROBLEMS ENCOUNTERED

The video rans in the middle will encounter opencv library error issues, it will be solved as soon as possible.

4. SELF EVALUATION OF THE PROGRESS

Need to increase the speed of implementation, as two weeks has been wasted. Schedule the flow of the project and project documentation properly.

Phaikching

Supervisor's signature

Student's signature

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 4
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Same as the before, person tracking, and person identification had been done.

2. WORK TO BE DONE

Implement transfer learning and fine tune the model for the new ID identified.
Adapt the model for any person identification video
Store IDs in a track list, as a person may have multiple IDs

3. PROBLEMS ENCOUNTERED

Not to train the model once the new videos is implemented but apply with transfer learning method.

4. SELF EVALUATION OF THE PROGRESS

Need to increase the speed of implementation.
Schedule the flow of the project and project documentation properly.

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 5
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Person Identification system was upgraded with the lists implementation.
The system was improved by using the model prediction.

2. WORK TO BE DONE

Build the UI design with the Jupyter notebook interface.
Try more demo situation (maybe outdoor and indoor environment)

3. PROBLEMS ENCOUNTERED

The DeepSORT tracker is very slow.
The person identification model has some minor issues to be repaired and fixed.

4. SELF EVALUATION OF THE PROGRESS

Need to speed up and complete the implementation and testing by Week 8.
Focus more on person tracking and person identification model issues (main issues).

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 6
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Basic front-end design and files upload had been done.

2. WORK TO BE DONE

Connect Visual Studio Code (Front end) and Jupyter Notebook (Back end)

Try to figure out how to remove the background, if possible.

2. PROBLEMS ENCOUNTERED

Discover ways to connect visual studio code and Jupyter notebook.

The accuracy is high, but it still made mistakes.

Solve the retraining model issues.

4. SELF EVALUATION OF THE PROGRESS

Be more efficient and focus more on the model and accuracy.

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 7
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Predict the person using the person id from the trackers.
Able to retrain the model for new person who comes in.
Create first version of user interface.

2. WORK TO BE DONE

Remove those bounding box, which confused the model.
Control the model with more restrictions, to prevent too much model training.

3. PROBLEMS ENCOUNTERED

The lines of the bounding boxes reduce the accuracy of the person identification.
Try to remove the background to increase the accuracy.

4. SELF EVALUATION OF THE PROGRESS

Need to proceed faster and solve more internal complex issues, such as bounding box and UI design.

Phaikching

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 8
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

The model system for the application is well connected with UI using Voila.

2. WORK TO BE DONE

Finish and complete the video UI.
Start working on the report.
Prepare for the project demo.

3. PROBLEMS ENCOUNTERED

UI for videos are slightly too complex and complicated to be done, as it exceeds 50MB.

4. SELF EVALUATION OF THE PROGRESS

The basic implementation had done, but the project still require enhancement.
Start writing FYP1 report.

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 9
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Propose on prototype (30%).

2. WORK TO BE DONE

Start writing FYP1 report especially Chapter 3.

3. PROBLEMS ENCOUNTERED

Not start writing report.

4. SELF EVALUATION OF THE PROGRESS

Start writing report based on the prototype, to prevent delay.

Phaikching

Supervisor's signature

Student's signature

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 10
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

1. WORK DONE

Draft FYP2 report

2. WORK TO BE DONE

Complete and improve the FYP2 report, especially Chapter 6.

3. PROBLEMS ENCOUNTERED

More details and diagrams in the report, for better understanding.

4. SELF EVALUATION OF THE PROGRESS

Be more effective in doing report.

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 11
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

2. WORK DONE

FYP2 Report in progress.

2. WORK TO BE DONE

Complete the FYP2 report according to the report templates.
Check the FYP report with Turnitin Plagiarism Checker.

3. PROBLEMS ENCOUNTERED

Follow the report format as provided in the report templates.

4. SELF EVALUATION OF THE PROGRESS

Be more effective time and have better time management in doing report.

Phaikching

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Semester 3, Year 3	**Study week no.:** 12
Student Name & ID: Soon Phaik Ching (18ACB03005)	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development of Person Identification Application for Video Surveillance	

3. WORK DONE

Final check on FYP2 Report.
Mock Presentation.

2. WORK TO BE DONE

Final check on FYP2 Report.
Prepare for presentation and report submission.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

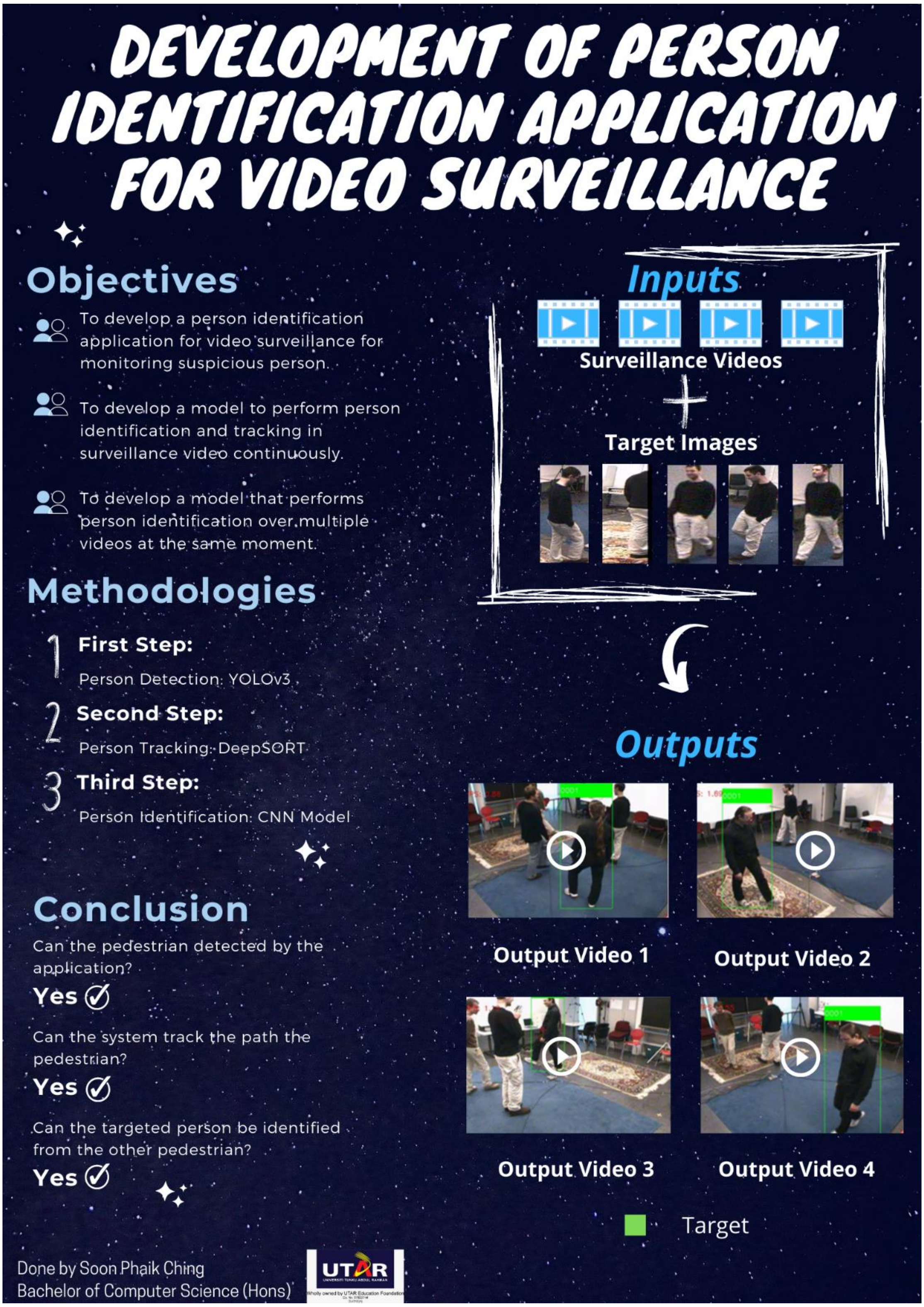
Submit the report on time.

Phaikching

Supervisor's signature

Student's signature

A-2 Poster



DEVELOPMENT OF PERSON IDENTIFICATION APPLICATION FOR VIDEO SURVEILLANCE

Objectives

- To develop a person identification application for video surveillance for monitoring suspicious person.
- To develop a model to perform person identification and tracking in surveillance video continuously.
- To develop a model that performs person identification over multiple videos at the same moment.

Methodologies

1 **First Step:**

Person Detection: YOLOv3

2 **Second Step:**

Person Tracking: DeepSORT

3 **Third Step:**

Person Identification: CNN Model

Conclusion

Can the pedestrian detected by the application?

Yes ✓

Can the system track the path the pedestrian?

Yes ✓

Can the targeted person be identified from the other pedestrian?

Yes ✓

Inputs

Surveillance Videos

+

Target Images

Outputs

Output Video 1

Output Video 2

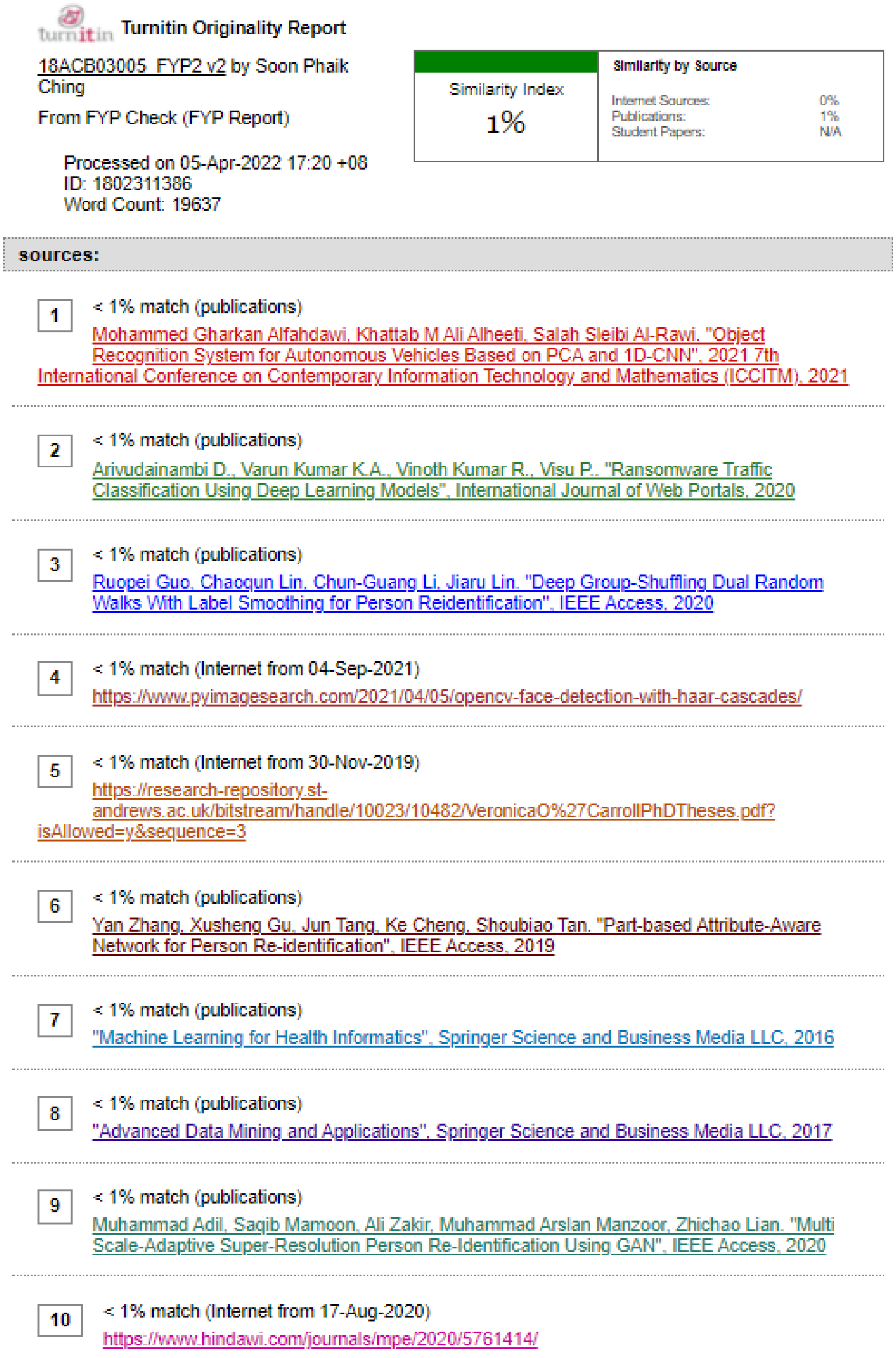
Output Video 3

Output Video 4

🟩 Target

Done by Soon Phaik Ching
Bachelor of Computer Science (Hons)

PLAGIARISM CHECK RESULT



Turnitin Originality Report

18ACB03005 FYP2 v2 by Soon Phaik Ching

From FYP Check (FYP Report)

Processed on 05-Apr-2022 17:20 +08
ID: 1802311386
Word Count: 19637

Similarity Index
1%

Similarity by Source

Internet Sources:	0%
Publications:	1%
Student Papers:	N/A

sources:

1 < 1% match (publications)
Mohammed Gharkan Alfahdawi, Khattab M Ali Alheeti, Salah Sleibi Al-Rawi. "Object Recognition System for Autonomous Vehicles Based on PCA and 1D-CNN", 2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM), 2021

2 < 1% match (publications)
Arivudainambi D., Varun Kumar K.A., Vinoth Kumar R., Visu P.. "Ransomware Traffic Classification Using Deep Learning Models", International Journal of Web Portals, 2020

3 < 1% match (publications)
Ruopei Guo, Chaoqun Lin, Chun-Guang Li, Jiaru Lin. "Deep Group-Shuffling Dual Random Walks With Label Smoothing for Person Reidentification", IEEE Access, 2020

4 < 1% match (Internet from 04-Sep-2021)
https://www.pyimagesearch.com/2021/04/05/opencv-face-detection-with-haar-cascades/

5 < 1% match (Internet from 30-Nov-2019)
https://research-repository.st-andrews.ac.uk/bitstream/handle/10023/10482/VeronicaO%27CarrollPhDTheses.pdf?isAllowed=y&sequence=3

6 < 1% match (publications)
Yan Zhang, Xusheng Gu, Jun Tang, Ke Cheng, Shoubiao Tan. "Part-based Attribute-Aware Network for Person Re-identification", IEEE Access, 2019

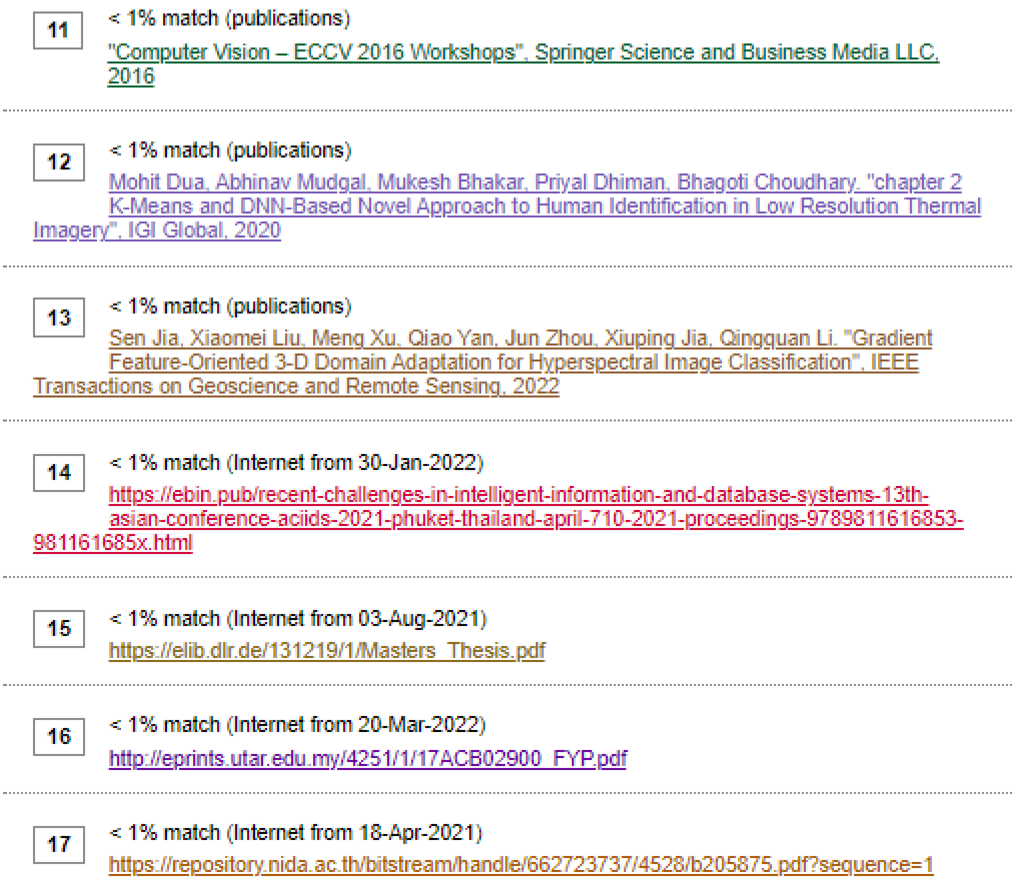
7 < 1% match (publications)
"Machine Learning for Health Informatics", Springer Science and Business Media LLC, 2016

8 < 1% match (publications)
"Advanced Data Mining and Applications", Springer Science and Business Media LLC, 2017

9 < 1% match (publications)
Muhammad Adil, Saqib Mamoon, Ali Zakir, Muhammad Arslan Manzoor, Zhichao Lian. "Multi Scale-Adaptive Super-Resolution Person Re-Identification Using GAN", IEEE Access, 2020

10 < 1% match (Internet from 17-Aug-2020)
https://www.hindawi.com/journals/mpe/2020/5761414/

PLAGIARISM CHECK RESULT



11 < 1% match (publications)
"Computer Vision – ECCV 2016 Workshops", Springer Science and Business Media LLC, 2016

12 < 1% match (publications)
Mohit Dua, Abhinav Mudgal, Mukesh Bhakar, Priyal Dhiman, Bhagoti Choudhary. "chapter 2 K-Means and DNN-Based Novel Approach to Human Identification in Low Resolution Thermal Imagery", IGI Global, 2020

13 < 1% match (publications)
Sen Jia, Xiaomei Liu, Meng Xu, Qiao Yan, Jun Zhou, Xiuping Jia, Qingquan Li. "Gradient Feature-Oriented 3-D Domain Adaptation for Hyperspectral Image Classification", IEEE Transactions on Geoscience and Remote Sensing, 2022

14 < 1% match (Internet from 30-Jan-2022)
https://ebin.pub/recent-challenges-in-intelligent-information-and-database-systems-13th-asian-conference-aciids-2021-phuket-thailand-april-710-2021-proceedings-9789811616853-981161685x.html

15 < 1% match (Internet from 03-Aug-2021)
https://elib.dlr.de/131219/1/Masters_Thesis.pdf

16 < 1% match (Internet from 20-Mar-2022)
http://eprints.utar.edu.my/4251/1/17ACB02900_FYP.pdf

17 < 1% match (Internet from 18-Apr-2021)
https://repository.nida.ac.th/bitstream/handle/662723737/4528/b205875.pdf?sequence=1

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	SOON PHAIK CHING
ID Number(s)	18ACB03005
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	DEVELOPMENT OF PERSON IDENTIFICATION APPLICATION FOR VIDEO SURVEILLANCE

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 1 % **Similarity by source** Internet Sources: 0 % Publications: 1 % Student Papers: NA %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required, and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and
(ii) Matching of individual sources listed must be less than 3% each, and
(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Dr Ng Hui Fuang	Name:
Date: 20/04/2022	Date:

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB03005
Student Name	SOON PHAIK CHING
Supervisor Name	DR NG HUI FUANG

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
/	Title Page
/	Signed Report Status Declaration Form
/	Signed FYP Thesis Submission Form
/	Signed form of the Declaration of Originality
/	Acknowledgement
/	Abstract
/	Table of Contents
/	List of Figures (if applicable)
/	List of Tables (if applicable)
	List of Symbols (if applicable)
/	List of Abbreviations (if applicable)
/	Chapters / Content
/	Bibliography (or References)
/	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
/	Weekly Log
/	Poster
/	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
/	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

PLAGIARISM CHECK RESULT

Khaikching

(Signature of Student)
Date: 20/04/2022