**Republic of Iraq**
**Ministry of Higher Education and Scientific Research**
**University of Babylon**
**College of Information Technology**
**Software Department**

# *A Developed Prediction Model for Moving Object Direction and Tracking across Non-Overlapping Topology of Surveillance Cameras Network Using Deep Learning*

A Dissertation
Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology/Software

***By***

**Wael Mahdi Brich Galgan**

***Supervised by***

**Prof. Dr.Israa Hadi Ali Hussein**

2022 A.D 1444 A.H

بسم الله الرحمن الرحيم

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ
دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ۝

صدق الله العظيم

[سوره المجادله: 11]

# Declaration

The work in this dissertation, ***A Developed Prediction Model for Moving Object Direction and Tracking across Non-Overlapping Topology of Surveillance Cameras Network Using Deep Learning***, is original and no portion of the work referred to here has been submitted in support of an application for another degree or qualification of this or any other university or institution of learning.

Signature:

Date: / /2022

Wael Mahdi Brich

## Supervisor Certification

I certify that the dissertation entitled (***A Developed Prediction Model for Moving Object Direction and Tracking across Non-Overlapping Topology of Surveillance Cameras Network Using Deep Learning***) was prepared under my supervision at the department of Software/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Doctorate of Philosophy in Information Technology-Software

Signature:
Supervisor Name: Prof. Dr. Israa Hadi Ali

Date: / / 2022

**The Head of the Department Certification**

In view of the available recommendations, I forward the Dissertation entitled "***A Developed Prediction Model for Moving Object Direction and Tracking across Non-Overlapping Topology of Surveillance Cameras Network Using Deep Learning***" for debate by the examination committee.

Signature:

Prof. Dr. Ahmed Saleem Abbas

Head of Software Department

Date: / / 2022

# Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled ( ***A Developed Prediction Model for Moving Object Direction and Tracking across Non-Overlapping Topology of Surveillance Cameras Network Using Deep Learning***) presented by the student (Wael Mahdi Brich) and examined him in its content and what is related to it, and that, in our opinion, it is adequate with the field of study as a dissertation for the degree of Doctor of Philosophy in Information Technology-Software.

Signature:
Name: Dr. Jamila Harbi Suad
Title: Prof Date:
/ / 2022
(**Chairman**)

Signature:
Name: Dr. Dheyaa Shaheed Al-Azzawi
Title: Prof
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Mehdi Ebady Manaa
Title: Asst. Prof
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Majid Jabbar Jawad
Title: Prof
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Alharith A.Abdullah
Title: Asst. Prof
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Israa Hadi Ali
Title: Prof
Date: / / 2022
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: **Dr. Hussein Atiya Lafta**
Title: **Prof**
Date: / / 2022
(**Dean of College of Information Technology**)

# Dedication

To

My father, although he was my inspiration to
pursue my doctoral degree, he was unable to see
my graduation. This is for his soul.

To

My family
My Mother, brother,
My dear wife,
My darling sons,
My lovely daughters

# Acknowledgement

First of all, I have to express all thanks and gratitude to my Allah who gives me the ability to achieve such this work and to his prophet Mohammed. I would like to express my sincere gratitude and appreciation to my supervisor (Prof. Dr. Israa Hadi Ali) for her invaluable guidance, supervision, and untiring efforts during my study. Special thanks go to all the staff members of Information Technology College for their faithful efforts to give us the utmost scientific topics and endless support in all directions aiming to bring into scientific perfection, and for their unforgettable kind and wise management of our affairs during the research period. I am immensely grateful to all my family members, mother, brothers, sisters, my wife and children for their patience and encouragement who have been my greatest impetus that enabled me to complete my research. Finally, I apologize to those whose names might be missed. But I am grateful to all of them for the help that I have received from in various research and day-to-day life matters.

# Abstract

The surveillance camera systems field is considered a significant area of research in recent years because it is used for various purposes like monitoring big buildings, streets, interesting places, etc. These systems deal with varying and difficult scenarios in detecting and tracking moving objects. The monitoring systems can intelligently analyze content of the video and discover unusual behaviors. These technologies can provide more accurate and safer surveillance through the support of police and security in cases of investigation. Prediction of the direction of object in the scene and across the camera's network is still a challenge because the object changing its appearance, pose, or occluded. Disappearing the target(person) from the scene causes degradation in the performance of the surveillance camera system because the system loses track. Re-identify the same target in the camera's network that needs to search for the target in all neighbor’s cameras will need more processing and more consumed time. A solution to this problem is increasing the association between states to re-identify the same target after occlusions and predicting the correct direction of the target to assign the specific next camera where the target may reappear.

The proposed system in this dissertation is tracking and predicting the direction of a moving target combines the techniques of deep learning with spatial information from the camera network topology. The proposed system consists of many stages: firstly, split video into sequence frames and features extraction using YOLOv3 algorithm for target detection. Secondly, tracking the target using Kalman filter and deep sort then constructed trajectory points of moving target. Thirdly prediction the direction of moving target using proposed algorithm with LSTM model to associate important features for a target across sequential frames. Finally, using person re-identification model to re-identify same target across adjacent cameras depends on target features, the result of prediction and spatial with temporal information from the cameras network topology to detect where the target may reappear.

Two datasets adopted in this dissertation are the “Multi-Camera Track Auto (MTA) video dataset” which is provided by the Computer Vision Foundation and Abbasid shrine in Karbala city dataset. The MTA dataset contains six cameras(indoor and outdoor) distributed in overlapping and non-overlapping fields of view; The second dataset contains six cameras with nonoverlapping distribution .Two types of datasets contains a number of videos with different scenarios of moving persons under various conditions of illumination. Finally, the evaluation results of prediction and re- identification models for MTA are 0.89 and 0.90 respectively and for Abbasid shrine datasets are 0.91 and 0.86 respectively. The proposed system achieves accuracy reaches 90%.

# Declaration Associated with this Dissertation

Some of the works presented in this Dissertation have been published or accepted as listed below.

1. Wael M. Brich, Prof. Dr. Israa H. Ali, " A review for object detection and tracking methods in visual surveillance system of cameras network " Webology, Volume 18, Number 4, pages 164-176, (2021).

2. Brich, W.M. & Ali, I. H, "Object Prediction in Surveillance Cameras Network Using (LSTM) ". Webology (Vol. 18, No.4, p:645-657), (2021).

3-Brich, W.M. & Ali, I. H, “Predicting Motion Direction and Person re-identification across surveillance Cameras Network using (LSTM)”, IEEE ,(2022).

# Table of Contents

## CHPTER ONE GENERAL INTRODUCTION

## CHAPTER TWO THEORETICA FUNDAMENTAL

## CHAPTER THREE THE PROPOSED SYSTEM

## CHAPTER FOUR THE EXPERIMENTAL RESULTS AND DISCUSSION

CHAPTER FIVE CONCLUSIONS AND RECOMMENDATIONS

# List of Tables

# List of Figures

# List of Algorithms

# List of Abbreviations

| Abbreviation | Meaning |
|---|---|
| **ADEM** | Adaptive Moment Estimation |
| **AF** | Activation Function |
| **ANN** | Artificial Neural Network |
| **BRNN** | Bidirectional recurrent neural networks |
| **CNN** | Convolution Neural Network |
| **CV** | Computer vision |
| **DNN** | Deep Neural Networks |
| **FOV** | Field of View |
| **GRU** | Gated Recurrent Units |
| **HOG** | Histogram of Oriented Gradients |
| **IOU** | Intersection Over Union |
| **KF** | Kalman Filter |
| **LSTM** | Long Short-Term Memory |
| **MOT** | Multiple Objects Tracking |
| **MSE** | Mean Square Error |
| **REID** | Re-identification |
| **RELU** | Rectified Linear Unit |
| **RNN** | Recurrent Neural Network |
| **SCNS** | Surveillance Cameras Network System |
| **SIFT** | Scale-Invariant Feature Transform |
| **SVM** | Support Vector Machine |
| **SORT** | Simple Online and Real-Time Tracking |
| **SURF** | Speeded Up Robust Features |
| **SGD** | Stochastic Gradient Descent |
| **YOLO** | You Only Look Once |

# CHAPTER ONE

# GENERAL INTRODUCTION

## 1.1 Overview

Computer Vision (CV) applications are developed and expanded over the last decade in various ways. Different algorithms have been proposed in the field of computer vision, which enhanced the level of performance for many purposes such as visual systems. As a result, the development in camera types and their low manufacturing cost helped to increase the demand for video surveillance systems(VSS) in many applications as camera networks are installed for monitoring in banks, train stations, and public places. The main idea behind the system of cameras is to analyze events and solve various types of troubles in processing of video like object detection, and object tracking and keep the same identification (id) for a unique target for a long time across the camera's network [1].

Object tracking means how to keep the identity of moving objects across a sequence of frames. There are many different general algorithms that can be applied for tracking tasks as a particle filter, optical flow, and Kalman filter, recently tracking process used an algorithm of the deep sort. Pedestrians are a significant subject in the field of video tracking therefore many studies focus on human behavior in the real environment and how can simulate and present these scenarios in machines [2].

Object movement prediction in video remains a complex task in the field of computer vision. Prediction the next positions (next points of trajectory) of moving target in video remains a complex process such as traffic surveillance, person re-identification (RE-ID), and intelligent monitoring [3]. Enhancing the tracking performance on a wide- area is need select an appropriate way that can solve the sending and receiving problems between cameras in the network. In surveillance scenarios, a number of cameras are distributed on a large area to covering interest places for clear monitoring. Economically, the distribution with nonoverlapping fields of view in the network is preferred and usually adopted in different fields [4]. Understanding the motion behavior for a moving person represented by a long sequence of frames over time, required the choice of an appropriate network for this objective. Recurrent Neural Networks (RNNs) are a public and predictable model that has a design comprising a “memory” which is used to save the previous information, then used this information with its history to predict new states such as video processing or language processing. The major aim of RNNs is to use sequential information to associate states based on previous information to predict new states, the information from previous frames can help in understanding the events in the current frame. A recurrent term in RNN means the network achieves the same task for each element of a sequence, and the output result depends on the calculations in previous steps. The gradient exploding and vanishing gradient problem are the main disadvantages of RNNs, it makes the training model of the RNN is difficult in several ways. The structure of long short-term memory LSTM network is an update of RNN and is used to solve the disadvantage of RNN by overwhelming the vanishing gradient problem. The model of LSTM is considered a special type of RNN that works for many tasks, better than the traditional structure [5][6].

## 1.2 Video Surveillance Systems

In video surveillance, the first step is building the cameras network topology by assigning the area such as the places and roads which require monitoring, cameras need to be installed at vital positions (suitable field of view), like significant intersections, different entrances of buildings and crosswalks for pedestrians. These places are characterized by better surveillance of pedestrian information. Creating a topology of multi-camera network for these significant places can often provide a level of security with a more accurate surveillance effect. Based on the topology of network, the spatial location, and the field of view of camera are combined to perform a fast and accurate pedestrian detection and tracking and produce interested information about complete track of pedestrian [7]**.** Detection, tracking, and understanding the moving objects behavior is important in real world scenes have been active research fields in CV for many years ago. Surveillance cameras network system (SCNS) refers to an automated visual monitoring process that require interpretation and analysis the visual events in video to understand the behavior of object in scene. Basic tasks of SCNS comprise on-scene interpretation and wide area surveillance control. Scene interpretation aims at detecting and tracking moving objects in a sequence of frames and analyzing their behaviors. In a wide area surveillance control task, multiple cameras or agents are controlled in a cooperative manner to monitor labeled objects in motion [8].

## 1.3 Related Work

Recently, many researchers have worked on surveillance cameras network systems based on a broad range of techniques which deal with object detection, tracking , motion prediction and reidentify moving object across cameras. The summary of these works can be presented in Table (1.1). This section, a review of techniques is accomplished for SCNS:

### 1.3.1 Object Detection and Tracking (YOLO and Deep Sort)

There are many researches using yolo and deep-sort algorithms for detection and tracking single and multi-objects as:

**Host Miran Pobar, Marina Ivašić-Kos and Kristina** ,2020 [9]were suggested using YOLOv3 object detector with the algorithm of deep sort for player tracking. The idea behind this research is to explain the scenario of detect and track process for all the players appearing on the handball court, and to analyze the particular athlete performance. This is an appropriate process and requiring of multiple objects tracking because the motion of players is fast, often the player may be changing his direction, and may be occluded or leave the view boundary for camera.

**Tuan Linh Dang , Gia Tuyen Nguyen and Thang Cao,**2020[10]were suggested a model for tracking moving object. This model is the enhanced version of YOLOv3 with the deep sort algorithm. The proposed model also includes the correlation tracker of the Deep learning, which is used to reduce the identity switches and keeping the id for long time. In addition, the proposed model is created with a parallel approach to boost the operating speed. Experimental results with various videos present that the suggested model obtained higher operating speed and lower identity switches compared with the conventional deep sort yolov3 approach.

**Yang Jie and Lilian Asimwe Leonidas**, 2021 [11] this study shows a method for the detection and tracking of ships based on two algorithms one for detection named You Only Look Once (YOLOv3) and the second new algorithm of tracking named Deep Simple Online and Real-time Tracking (Deep SORT). Three enhancements are made to the detection algorithm (YOLOv3), Firstly, using the clustering algorithm(K-means) to enhance the starting values of the estimated anchor box for the frame to make it much more appropriate for track of ship scenarios. Secondly, the classifier output is adaptive to a single SoftMax classifier to suit the ship dataset which has three types of ships. lastly, Soft Non-Maximum Suppression is applied to overwhelm the limitations of the Non-Maximum Suppression when screening candidate frames.

### 1.3.2 Prediction of Object Direction (RNN and LSTM)

There are many researches using RNN and LSTM algorithms for predicting a single and multi-object as:

**Manh Huynh Trung and Gita Alaghband**, 2018 [12]. Two coupled LSTM networks, Pedestrian movement LSTMs (one per target) and the corresponding Scene-LSTMs (one per grid-cell) are trained simultaneously to predict the next movements. It shows that such common path information greatly influences prediction of future movement. Their further design a scene data filter that holds important non-linear movement information. The scene data filter allows us to select the relevant parts of the information from the grid cells' memory relative to a target's state.

**Wanli Ouyang and Pu Zhang**,2019 [13]. Recently many studies are using LSTM models because the model presented an excellent capability to understand and learn social behaviors. Therefore, a number of these approaches ignore the important current intention of the neighbors and depend on previous neighboring hidden states to suggest a structure of an LSTM network to solve the problem of linked predicted trajectory points for moving persons in the crowded space. The method deals with LSTM as an extractor of feature to refinement module, the current features of all persons will refine adaptively based on the mechanism of message passing. In addition, can define social-aware information selecting system comprising a movement entrance and which interest of a pedestrian to select appropriate features of each neighbor.

**Ehtesham Hassan**, 2020 [14] the proposed motion representation and deep features representing objects appearances are fused in an embedded space learned by the hierarchical LSTM structure for predicting the object to track association. The authors present experimental validation of the proposed approach on multiple objects tracking challenge datasets and demonstrate that their solution naturally deals with major tracking challenges under all uncertainties.

**Xurshedjon Farhodov and Kwang-Seok Moon**,2020 [15] were used the LSTM Network-based Tracking association method for tracking multiple objects over frames in real-time by building one of the efficient LSTM models that are used with tracking, this way supports the tracking process in long-term with solving limitations. In their approach, the main idea focuses on the learning trackable objects' motion, locations and appearance, the next step is predicting the new position of objects (new coordinates of bounding boxes) with respect to their starting position. The evaluation results of the association object tracking system has presented that the LSTM model is much robust and capable of working on a real-time multi-object tracking task.

### 1.3.3 Re-identify Moving Object Across Cameras Network

There are many researches using re-identification model for re-identify a single and multi-object as:

**Yeong-Jun Cho and Jae-Han Park**, 2017 [16] were proposed a system based on less previous knowledge about the nature of environments to solve both cases re-identification of person with topology of camera network problems. It considers general multi-camera network environments and is applied to person re-identification in a wide area. Dataset with all annotations used for new person re-identification, called SLP, captured from nine cameras distributed as non-overlapping view. Evaluation results using public datasets with their person re-identification to present that the suggested approaches are capable for solve the limitations related with both camera topology inference tasks and person re-identification.

**F´abia Isabella Pires Enembreck1 and Erikson Freitas de Morais**,2020 [17]the aim of research was built two robust models depending on deep learning methods for person re-identification. The first model uses a Siamese neural network comprised of two identical subnets. Two images as input for model that may or may not be from the same person. The second model includes three identical subnets based on a triplet neural network, which receives the image as a reference from a specific person. The second image represents the same person but in different scenario and the last image is from a dissimilar person. Two models used the same subnets, created by a convolutional neural network(CNN) which extracts general features from each image and an autoencoder model.

**Mianjin Wei and Jihong Pei** ,2020 [18]were suggested (use yolo-v3 for target detection, and deep sort to correlate targets) of tracking for multi persons using deep learning methods with non-overlapping distribution for multi-cameras. The tracking process in this method is classified into three stages: firstly, using deep learning to extracting structured information (ps = [$c_s$, $z_s^{in}$ , $z_s^{out}$ , $t_s^{in}$ , $t_s^{out}$, $k_s$]) from single camera where (c) indicating the camera number, $z_s^{in}$ and $z_s^{out}$ indicating the position of the pedestrian in the camera's field of view, $t_s^{in}$ , $t_s^{out}$ the time of entering and leaving the camera; secondly, creating topology of camera network; finally, the candidate pedestrian can be extracted in neighbor's cameras by structured information and topology of camera network, then using person re-identification method.

Table (1.1) latest related works and their datasets

| Technique | Dataset | Limitations | Evaluation |
|---|---|---|---|

| (year) | | | |
|---|---|---|---|
| YOLO and Deep Sort, 2020 [9]. | The dataset contains a subset of high-quality video recordings of handball matches using indoors view during a handball school. Using a stationary camera Nikon D7500 DSLR, with a Nikon 18-200mm VR lens, in full HD resolution (1920x1080) at 60 frames per second the total duration about 6min and 18s of the annotated datasets . 10- 11 players appear in most video frames. | Due to the relatively large number of players in the video, frequently changed positions, and occlusion, a large number of identity switchers are present(1483). | The results show that for each player that should be tracked, the identity switches caused the creation of, on average, 5-6 additional tracks by the Deep SORT algorithm, so there are 5 times more tracks than in ground-truth data, IDF1 24.7%, IDP 24.7% IDR 24.7%. |
| YOLO and Deep Sort, 2020 [10]. | The dataset used to test the proposed deep sort yolov3 architecture contains video filmed at Akihabara in Tokyo, Japan. To test various situations, two different videos were tested. Many people appear in the first video while in the second video the people were fewer. Each video was a 26-second video, input frame was 640×480, IoU threshold = 0.64. | This approach requires much time for the tracking process. Also, the efficiency of the Deep SORT algorithm is based on the results of the YOLO process. If the YOLO process cannot detect any bounding box of an object, the Deep SORT algorithm cannot track this object. | The Deep SORT- Dlib architecture with the uncrowded video had 81 objects that had identity switches in 6709 detected objects. Compared with the uncrowded video, the number of detected objects and the number of switched increased in the crowded video. However, in both situations, the DeepSORT-Dlib approach obtained lower identity switches. The results 81/6709 (1.21%) 324/10298 (3.15%). |

| | | | |
|---|---|---|---|
| YOLO and Deep Sort, 2021 [11] | The dataset was collected from the Yangtze River which is found in Wuhan, China. Yangtze River is a shipping canal for many domestic ships. There are various video and images were shot on both sides of the river to build this dataset, the size of image input is 608 × 608, threshold of IOU is 0.45, and target score threshold of 0.4. | They cannot track the ship entering the picture in the middle of the video. | When Ot equal to 0.1 and the image size is 480, the mAP reaches 0.909, but the FPS is lower.t. When Ot is equal to 0.1 and the image size is 480 × 480, the mAP reaches 0.955, which is about 0.05 higher than the highest value before the improvement. |
| LSTM,2018 [12] | Evaluate the model on two publicly available datasets: walking pedestrians dataset provided by ETH Zurich (ETH) [1] and crowd data provided by University of Cyprus (UCY). These datasets contain 5 video sequences (ETH-Hotel, ETH-Univ, UCY-Univ, ZARA-01, and ZARA-02) consisting of a total of 1536 pedestrians with different movement patterns. | The limitations were in a given grid-cell, there can be multiple human trajectories possible (dissimilar walking patterns: different directions, velocities, and degrees of non-linearity). | The results show that the method reduces the location displacement errors compared to related methods and specifically about 80% reduction compared to social interaction methods. |
| LSTM, 2019 [13] | These two datasets contain five crowd sets, consist of, UCY-zara02, UCY-zara01, UCY-univ, ETH-hotel and ETH-univ. There are 1536 pedestrians in total with thousands of non-linear trajectories used to test the model on these 5 datasets. | Sudden change in pedestrian movement will effect in predicting the trajectory of pedestrian in scene. | There are two types of metrics for evaluating the performance of trajectory prediction, including the Mean Average Displacement (MAD) error and Final Average Displacement (FAD). The |

| | | | |
|---|---|---|---|
| | There are two kinds of measurement for evaluating the prediction of performance, comprising Final Average Displacement (FAD) and the Mean Average Displacement (MAD) error. | | result for two metric was 0.45/0.94 in meter unit. |
| LSTM, 2020 [14] | The proposed algorithm has been evaluated on MOT15 and MOT16 datasets available. MOT15 contains 11 training and 11 testing sequences; MOT16 contains seven training and seven testing sequence. | Occlusion and illumination changes, which cause loss of targets; ID switches (IDS) as well as re-entry of targets in the scene. | Tracking result on MOT15 dataset. MOTP was 72.6% and the result on MOT16 dataset. MOTP was 76.2%. |
| LSTM, 2020 [15] | This approach, using datasets KITTI tracking and MOT16, these datasets are containing out-door and in-door videos from the environment that includes single and multiple object scenes of vehicles and human. Training data around 13K images which means in the KITTI-tracking a set has 20 video sequences , around 8K images; in the MOT16 training data 7 video sequences around 5.3K images within has been trained. | A number of attempts to figure out multi-object tracking most common limitations, such as, similar appearance, frequent occlusion, motion of several objects the same time, edge problems realized that adaption of deep learning approach. | The method (LSTMNET_TA) performance on KITTI-tracking dataset was MOTA=88.08% and MOTP=89.22%. |
| Re-identification model,2017 [16] | using public datasets NLPR MCT, it consists of four subsets with multi-camera networks have non-overlapping | The pose of a camera can change due to vibration, heavy wind, and so on. Even a slight change in | To evaluate the accuracy of the camera network topology, we adopt two evaluation metrics: |

| | | | |
|---|---|---|---|
| | distribution and different many people. Among them, they select two datasets ( DATA1, DATA2) comprises a lot of people. Each dataset has 20 min Duration. Since the datasets are not large-scale datasets (small number of people), we used all videos (20min). | pose can cause a considerable viewpoint change. | transition time error and topology distance. The accuracy of the system was 87.5%. |
| Re-identification model,2020 [17] | Three public datasets were used: VIPeR, i-LIDSVID and CUHK03. for network training and testing. It contains a total of 1264 images of 632 pedestrians, and for each pedestrian there are two images captured by different cameras, with changes in the lighting and poses of persons. The i-LIDS Video Re-Identification dataset comprises images of 300 different pedestrians obtained through two non-overlapping cameras. The CUHK03 dataset was contains about 13164 images of 1360 different persons , from six cameras. | The limitations, such as, similar appearance, frequent occlusion, motion of several objects the same time | The results showed that the use of AE generated a significant gain, both for the Neural Siamese Network and the Neural Triplet Network. With the AE, the accuracy rates increased in all cases, as well as the CMC Curve, and it was found that the AE provided a gain of up to 71.05 %. |
| Re-identification model,2020 [18] | From the path of the lawn around the Science and Technology Building of Shenzhen University record this dataset , contains on four cameras the identities in dataset | The method of this paper also has a disadvantage that after matching the wrong pedestrian under the adjacent camera, all the track information will be | At the speed of recognition, each of our pedestrians needs to match from all the pedestrians under the camera, and after the pre-sequence |

| | | | |
|---|---|---|---|
| | about 60, every person image is resized to 128*256 . | wrong afterwards. This depends on the performance of person's feature descriptor and the position of the camera. | extraction, the number of comprehensive queries is reduced from 2046 to 765 and trajectory accuracy was 73.1% |

## 1.4 Problem Statement

Object occluded in (scene) or across nonoverlapping cameras network ( blind area) is difficult task because the target is disappeared and that effect on performance of surveillance system. May be lost the identity of track by id switch or missing detection. When object leave scene, the camera will send message with the information of target for all the neighbors' cameras and each camera must do matching with all objects for checking if target is reappeared or no. This operation will be causes more processing and consume time in neighbors' cameras.

## 1.5 Motivations

Based on digital world technology evolution, there are many motivations points of this research that should be listed in short as follows:-

- Pedestrians are considering a significant matter in field of video tracking so; several studies focus on behavior of human in real environment and how can simulate and present this performance in machine.
- This proposed system support surveillance cameras systems to detect and track the target with efficiency across network by improving the performance of monitoring system depending on association of states for moving target across short and long-term occlusions.

## 1.6 The Aims of Dissertation

The aim of the dissertation can be listed as follows:-

- Building prediction model for predicting the direction of the moving person in scene to tracking and monitoring the target across cameras network.
- Predicting the direction of moving target (across blind area) and candidate a next camera then re-identifies where the target may reappear in cameras network that will reduce processing cost and consumed time in neighbor cameras(static camera).
- Using a nonoverlapping camera distribution in a wide area instead of overlapping distribution which has more cost.

## 1.7 Challenges

The challenges for moving objects in scene can be listed as follows:

- Prediction the direction of moving target in scene still challenges like occlusion, pose, a change in appearance and background clutters.
- Re-identification the same moving target across cameras network (blind area) is difficult and complex process.
- Finding an appropriate dataset that covers a large area to apply the concepts and operations on it, stay a difficult task because of the specialty law.

## 1.8 Limitations

The work limitations can be listed as follows:

- The distance between non-overlapping cameras (blind area) and the places geometrical.

- Using static cameras in topology of network.
- Crowded places and moving multi-objects in scene.

## 1.9 The Contributions of the Dissertation

The Contributions of the dissertation can be listed as follows:

- Proposing method for predicting the next position and direction of the moving target depending on change in its angle of direction and coordinates.
- Covering (blind) area between non- overlapping field of view FOVs depends on prediction (at any place the target may reappear) by association the states of moving target between disjoint cameras(blind area) using LSTM model.
- Propose efficient technique using deep learning (LSTM, YOLO and Deep sort) algorithms with ( Trajectory features , learning features and place geometrical) for prediction at any camera the target may be appear.
- Create re-identification model using convolution neural network with previous features vector of target which is received from previous camera to identify target across cameras network.

## 1.10 Dissertation Organization

In additional to chapter one this dissertation contents four chapters. Each chapter begins with an overview and ends with a short summary that offers an impression of that chapter.

1. Chapter Two provides an extensive and comprehensive theoretical fundamental of the used techniques to accomplish the proposed prediction system. It represents the objects' detecting, tracking, identification methods, RNN and LSTM concept for best results, camera network types, data (video), discusses and identifies the detection, tracking, prediction.

2. Chapter Three explains the proposed prediction system stages by discussing and identifying each stage of the system using LSTM with trajectories features and re-identify the moving target with the experimental study setting and execution.
3. Chapter Four explains a comprehensive analysis and discussion of the obtained results for target detecting, tracking, predicting direction of moving target, re-identify target across cameras network and system test.
4. Chapter Five concludes the research findings, recommendations, and future work.

# CHAPTER TWO

# THEORETICAL FUNDAMENTAL

## 2.1 Introduction

In this chapter, a theoretical fundamental is stated and described for monitoring cameras network system. It explains the suitable techniques that are used in the object(person) detection as the foreground objects extraction method, tracking methods as the deep sort (Hungarian algorithm and Cascade algorithm) for association target states and suitable metrics such as intersection over union with similarity distance. Describe deep learning techniques as artificial neural network(ANN) with its kinds as the convolution neural network(CNN), (RNN) and (LSTM), which are used for remembering previous information for complete prediction tasks. Explaining for the moving target across a multi-camera network (overlapping and non-overlapping) with the re-identification process as well as the dataset used in the dissertation.

## 2.2 The Cameras Network

The computer vision with sensor network as well as the steps of development in communication systems are contributed to increase the demanding to expand for installing different kinds of camera networks. Visual monitoring of the Camera network is an active and vital field in our life. There are many applications needed using the video camera network, like security, environment monitoring. It is essential to develop techniques for analyzing the data collected from cameras automatically, processing and summarizing the results in a way that is meaningful to the end user. Depending on some factors as area, cost, the aim of service and based on the topology of the network can be classified wireless camera sensor networks into two kinds centralized or distributed networks where the last kind can be nonoverlapping (blind area between cameras) or overlapping (interference cameras

FOV). The Spatial-temporal relations, zones and the type of connections between the cameras in the network can reflect the topology of a network [19].

The inter-camera transition distributions, indicate the movement of objects in a scene across cameras over time and show the connectivity strength between the cameras. In network topology, two cameras can be adjacent if an object can move between them without passing through any other cameras. The inter-camera transition distribution of objects between adjacent cameras usually, can be referenced from the camera network topology [20].

The camera's network can cover a large area as towns, airports, .etc. Monitoring and analyzing data for the flow of traffic can be a basic task in wide-area surveillance. In the non-overlapping field of views kind, there exists an unseen area in the camera network called (a blind area). In a scenario, it is critical to understand where an object leaving the view boundary of a specific camera is likely to appear next [21].

TCP/IP is a protocol used for communication between cameras nodes. This way of communication is performed in a separate thread in parallel and shared memory with frame processing. A unique ID assigns to every camera node and maintains tables as a list containing the IDs with information of spatial and temporal relations for its camera's neighbors. The topology of the network means each camera has a number of cameras neighbors located on its east, west, north or south neighbor depending on location of these cameras, distribution can be added or removed or changed the location of any camera in the system. For the identification of objects across several cameras, each camera in the network will need to receive the features vector from the previous camera as well as send the required information to the next camera. In other words, a camera node is work as a receiver and sender at the same time[22].

More recently, multi-camera monitoring systems can exchange the interesting information between them using a P2P manner for communicating, which means each camera in network can have the ability to detect and track objects as well as processing power by itself. All nodes can work together to give the solutions for the association problems, by exchanging object features, and retrieving the correct positions of the occluded objects. Thus, when the data that need to be exchanged between cameras can be reduced significantly. Also, each camera has the ability to initiate a request, give a reply, and make its own decisions. This eliminates the need for a central server and reduces the required communication bandwidth. Therefore, this type of camera network system is usually able of execution object tracking task in specific time[23].

## 2.3 Surveillance Cameras Network System(SCNS)

Video surveillance systems are common and effective in various information technology software environments. Video surveillance was a necessary element to ensure security in banks, airports, monitoring of vehicular and pedestrian traffic, people counting and monitoring human activity to detect unnormal activity. Video surveillance cameras systems can provide monitoring coverage over a large area[18][21].

For a clear and ensuring visibility of object over a wide range of depths in scene and can be employed to disambiguate occlusion. Interest Events are recognized as a moving target, or any object that needs to coordinate in the multi-view system and must tracked the important events across the scene. With the availability and increased bandwidth in wireless network, also spread of cheap cameras, the deployment of a large number of security surveillance cameras is economically and technically reasonable [21].

As the size of SCNS expands, the object tracking task becomes gradually more complex. When a target moves in a scene the system tracking the target and when the target leaves a specific camera’s view and inter the blind area that means target disappearing, to re-identify the target that need will be search about the target in the views of the adjacent cameras that will need to time consuming. The topology of cameras and predicting the direction of moving target can significantly increase the effectiveness of system for tracking the target in a camera network[22].

Urban Video-surveillance architecture of large areas involves a network of multi-cameras with other tools as shown in Figure(2.1). One of the important tasks of the camera surveillance system is to keep the correct identity of objects moving across camera nodes in the environment. Many automated surveillance camera networks involve overlapping Fields of View (FOVs) for the tracking processes across multiple cameras. However, using overlapping distributed for cameras is always not possible especially when the size of the camera network is increased to cover wide areas[22]. Distributing more cameras to cover a wide area required more cost and consider a significant maintenance task since cameras can distribute with nonoverlapping topology to reduce the numbers of cameras in a network which means less cost for installation and maintenance. The new distribution for cameras network has some limitations as appearing the blind area between non-overlapping cameras. During tracking an object moving across cameras when the object leaves the boundary of a camera and disappears in a blind area with total occlusion, the monitoring system will lose the track of that object for a period of time[20]. The dissertation proposes a framework to improve the tracking task for moving target across cameras network by using prediction and re-identification models.

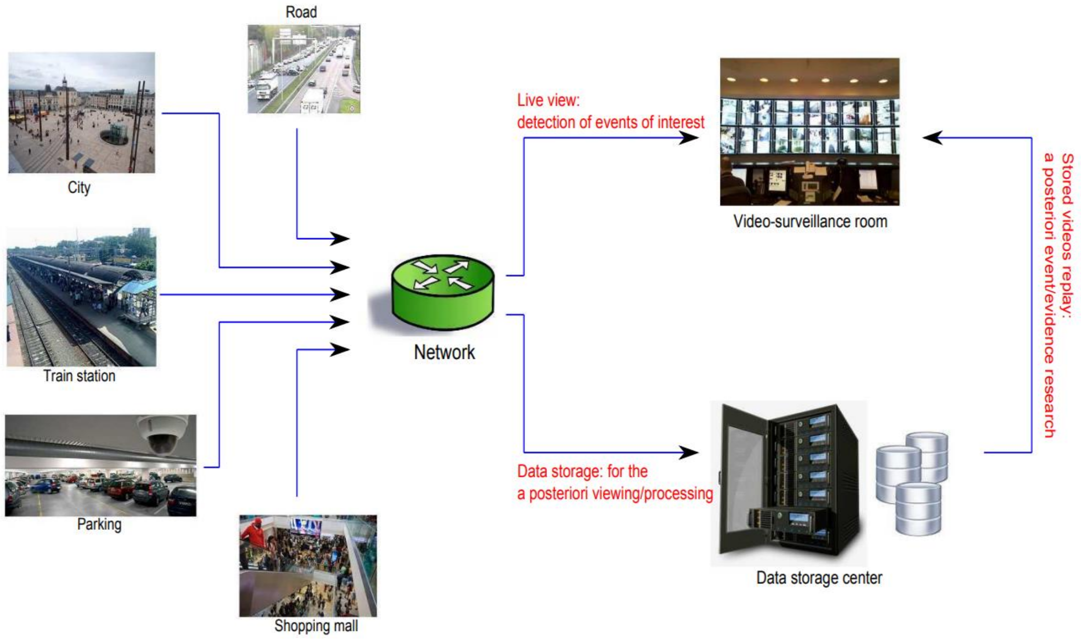


Figure 2.1: Video-surveillance architecture [4].

The monitoring systems store a huge volume of videos that means they apply compression process on video data, but most of methods for compression and decompression tasks have limitation such as a loss of information and some noise. These issues effect the performance of surveillance system which depend on gait and face recognition. In Addition, the biometric methods are dependent on the camera view, resolution and the orientation of the target which moving in scene. If the face was occluded or not visible (the person is seen from behind or from the side),that means the system is unable to perform the recognition of face . For these reasons, it is preferred to work on the whole body of object using appearance-based approaches, which have fewer limitations than biometric ones, and are more adapted to the video surveillance requirements [4].

## 2.4 Preprocessing of video frames

The pre-processing is needed for any type of data to refine for more processing due to poor captured quality. Cleaning of data is an important step and most the software engineers before building the model usually spend a good amount of time in data pre-processing task. Some examples of pre-processing of data comprises unwanted detection, treatments for missing value and eliminating the noisy data. The objective of pre-processing is an enhancing of the image data that removes unwanted deformations or improves some characteristics of image to be appropriate for further processing and analysis process. Image pre-processing may also decrease time of model training and increase model inference speed . If input images are large, reducing the size of these images will significantly enhance time of model training without dramatically reducing the performance of model. For example, fully connected layers in CNN required the same sized arrays for all these images [24][22].

Pre-processing of an image is considering the first step to image recognition and feature extraction. Regardless of what device is used for image acquisition, the input of images is always not satisfactory and requires improvement. For example, the image deformations mean the interest region in the image is not clear or may present clutter such as interference between multi objects in the image. Various pre-processing methods should be used for various applications of image. This method may comprise, orienting, resizing, and color corrections. Image filters can be classified for two kinds into linear or nonlinear. The first type is also known as convolution filters when represented by matrix multiplication, image equalization and thresholding are examples of nonlinear filters, as is the median filter[24].

### 2.4.1 Median Filter

The Median filter is belonging to a non-linear filters class, it is used to remove the noise from original image, filtering process begins with calculated the median value by organizing all the values of pixel from surrounding k × k neighborhood into any order descending or ascending and then replacing the pixel being considered with the middle pixel value. In simplify if a numerical list is having the media $\delta$ with the order as follows: half of the numerical list is greater than or equal to δ and the other half is less than or equal to δ, if the order δ in the list of numbers is even the median is the average of the two elements which are the two middle values after ordering, else the median is the value which is the middle value after ordering [22][23].

### 2.4.2 Gaussian Filter

The Gaussian Filter is an adapted version of the Mean Filter where the impulse function weights are spread normally around the origin, where the center of the filter represents the highest value of the weight and falls normally away from the center. When displaying this filter as an image, the highest value of intensity for pixels appears at the point of origin, and it distanced away from the value of center, this kind of filter is used to reduce unwanted distortion and noise by suppressing the components of high frequency. However, producing a blurred image after removing the high-frequency components of image, called Gaussian Blur as present in equation (2.1)[24].

$$G(x,y)=\frac{1}{2\pi\sigma^2}\,e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (2.1)$$

Where $\sigma$ refer to stander deviation and x , y represents the position[24].

## 2.5 Object Detection Techniques

Objects detected and identifying the class of object is an important stage, therefore selecting the suitable algorithm for completing this task is considered an important decision. Tracking by detection method depends significantly on the result of detection which means it required using a robust algorithm that gives acceptable detections under different conditions of the environment. Can classify object detection methods into two types called classical based on non-neural approaches and recently based on neural approaches. Non-neural techniques such as (Scale-invariant feature transform (SIFT), Viola-Jones object detection framework based on Haar features, Histogram of oriented gradients (HOG) features)[8]. These methods at first define features depending on the edge, line, and contour points, then use approaches like support vector machine (SVM) to perform the task of classification. In contrast, Neural network approaches can achieve end-to-end object detection without specifically defining features and are usually based on (CNN). Object detection based (ANN) can be classified into one and two stages methods as shown in Table(2.1) [25]:

Table (2.1): Represents one and two stages detection algorithms

| Important **one-stage** object detection algorithms | | Important **two-stages** object detection algorithms | |
|---|---|---|---|
| **Name of algorithm** | **Year** | **Name of algorithm** | **Year** |
| YOLO | 2016 | RCNN | 2014 |
| SSD | 2016 | SPPNet | 2014 |
| Retina Net | 2017 | Fast and Faster RCNN | 2015 |
| YOLOV2 | 2018 | Mask R-CNN | 2017 |
| YOLOV3 | 2019 | Pyramid Networks/FPN | 2017 |
| YOLOV4 | 2020 | G-RCNN | 2021 |
| YOLOVR | 2021 | -------- | ------ |

In two stages algorithm, the architecture involves a first step object region proposal by using suitable methods of computer vision or deep networks, followed by a second step when features are extracted from the proposed region with bounding-box regression to perform the classification process for the object. Two-stage methods achieve the highest accuracy of detection but are usually slower, because of the many steps apply per image to achieve the stages. The performance (frames per second) is not as good as one-stage detectors in maintaining accuracy, then will focus on this algorithm, and explain it in detail in the next section[26][27].

### 2.5.1 You Only Look Once(YOLO3) Algorithm

In the computer vision field, object detection is the most daunting challenge. Various algorithms are used to perform detection for any object in a scene, YOLO is a suitable algorithm of deep learning models designed for detection with good specifications as accuracy and fast. It is the most popular object detection algorithm based on CNNs. Various Versions of algorithms (from 1 to 3 versions) were created by (Ali Farhadi and Joseph Redmon). The advantage of YOLO is that feature extraction and object localization was unified into a single monolithic block. YOLO usually, depends on the single shot process when reading the interesting information from the whole image at the same time and it avoids the technique which uses region proposal or a sliding window. A single CNN can be used for both localizing and classification objects based on features extracted from bounding boxes, while other previous algorithms based on two tasks with different kinds of architectures to perform the detection process. When applying YOLO to the original image the algorithm will divide this image into an S x S grid. If the grid cell contains the center point of an object, therefore it is considered responsible for detecting process for that object such as in Figure(2.2)[28].

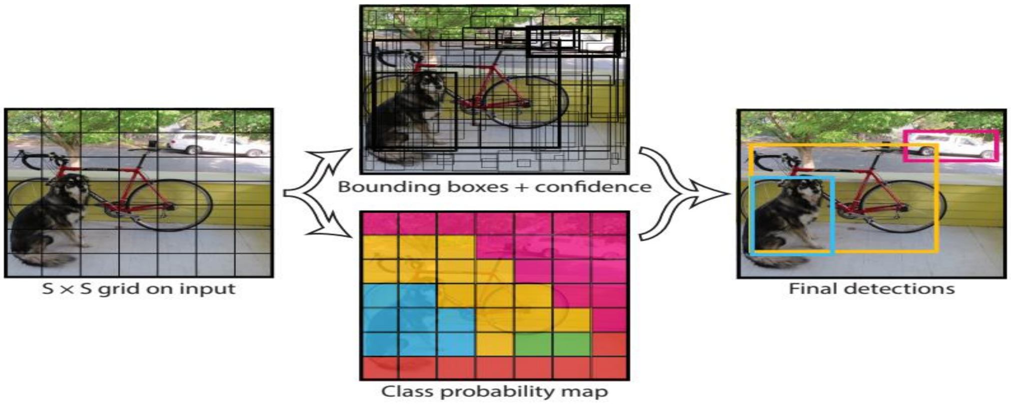
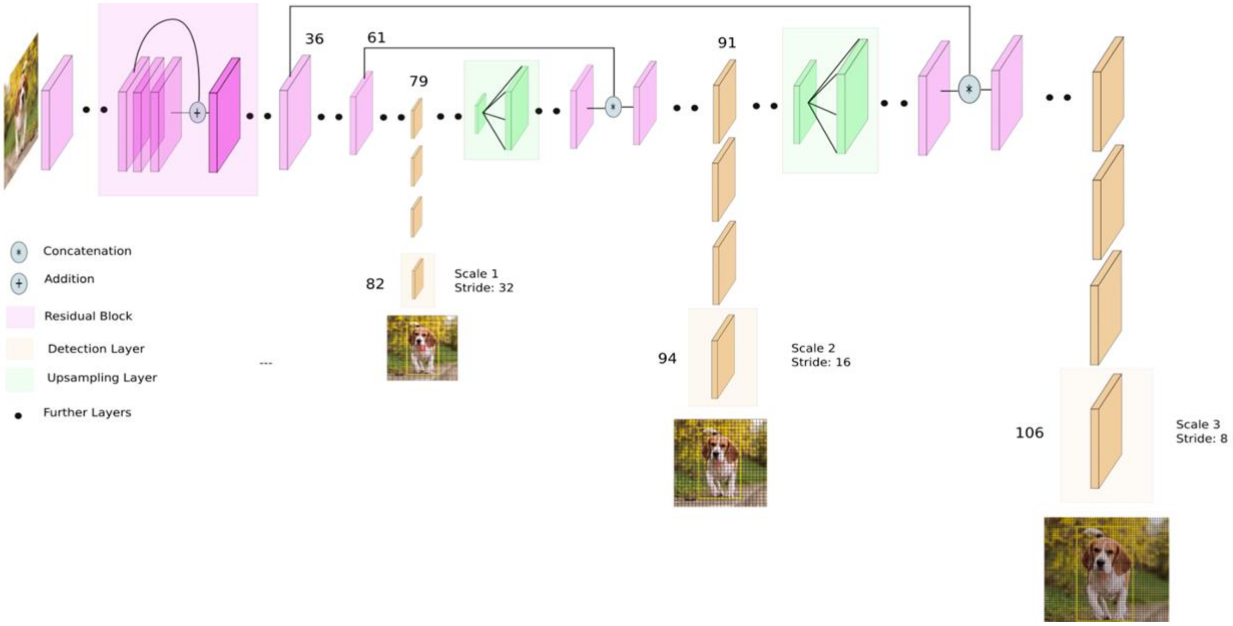


Figure (2.2): Steps of detection for many objects based on YOLO algorithm[28]

There are many versions of YOLO algorithm as presented in Table (2.1) each one of these algorithms have structure building based on some features such as balancing between speed and accuracy. Yolov3 was a suitable selection for detecting and classifying the target (person) in SCNS because that version of the algorithm has made some adaptive improvements as including different bounding box prediction, multi-label classification and multiscale recognition compared with the previous two versions. In addition, it is more robust but a little slower than its previous versions. The structure of YOLOv3 is created from improvement YOLOv2 as shown in Figure(2.3). The first step begins when an original image is input into the algorithm, the next step is to divide the image into S×S grids and check these grids to detect a bounding box to select the correct boxes which are full of real objects, followed by steps to classify all objects with a value of confidence score. Finally, can assign the values for location and confidence for each object candidate box and actually output on the image. The core idea behind this algorithm refers to using it only one convolutional network to estimate the classes and locations of the objects to perform more object detection. Where Object positioning, classification and recognition are considered regression problems [29].



Figure(2.3):Network Architecture Diagram of YOLOv3 [30]

The algorithm is adopting the structure of Darknet-53 with a deeper network layer and adds a residual module to the network to better extract object features. Darknet-53 also gets the maximum measured floating-point processes per second, which means this structure of the model better utilizes the graphics processing unit GPU, making it more effective to evaluate and fast processing. It has the same idea as the ResNet network, but it adopts using more residual modules in the network, where 1, 2, 8, 8, and 4 refer to the repeated number of residual modules, where each residual module has a residual layer and two convolution layers. The structure of the entire network has no pooling layer and can apply the down-sampling stage to reduce the size of the image by half as shown in table(2.2), by setting the convolution step size to (2). The steps that describe the structure of yolov3[30] can be listed as:

1. YOLOv3 depends on a structure of Darknet-53, which usually has a 53-layer network trained on ImageNet.
2. To perform the detection process, update the architecture for YOLOv3 by add new layers on 53 layers to produce a 106-layer fully convolutional layers.
3. In YOLOv3, to achieve the detection process applying kernel detection with size 1 x 1 on feature maps of three different scales at three different places in the network.
4. The Kernel shape is represented by vector as 1 x 1 x (B x (5 + 80)) which is used for detection object. Where parameter B refers to the total number of bounding boxes, and number 5 represents the five parameters (pc, bx, by, bh, bw) when pc refers to one object confidence, other parameters refer for bounding box attributes and number 80 is referred to a number of classes.
5. Binary cross-entropy is used in YOLOv3 for calculating the classification loss for each label, but class predictions and object confidence are predicted through logistic regression.

For explaining the steps of object detection is based on yolov3 with example illustrates how can the algorithm detects, localizes, and classifies the object in an image or in frames of video. The first step is the input batch of images with dimensions (416,416,3) for processing, then the YOLOv3 will pass this image to (CNN) using the Darknet-53 structure type. The image is divided into grids such as (13x13) cell then the algorithm, begins to predict the objects in an image. The first parameter (pc) of the vector (pc, bx, by, $b_h$, $b_w$ , c1………80 class) is to identify if an object has appeared in the grid or not (using a probability) if a cell does not contain any object that means (pc=0), bx, by, $b_h$ and $b_w$ represent the coordinates points for detected bounding boxes if there is an object. The parameter (c1 to 80 class) represents the number of classes which algorithm training to detect these classes. Consequently, if the object is a person, c1 will be (1) and other classes will be(0) , then, bx, by, $b_h$, and $b_w$ will be computed relative to this grid only. The parameters bx, by represent the coordinates of the midpoint of the object relative to this grid. The parameter $b_h$ is refer to the ratio of the bounding box height relative to the height of the grid cell and the parameter bw is refer to the ratio of the bounding box width relative to the grid cell width $p_{w,}$, $p_h$, $e^{t_w}, e^{t_h}$represent the output of NN*.* The range values for bx and by taking any number between 0 and 1 as the midpoint which usually lie within the grid. But $b_w$ *and* $b_h$ can take any value more than 1 in case the bounding box dimensions are more than the dimension of the grid, Figure(2.4)and the equations(2.2-2.5) show how calculate the parameters($b_x, b_y, b_w, b_h$)[30].

Table(2.2): Architecture of YOLOv3 [29].

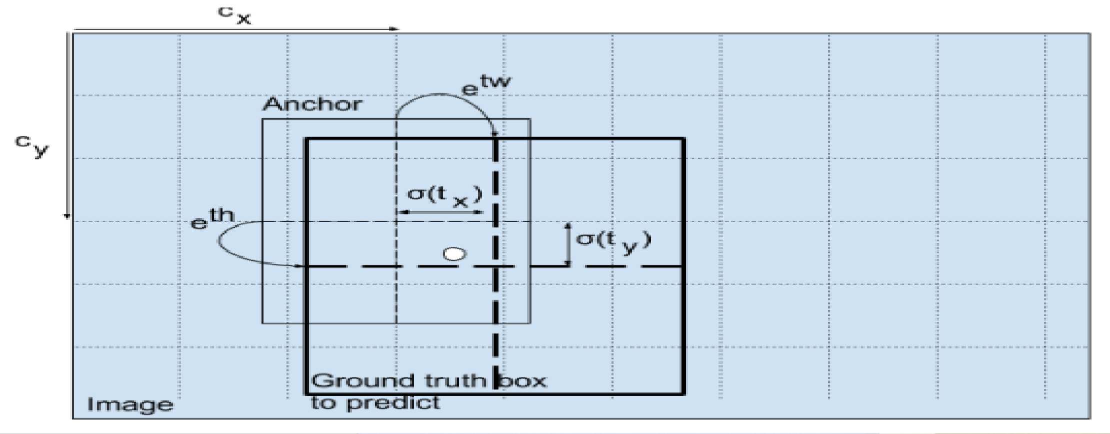
| **Layer** | **Filters** | **Size of filter** | **Repeat** | **Output size** |
|---|---|---|---|---|
| Image | | | | 416x416 |
| Conv | 32 | 3x3 | 1 | 416x416 |
| Conv | 64 | 3x3/2 | 1 | 208x208 |
| Conv<br>Conv<br>Residual | 32<br>64 | 1x1<br>3x3 | x 1 | 208x208<br>208x208<br>208x208 |
| Conv | 128 | 3x3/2 | 1 | 104x104 |
| Conv<br>Conv<br>Residual | 64<br>128 | 1x1<br>3x3 | x 2 | 104x104<br>104x104<br>104x104 |
| Conv | 256 | 3x3/2 | 1 | 52x52 |
| Conv<br>Conv<br>Residual | 128<br>256 | 1x1<br>3x3 | x 8 | 52x52<br>52x52<br>52x52 |
| Conv | 512 | 3x3/2 | 1 | 26x26 |
| Conv<br>Conv<br>Residual | 256<br>512 | 1x1<br>3x3 | x 8 | 26x26<br>26x26<br>26x26 |
| Conv | 1024 | 3x3/2 | 1 | 13x13 |
| Conv<br>Conv<br>Residual | 512<br>1024 | 1x1<br>3x3 | x 4 | 13x13<br>13x13<br>13x13 |
| Avgpool<br>Connected<br>SoftMax | | Global<br>1000 | | |

$b_x = \sigma(t_x) + C_x$ …………………………..………………………(2.2)

$b_y = \sigma(t_y) + C_y$ …………………………..…………………….(2.3)

$b_w = p_w e^{t_w}$ …………………………..……………….(2.4)

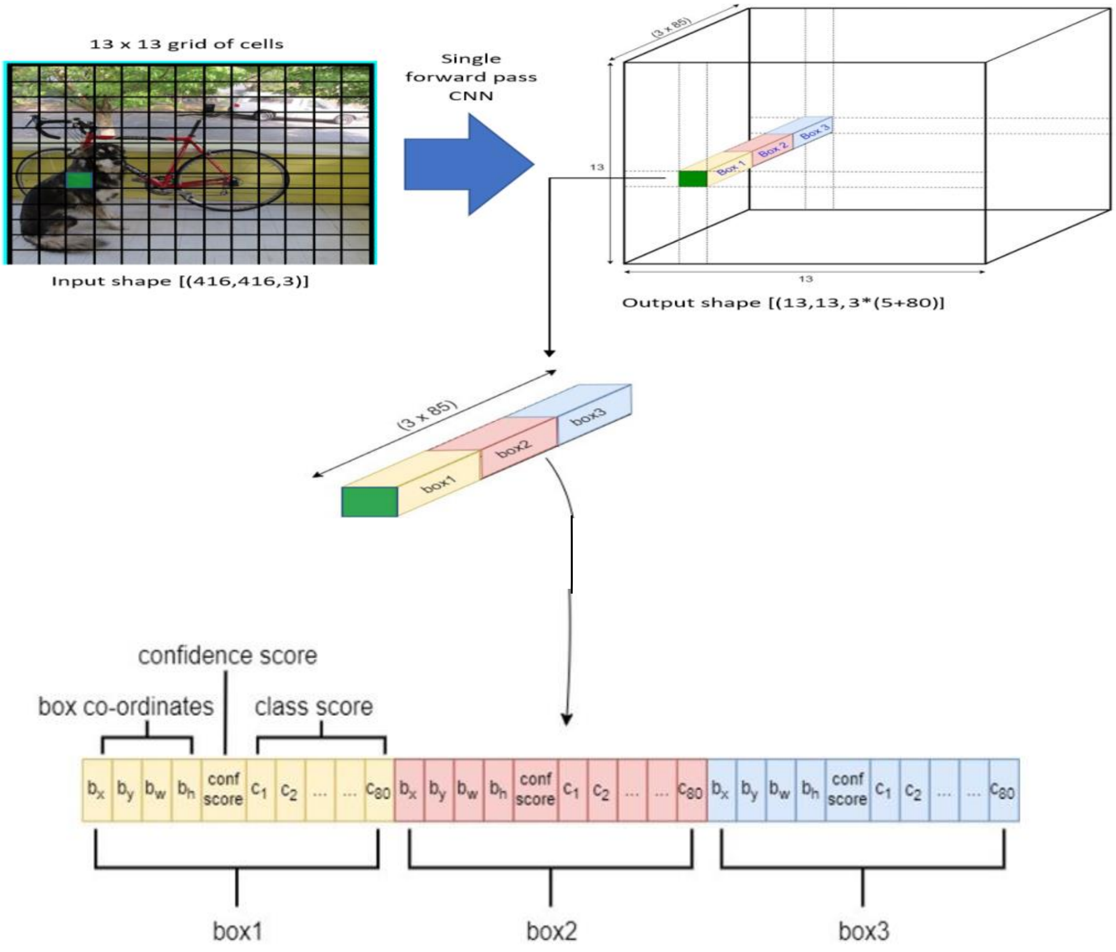
$b_h = p_h e^{t_h}$ …………………………..……………….(2.5)



Figure(2.4): Dimensions of bounding box[29]

A list of bounding boxes is detected as the output of YOLOv3 with the recognized classes would be each cell of a 13 x 13 grid returns 765 numbers, as a result, multiplying 9 anchor boxes by 85. Obviously, this process led to a lot of estimates, and many different objects in one image. Many detections will have a minimum probability of being actual objects[31].

The following step in the YOLOv3 algorithm is to apply two metrics the Intersection over Union (IOU) and the Non-Max Suppression to remove the overlapping boxes which have probability values less than a threshold as (0.5) that assign to these probabilities. The detection confidence result will be ignored if its value was below the threshold. Lastly, sometimes the same object appears in two or more bounding boxes after doing the prediction. To solve this problem, apply a non-max suppression algorithm to all the bounding boxes detected. This metric calculates the overlapping area by two bounding boxes that detect and contains the same object type. The lower probability bounding box will be deleted if the result value of IOU is higher than a certain threshold.Yolov3 can predict objects with three scales (big, medium, and small)[31] as presents in figure(2.5).


13 x 13 grid of cells
Single
forward pass
CNN
Input shape [(416,416,3)]
(3 x 85)
Box 1
Box 2
Box 3
13
13
Output shape [(13,13,3*(5+80)]
(3 x 85)
box1
box2
box3
confidence score
box co-ordinates
class score
bx by bw bh conf score c1 c2 ... ... c80
bx by bw bh conf score c1 c2 ... ... c80
bx by bw bh conf score c1 c2 ... ... c80
box1
box2
box3

Figure(2.5): How they can divide the input image into a number of cells encoding each bounding box[30]

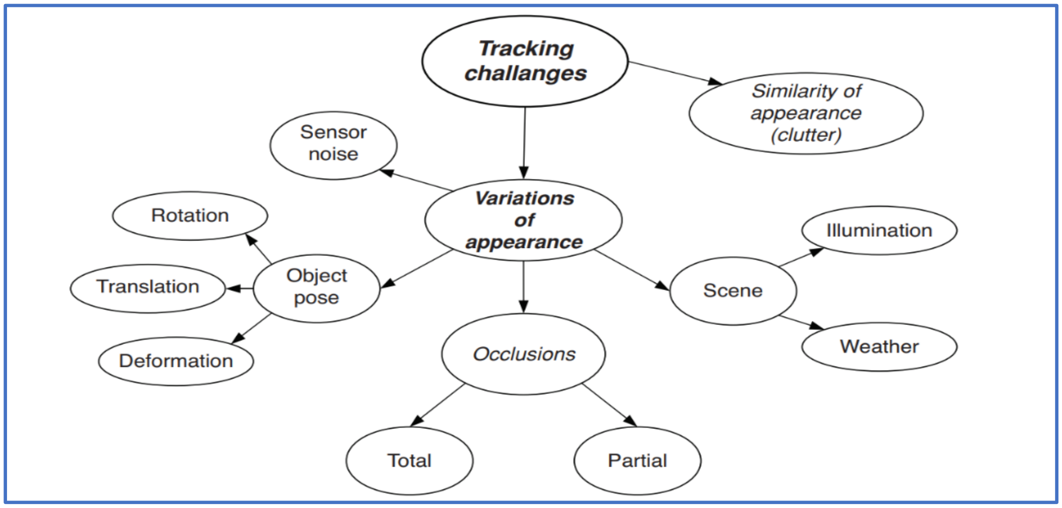
## 2.6 Object Tracking Techniques

Various applications in our life using object tracking in the field of computer vision, such as security depend on monitoring, traffic flow surveillance, and human activity recognition. The Object Tracking idea is how to identify an object surrounded by a bounding box and keep the unique object within the bounding box even if the object moves through the image plane or changes its size and shape[32].

Person tracking through video analysis and surveillance is still a challenging task, therefore, in recent years many approaches have been developed such as machine and deep learning. These approaches are depending on frontal view images/video sequences and on the improvement of CNN to develop the way of object tracking. The layers of (CNN) models are trained on a large number of video sequences or images to simulate the motion and other features of object behavior in the scene to enhance the accuracy and speed of object tracking[33].

The main challenges that must be considered when building and operating a tracker, can be the similarity degree of appearance features between the target and the other objects in the scene (clutter) such as in places of traffic, or the variations of appearance for the target itself. Video tracking is having many limitations such as the changes in the appearance of the target in the image plane which consider is one of the important challenges shown with one or more of the other factors as present in Figure(2.6)[34].

Occlusions are one of the important challenges in video tracking and can be classified as partially or totally occluded. The partially occluded happened when a part of the target body is occluded by a static object as a wall, column, or maybe other moving objects. The totally occluded means when the object disappears totally from the view may be for long period in a scene or when the object leaves the scene and enter the blind area and then reappear in another view[33][35].



Figure(2.6): The challenges of tracking[44]

### 2.6.1 Feature-Based Tracking Methods

This way is one of the suitable methods are used for the object tracking process. To track any object in a single or multi-objects state, many properties as optical flow, texture and color, will be extracted first. These features extracted in the previous step must be unique to recognize the objects with ease in the feature space. The features are extracted once, then the next step is using those features and based on some similarity criterion is to recognize where the more similar object in the next frame. One of the challenges with these methods is the feature extraction should be exact, unique and reliable features of the object so that it can recognize the target from other objects [36][37].

### 2.6.2 Segmentation Based Tracking Methods

Object Segmentation is an essential step and the most significant stage in video tracking. This step is applied to images or frames in sequence video to split the foreground for each object from the background scene. Generally, the foreground represents the area of moving objects, to track these moving objects, should segment each object from the background scene. Object tracking methods based on the segmentation can be joint-based or bottom-up-based methods[38].

### 2.6.3 Estimation Based Tracking Methods

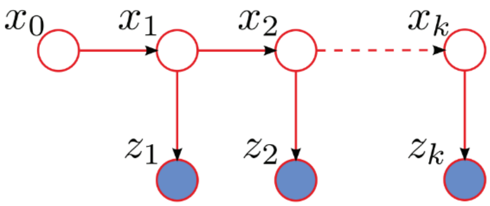
This method works recursively on data, the core idea is the estimation used the Bayesian methods for the dynamic mode estimation problem. When the object moves in the scene there are many states (state vectors) that can be describing the motion behavior of object such as velocity, direction and position [38]. The filters based on the Bayesian method allows the moving target in a scene to perform an update on its position (new coordinates) based on the latest sensor data.

The estimation algorithm uses two steps, a prediction step on the previous state then follows by updating step. With the recursive way, the first step uses the state model to estimate the new coordinates for the target position in the next step, while the updating step uses the observation model to update the position of the target depending on the current observation. The prediction steps follow by updating step when both steps are applied to every frame of the video. Particle and Kalman filter are two examples of this recursive method[39][37].

- **Kalman Filter**

The algorithm of Kalman was discovered at first in (1960) by Rudolph E. Kalman. It is one of the simplest models which deals with constant velocity and linear behavior for states, it is used broadly in many applications if the measurement models are linear functions. (KF) estimates the unknown variables of states space by taking a sequence of measurements that may be comprised of noise and other inaccuracies and estimates these variables. While the prediction of states performs recursively over time there is some uncertainty in measurement values. The Ambiguity may have appeared with measurement values representing the Gaussian in nature when the filter uses linear equation systems with white Gaussian noises as a standard model to perform the estimation process when this process includes noise and measurement noise contained in the measured values. KF estimates the optimal unknown state based on the recursively way which KF applies it to the stream of noisy input data. KF requires two sufficient inputs from a previous state with its current measures to estimate the current state using a recursive estimator for the process. These reasons allow using the filters in the tracking of robotics and other systems depending on real-time that required reliable information. Two operations can apply for object tracking with KF, the prediction and correction steps. The first stage of the estimation task is how can estimating the object states to obtain an a priori estimation of the next step depends on the current state projection. On other hand, the correction step in the prediction task will enhance the a priori estimation obtained in the stage of prediction, depend on values of measurement results, and obtains a posteriori estimation. These two operations in the discrete-time systems are expressed using the equations(2.6) and (2.7)[42][43].

The problem of uncertainty can address by using Bayes trackers by modelling the state $x_k$ and the observation $z_k$ as two stochastic processes. Based on the Markovian rules (i.e. given the observations $z_k: k-1$ , the current state $x_k$ depends only on its predecessor $x_{k-1}$ and $z_k$ based only on the current state $x_k$ as a shown in figure(2.7), the recursion is fully determined by the observation equation (2.6) [44].



Figure(2.7): Graphical model steps of dependencies of tracking based on Markovian rules [44].

$z_k$= $g_k$ ($x_k, n_k$ ) ……………………………………..(2.6)

and by the dynamics of $x_k$, defined in the state equation $f_k$, as

$x_k$= $f_k$ ($x_{k-1}, m_{k-1}$) ……………………………………..(2.7)

$g_k$ and $f_k$ are a matrix describing the linear relationship between sequential states and between the states itself with observations and the two-noise represented by two parameters $n_k$ and $m_{k-1}$ have zero mean and covariances $R_k$ and $Q_k$ respectively.

The aim of a Bayes tracker is to predict $p_k$|k ($x_k|z_k$:k ), the probability density function (pdf) of the object being in state $x_k$, given all the observations $z_k$ up to time (k). The estimation process is achieved recursively through two stages, called prediction and update.

Mathematically, the KF is used under a hypothesis of linearity and Gaussianity of the prior $P_{k-1/k-1}($ $x_{k-1/} z_{1:k-1}$ ) and of the two noise processes, $n_k$ and $m_{k-1}$ as present in equation(2.8) and(2.9) [44].

$Z_{k=}\ G_k\ x_k\ +n_k$ ……………………………………………(2.8)

$x_{k=}\ F_k\ x_{k-1}\ +m_{k-1}$ ……………………………………………(2.9)

Where $G_k$ and $F_k$ are a matrix describing the linear relationship between sequential states and between the states itself with observations. Given the first- and second order statistics and given the mean $\bar{x}_{k-1}$ and the covariance $P_{k-1}$ of the prior $P_{k-1}$ $|k-1$ ($x_{K-1}$|z1: $k-1$), depend on the linear relationship of equation (2.10) and the prediction stage of equation (2.11),can obtain the statistics of the prediction density in the form of the mean prediction [44].

$\bar{x}_{k/k-1} = F_k\ \bar{x}_{k-1}$ …………………………………(2.10)

and the prediction covariance

$P_{k/k-1} = F_k\ P_{k-1} F_k' + Q_k$ …………………………………(2.11)

where the prior uncertainty propagates through the state transition as in Equation (2.10) and adds to it the uncertainty using the ($Q_k$ ). Also, from Equation (2.11) one can calculate the predicted measurement $\hat{z}_k$ as:

$\hat{z}_k = G_k\ \bar{x}_{k/k-1}$ ……………………………….(2.12)

When the new measurement ($z_k$)becomes available, from Equation (2.12) one can derive the mean residual as:

$\bar{r}_k =\ z_k\ - \hat{z}_k$ ………….……………………………(2.13)

the $S_k$ residual covariance is

$S_k \; = G_k \, P_{k/k-1} \, G_k' \; + R_k$ ..........................................(2.14)

And $K_k$ , the Kalman gain, as in Equation(2.15)

$K_k \; = P_{k/k-1} \, G_k' \, S_k^{-1}$ ..........................................(2.15)

Lastly, to achieve the recursion, the first- and second-order statistics of the posterior are the mean estimate using Equation(2.16)

$\bar{x}_k = \bar{x}_{\frac{k}{k}-1} \, K_k \bar{r}_k$ ..........................................(2.16)

and the $P_k$ posterior covariance calculated from Equation (2.17).

$P_k = (I - K_k G_k) P_{k/_{k-1}}$ .......................................... (2.17)

### 2.6.4 Learning-based Tracking Methods

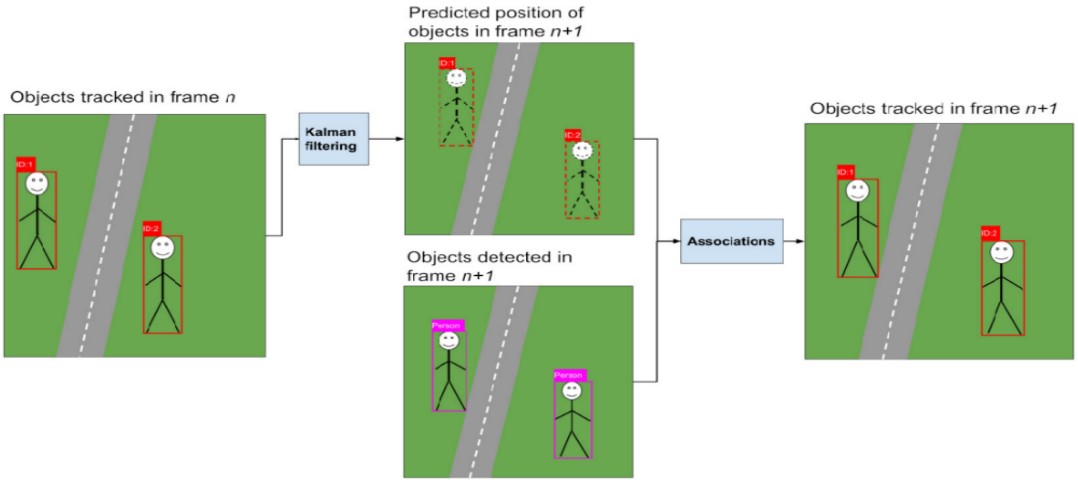
Methods based on learning such as deep learning approaches in our day are considered important methods for solving a number of challenges in many applications in computer vision, as improving the detection process by using deep learning-based object detectors to increase the performance of objects tracking.

There are many techniques of video analysis for pedestrians, which means detecting and tracking anomaly objects or individuals[45]. In recent years, the tracking-by-detection paradigm has developed and become popular, driven by the growth in object detection, such of these methods require two separate and important steps (detection of objects on all sequence frames then tracking by the association of those result detections over video frames). The tracking by detection with deep learning algorithms has become the most widespread tool in the multiple objects tracking (MOT) research area. This method is based on the advantage of object position knowledge to building an association model that would be able to join the states of objects across time, which computes the assumptions across measurements to predict if an object should be associated with a track, be set as a new track, or if it is a mismeasurement. It uses the KF to predict the states of object and distribution of probabilistic over assumptions to associate measurements to tracks[45][46].

Recent works also use the KF algorithm, as a motion model, to enhance the association task between states of objects over time. Proposed SORT algorithm to support a KF to improve the estimation task for object states, then follow by Hungarian model to linking the results of the predictions by KF with new object detections. A year later, Wojke et al. a new algorithm proposed called Deep sort, which is an improvement of SORT, based on a unique cascading association step that uses appearance features extracted based on the CNN model. The data association algorithm consists of the object similarity (features of appearance) with the metric Mahalanobis distance between object states and, at the last step for states that do not match, uses the SORT's data association[46].

- **The Deep Sort Tracking Algorithm**

Simple Online and Real-Time Tracking SORT is a algorithm of tracking that was introduced by Bewley et al. The main objective of SORT is to track objects across frames by associating the different states for unique moving objects in the tracking-by detection framework. The advantages of the SORT algorithm are that it only tacks the detection information from the previous and current frames, it makes the algorithm able to perform tasks online and real-time tracking. CNN-based object detectors used in SORT instead rely on more accurate object detections[46]. For each new frame, SORT first begins with propagating objects that are already tracked into the current frame. Then apply the prediction stage to predict all new coordinates of positions for these already tracked objects by KF based on a linear constant velocity model. Next, an object detection algorithm detects objects that appear in the current frame. These detected objects are then compared to already tracked objects and create a cost matrix. This matrix is responsible for computing the IOU between each detection and each of the already tracked objects. Then using the Hungarian method to assign these Detections to already tracked objects. A new track is initialized when an object is detected in many consecutive frames without overlapping with any of the already tracked objects. Figure (2.8) shows how to associate the predicted positions after comparing to object detections to assign identities in new frames. The disadvantage of the SORT algorithm because it does not have any memory and a tracked object is lost if SORT fails to detect it in a frame [11].


Objects tracked in frame n
ID:1
ID:2
Kalman filtering
Predicted position of objects in frame n+1
ID:1
ID:2
Objects detected in frame n+1
Person
Person
Associations
Objects tracked in frame n+1
ID:1
ID:2

Figure(2.8): Object association between frames in SORT[46]

Deep sort algorithm is improved of sort algorithm using (tracking by detection framework). Kalman filter has an incompetent in tracking task over various occlusions as they usually appear when obscure the vision from the body in view camera scenes. To deal with this situation the algorithm is based on using motion behavior and appearance information to complete the association task. The algorithm of Deep sort is fast, and powerful making it the better choice as an algorithm in detection by tracking method. It uses a pre-trained neural net to extract the interesting features for subjects so that an association can be made based on the degree of similarity for these features as well as overlapping. The algorithm of Deep sort has included the motion model, the appearance information, and the intersection-over-union (IOU) metric, and has been performed to track but it is based on information of detection in each frame. It is similar to SORT when reducing the number of identity switches and handles state estimations with a Kalman filter[46].

Deep SORT differs from SORT in that it makes use of additional techniques when assigning detections to already tracked objects, it is utilizing two different distance metrics when comparing detections to already tracked objects (Mahalanobis distance and cosine distance) between appearance descriptors. The Mahalanobis distance measures how the position of a new detection differs from the positions of already tracked objects in terms of standard deviations from the mean of the tracked objects. This metric allows Deep SORT to avoid assigning a new detection to an already existing track where the frame-to-frame motion would be unreasonable. Appearance descriptors are computed by forwarding each bounding box through a CNN that has been pre-trained previously on a dataset of a person re-identification. The descriptor of appearance for each detection of the new bounding box is then compared with the appearance descriptors of already tracked objects by computing the cosine distance metric value between two descriptors[47].

The advantage of the algorithm can save the tracked objects with its matrix of descriptors for a number of frames after they are lost so that this algorithm has the ability to resume tracking identities that have been lost for a number of frames. Using the values of the appearance matrix in this way gives Deep SORT the ability to detect a previously tracked object even if it has been occluded for a number of frames[47][46].

Deep Sort algorithm depends on both bounding boxes which represent the results of detection from the algorithm of detection such as YOLO and the information of appearance for moving object in the scene. The mathematical expression for associating the detections in the new frame with previous detections by using the Hungarian metric is used to assign the detections for previously tracked objects in a new frame to existing tracks when the cost function reaches the global minimum. The cost function involves calculate a visual distance $d^{(2)}$ that is based on the appearance features values of the detected object and the previous appearance features values of the tracked object with the spatial distance using Mahalanobis metric $d^{(1)}$ of the detected bounding box from the position predicted according to the previously known position of that object. The mathematical expression is given in equation(2.18) refers to the cost function that assigns a detected object $j$ to a track $i$[47].

$$c(i,j) = \lambda d^{(1)}(i,j) + (1-\lambda)d^{(2)}(i,j) \qquad \text{...........(2.18)}$$

Where the parameter λ can be set to determine the effect of the visual distance $d^{(2)}$ and spatial distance $d^{(1)}$.Where $d^{(1)}$ is given in equation(2.19) by the expression[11]:

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T {S_i}^{(-1)}(d_j - y_i) \qquad \text{...................(2.19)}$$

where $y_i$ refer to value of mean and $S_i$ refers to the values of covariance matrix of bounding box observations for the $i$ -th track, where the parameter $d_j$ represents the $j$-th detected bounding box. Where $d^{(2)}$ represent the parameter for visual distance based on the appearance feature descriptors and is given by equation(2.20) as[11]:

$$d^{(2)}(i,j) = \min\{1 - r_j^T r_k^{(i)} / r_k^{(i)}\} \in R_i \qquad \text{..................(2.20)}$$

Where $r_j$ is the parameter refer to descriptor of appearance which extracted from the interested part of the image within the j-th detected bounding box, and $R_i$

represents the last of 100 appearance descriptors $r_k^{(i)}$ linked with track $i$. The parameter $d^{(2)}$ measure depends on the cosine similarity between the i-th track and j-th detection to determine the track where visually the most similar detection was previously found in the current detection. The appearance descriptors extracted using a wide residual neural network that consists of two convolutional layers followed by six residual blocks that output a 128-element vector. The network is pretrained on a person re-identification dataset where the dataset is about more than a million images of 1261 pedestrians[47][11].

## 2.7 Trajectory of Moving Object

A trajectory is a sequence of location points for a moving object, it's connected to drawing a path through space as a function over time for each object moving in a scene. To detect the precise and complete trajectory of many objects with different motions in the scene, that needs to overcome some difficulties such as data capture and storage methods. The more accurate the trajectory is mean the capture more points to represent the shape of the trajectory as shown in Figure(2.9). However, using the high rate for obtaining the points of position for moving objects to produce the trajectories paths may be give a huge of data and can cause a high cost in the storage of data, cost in processing and communications [48]. The positions of tracked objects are sent periodically as a message to the central server. The points of location data obtained from trajectories are then uploaded as a database for moving objects. On the other hand, there are many applications that can achieve a query to the central server to retrieve interest behaviors of moving objects (as well as their properties such as positions and other patterns discovered from behaviors of moving objects) to meet various application requirements[49].

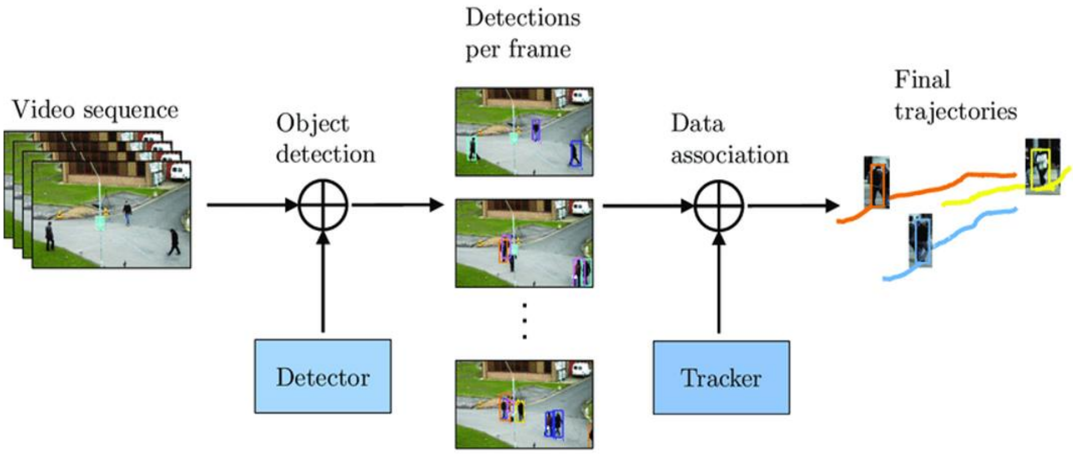


Figure (2.9): The object detection, tracking and data association for final trajectories[40].

### 2.7.1 Trajectory Construction

A trajectory is a path represented by location points, as a result, to object motion over time in space. These points are denoted by a sequence of coordinates as $\{x_1, y_1, t_1, x_2, y_2, t_2, \ldots .x_N, y_N$ and $t_N$ $\}$ where $(x_i, y_i)$ represent the geographic position of the object at each time steps $(t_i)$ and the symbol of (N) is referring to the all number of steps in the series. The trajectory points detection means capture each point of position in each frame represented by coordinates (x, y) at time (t). the position at each frame is drawn separately for two of its coordinates (x) and (y) along time[50].

The trajectory length is used as an index for evaluation, where the longest trajectory is selected and considered as the accurate trajectory, otherwise rejecting the trajectories which have shorter lengths. In fact, the length of the trajectory is not used for just evaluation but also is used for separate trajectories by measuring the angle and the distance between two sequential candidates in the trajectory. Sequences of frames are collected into one track which is used to build a unique trajectory of the moving person. A prediction model and a Kalman filter algorithm are used to confirm the positions of the target and to estimate the lost candidates respectively, many techniques are used for keeping and predicting the new positions of the moving target for constructing the trajectory of the target in the scene. There are many techniques used to predict the position of objects which move across frames of video. Therefore, when a target moves at a specific time consequences can be based on known or predicted speeds across elapsed time[50].

### 2.7.2 Trajectory Features Extraction

The tracking task for any object can be described by keeping the identity of the moving object in a sequence of frames. This process can be performed depending on appearance features by using the extracted features used to assign (id) for each object after the detection process. For more accuracy can calculate the motion features as the position, velocity and direction for the location's values of the moving target in each frame. Due to detection that results from YOLOv3 for a target which represented by plotted a rectangle bounding box around the foreground of the target in the scene and using the dimensions of the bounding box to determine the centroid of the box. The location of the centroid is representing the position of an object at the specific frame and many frames mean many points which are used to construct the trajectory, these points of the centroid are stored in the list then it used to calculate the distance between two points based on Euclidean distance metric. The object speed moves from frame to frame is computed using the distance and frame rate of the recorded video[51]. The distance step after each frame can be determined by using the centroid of the bounding box of the moving target, computed using the Euclidean distance metric as shown in equation(2.21). The parameters are represented by the pixel location of the moving object from the initial stage to the final stage [52].

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.21)$$

Where $x_1$ is refers to a previous position of object and $x_2$ present a pixel position after one step in width (next frame) and in a same way the $y_1$ parameter refers to a previous pixel position and $y_2$present pixel position after one step in height (next frame). The object velocity can be defined by using the value of distance divided by the value of frame rate at each step. The velocity defined is of 2-dimension (fixed camera), Velocity of moving object can show in Equation(2.22):

Velocity = distance between two frames / Frame rate ……………………..(2.22)

Can define the velocity of moving object in scene in pixels / second [53].

Direction feature of moving object can compute at each location using the equation (2.23) as follow:

$$\Theta = tan^{-1}\left(\frac{\text{Opposite line}}{\text{Adjacent line}}\right) = tan^{-1}\frac{y2-y1}{\text{x2}-\text{x1}} \qquad \text{……………………… (2.23)}$$

When (Θ) value is represent the angle of direction computed between two coordinate points( previous and current point of target location in scene). Then it can calculate the one value of four direction( East, West, North, South) using the value of (Θ) with any change in coordinates for (x) and (y) values [52][53].

## 2.8 Data Reduction

Data reduction is an important step in many applications which deal with enormous volumes of data. The systems which used a service based on location such as capturing a trajectory of a moving target, this application usually produce a huge volume of data represented by target positions with measurement noise. Therefore, the filtering methods and reduction tasks that apply to massive data are essential for transmission, cleaning, and storage of data points of trajectory[50].

When the target is moving with a continuous path, the trajectory of an object will be represented in a discrete manner as a result of the nature of the sampling-based data acquisition technique[54]. Therefore, any approach that can be used to perform reduction on the big data of trajectory should reduce the sampling rate for data obtained or be based on any strategy to decrease the number of sample points in the trajectory. Nevertheless, the reduction process must be performed and dropped some of these points without effect on the data quality of trajectory. The important points are necessary for supporting the tracking applications. Add on that, identify these approaches that can be used to efficiently filter measurement noise not only in the raw location points of trajectories but also in high-level features of trajectories such as velocity and direction. The idea of normalization can be referred to for two main purposes, the first is the need to shrink by simplifying the trajectory to obtain more knowledge from the trajectory. The strategy of search for the trajectory data points and the metrics are responsible to evaluate and decide which of them will save in the new list[50].

Reduction position points of trajectory depending on the angle consider is one of many strategies for point selection criteria used for selecting the interesting points that can be suitable for specific applications. Many types of normalization algorithms depend on an angle to reduce the trajectory points using the difference of angle between several trajectory points to make the decision. When performing reduction, the angle produced based on the trajectory points can be compared with other angles along the trajectory. Due to the noise, many angles may appear as a result of the low accuracy of trajectory data. When dealing with the angles there are several optimal and suboptimal simplification algorithms which specific direction preservation[55] as:

**1-**Latecki and Lakämper this method depends on the angle difference between the current and previous direction vector.

**2-**Wang et al this method uses three consecutive trajectory points to generate the angle. The generated an angle, which is intermediate with respect to the other two is named the open angle, where this angle is far from 180 degrees, show a sharp turn that must be kept in the normalized list, and compute the difference between angles, the current and the previous direction vector[55].

**3-**Ke et al. suggesting a grouping the difference of the angular values of the vectors, by the comparison the sum of a change of segment with specific threshold. Usually, the change in segment is result from change in several trajectory points as in Figure (2.10)[55].

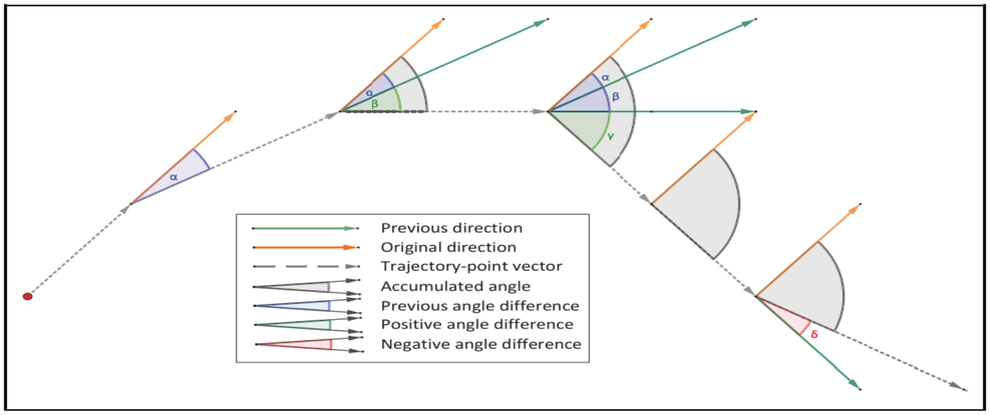


Figure (2.10): Example of angle comparative criteria[55]

## 2.9 Deep Learning Networks

Deep learning is technique depend on multiple processing layers to construct computational models used to solve many problems through discover intricate structure in large data sets depend on learn representations of data with multiple levels of abstraction. That indicates its concern with a manner for a machine to change its inner parameters used to calculate the representation in each layer from the prior layer representation. It's a new Machine Learning trend that makes Machine Learning closer to Artificial Intelligence. Many applications improved with deep learning such as visual object recognition, object detection, speech recognition, drug discovery[57][25].

### 2.9.1 The Convolutional Neural Network

CNN consider is one of Among popular kinds of deep neural networks, it has been used commonly with the application of video and images. It can analyze the original images to obtain robust features directly from raw pixels. The architecture of CNN is ANN with at least one convolutional layer. The type LeNet-5 is created from developed a multi-layer of ANN. However, the property of the network is the usage of the convolution process in one of its layers at least. The CNNs are designed to process even images with small sizes which comprise a huge amount of information. Assuming that each pixel intensity of image with size (416*416) is taken as an input to a fully connected network, that means every neuron of this network needs (173,056) weights, due to the increase in image size the number of free parameters in the network becomes extremely large. This may be making the performance slow and leading to an over-fitting case. The advantage of CNNs, they can reduce the large number of free parameters using the convolutional layers. Another benefit of using CNNs is their invariance during translation. In many tasks for pattern detection, the same pattern can be detected in different places in the image, which means is inefficient to train neurons to distinguish the same pattern in different positions independently[14]. In the next section, we will review the important CNN layers such as convolutional, pooling, and fully connected layers.

- **The Convolutional Layer**

The aim of this structure is to capture the interesting features (learn features) from the raw values of the inputs. It is comprised of a number of convolution kernels that may be different sizes that are used to recognize and calculate different feature maps. These interesting maps of Features can be captured firstly by applying a learned filter for convolving the input values and then passing the obtained results to a suitable non-linear activation function. In Figure (2.13), the kernel is shifted with a specific stride to convolute the input values, then at each position, these values product with kernel values to superimposes (receptive field) is taken to obtain the activation map on the right, which is then passed to the activation function. Repeating this process using different kernels to gather as many feature maps as desired. The goal is to update the weights of the kernel to improve the final decision of (classification/regression) which result from the whole network [57].

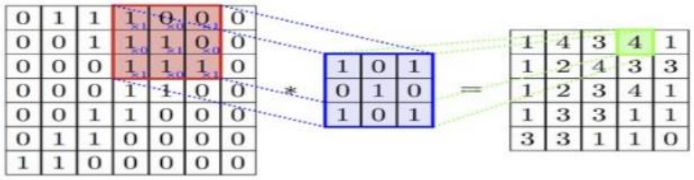


Figure (2.13): The convolution operation used in Computer Vision[57]

Note that the number of kernels is equal to output feature maps. The shape of the resultant feature map can be determined by the equation (2.24)[62]:

$$fn = \frac{i_n - k_n + 2P_n}{s_n} + 1 \quad , n = 1,2, \dots \dots \dots N \qquad \dots\dots\dots\dots\dots\dots\dots\dots(2.24)$$

Where $(fn)$ is the feature map shape in the axis $(n)$. where $(k)$ and $(i)$ are kernel shape and input shape parameters, in the axis $(n)$, respectively. $(p)$ is the amount zero-padding to add on the axis $(n)$, and the amount by which the filter shifts is the stride length $(s)$.

- **The Pooling Layers**

A common problem of feature maps is that they are sensitive to the location of the inputs. The pooling layer aims to perform shift-invariance on result values from the convolution layer depending on an approach to complete down-sampling feature maps. It is applied on a results value between two convolutional layers. Figure (2.14) present the basic pooling methods, average and max pooling. Note that, unlike convolution kernels, the pooling layer has no learning parameters [62].

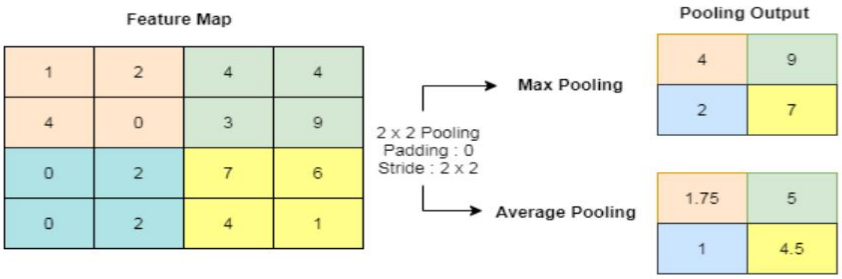


Figure (2.14): The result of max and average pooling[57]

- **The Fully Connected Layers**

The aim of this part of the network is to achieve high-level reasoning. It takes the last layer at the end of CNN architecture; it came after complete steps of many convolution layers and several average/max pooling layers. FC layers depend on extracted high-level features from previous layers, this layer will try to detect a class score or predict coordinates of the object as output from applied activations. FC layer takes all neurons in the previous layer and connects them to every single neuron of the current layer to produce global semantic information. Note that a 1 × 1 convolution layer can replace FC layers Nowadays, there are many techniques introduced to enhance the CNN learning process such as dropouts, regularization, $\ell p$ norm regularization, data augmentation, and batch normalization [57][58].

### 2.9.2 Recurrent Neural Networks (RNN)

This network is a special kind of ANN network designed to work with sequential data. It can be used with many applications when all outputs and inputs are independent of each other such as tracking, forecasting, natural language processing and many other applications which need dealing with data sequence. When the data can be a series such that the result of the current value based on the previous value, this kind of data needs to use a suitable type of neural network to integrate the dependencies between these data points. RNNs depend on the idea of 'memory' that helps them to save the sequence of states or the previous information of inputs to produce the next output of the sequence[14].

RNN can be dealing with a series kind of data well. The output which results from the previous time sequence pass to the hidden layer which is used as a kind of memory that retains learned features and then send it to the next step using features from the previously hidden layer and the input layer. The hidden layer receives weights from the input layer and generates an output through an activation function. RNN achieves prediction or classification by learning sequential data from algorithms of deep learning. CNN models used the virous kernels within convolutional layers to data processing as transform data before being passed to the next layer[14][56].

The RNN network with the full sequence as shown in figure (2.15) can be explained when ($x_t$) represents the input at time step ($t$). To make it clear, if this is a tracking process, ($x_t$) will represent the detection coordinates of a tracklet and ($t$) represents the frame number, ($s_t$) is the hidden state (memory of the network); note that we might have one shared state for all time steps. $U$, $V$ and $W$ are the parameters to be learned. $U$, $V$ and $W$ weights the input, output, and the propagated hidden state, respectively, hidden states and outputs are computed in equations (2.25), and (2.26) as follows[14] :

$$S_t = \mathcal{F}(U \cdot X_t + W \cdot S_{t-1}) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (2.25)$$

$$O_t = softmax(V \cdot S_t) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.26)$$

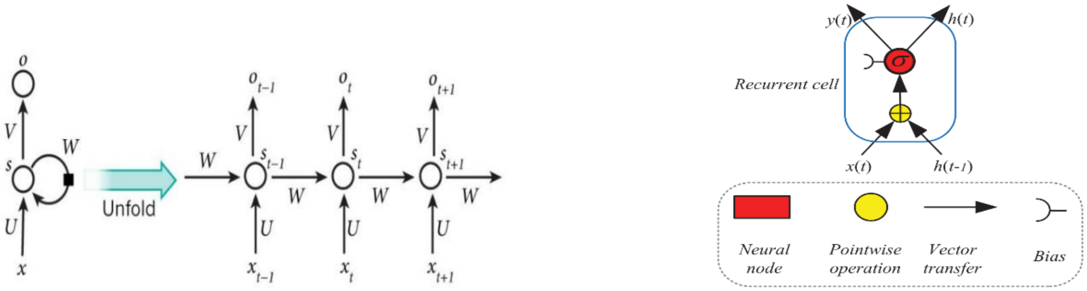


Figure (2.15): The recurrent cell with(input, output and recurrent process)[14]

The function ($\mathcal{F}$) represents the non-linear activation function, usually $tanh$ is used. Backpropagation Through Time is utilized to update the learnable parameters, the main problem with RNN is that they don't have long memories. The reason is that when we update ($W$) in the unrolled temporal loop, the gradient descent needs to backpropagate the gradient from every time step and multiply by the state until reaching the first-time step. The weights of neural networks are initialized close to zero, the more time steps we have, the faster $W$ weights will vanish due to consecutive multiplications. On the other hand, if the weights are initialized above 1, $W$ values will explode when backpropagating the gradient for the same reason, usually, this issue is referred to as "exploding gradients". The types of RNN are (Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM), Bidirectional recurrent neural networks (BRNN)). In next section will explain the LSTM models[56].

- **Long Short-Term Memory Neural Networks (LSTM)**

LSTM is a particular type of RNN, due to the architecture of the network, the model is efficient and able of learning long-term dependencies. At first, once is introduced by Hochreiter and Schmidhuber (1997). LSTM is passing the input data through the hidden layer with short and long-term states and creates output data through the output layer. RNN and LSTM are similar in structure when both networks have two states in the hidden layer (short-term states and long-term states), but the RNN has a limitation called gradient vanishing. LSTM has overcome this problem by solving gradient vanishing by using three gates. The architecture of LSTM is used a forget gate which decides if the network will remember the last information in RNN or not. The figure in (2.17) show the internal design of the LSTM comprising three gates (input, output and forget) that are controlled by state information for the current node. The forget gate decides whether to retain the old state information or not, where the input gate decides whether to save new information being entered, and the output gate is responsible for the output of updated cells. LSTM is used with sequence data or with extracted features for target tracking in single and multiple cases [61][62].

A standard LSTM cell (Figure 2.17) has four elements:

**A-Input gate:** It updates the cell state $Ct-1$ . It takes the input $xt$ at time step $t$ with the previous hidden state $ht-1$ to decide on which weights need to be updated for a better prediction as presents in equation(2.27).

**B-Cell state connection:** a link that connects all gates together to output the new cell state $Ct$ as shown in equation(2.31).

**C-Forget gate:** Prevent the internal state weights to grow without limit (solves gradient expansion problem). It provides LSTM the ability to reset the cell state $Ct-1$ if it needs. In other words, the essential state values are weighted by a number close to one. Otherwise, a number close to zero as shown in equation (2.28).

**D-Output gate:** It updates the previous hidden state $ht-1$ by weighing it with the new cell state $Ct$ as present in equation(2.29).

Can explain the symbols which refers to the output-specific node and then the inputs of others. The circles with pink color represent operations, such as addition or dot product that apply on vectors, and the yellow boxes are learned neural network layers. Concatenation is referred to as Lines merging, and a line forking shape refers to dividing its content into copies where each copy goes to a different location.

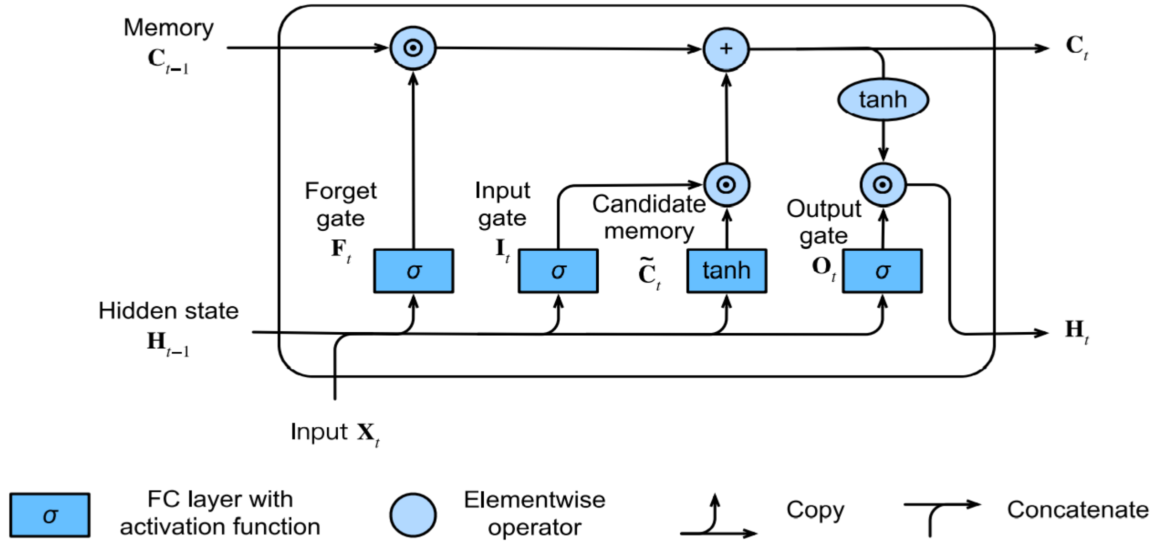


Figure (2.17): Structure of LSTM network, when each line contains an entire vector[62]

The cell state is one important factor in LSTMs , the horizontal line shows the state of cell when it is passing through the top of the diagram. The cell state works as conveyor belt. It joins the entire chain, with a linear acts and information flow along it without change. The design of network is allowed to add information to cell state by gates or remove information from it. Gates are combination of a sigmoid function and a multiplication operation. LSTM can protect and control by the cell state through three of these gates . As a new input comes and input gate it is activated, new information will be collected to the cell. Also, if the forget gate ($f_t$) was active, the past cell status $c_{t-1}$ could be “forgotten”. The output gate(ot) is controls whether the latest cell output $c_t$ propagated to the final state ($h_t$) or not. The LSTM architecture use the history of information which store in memory cells to predict a long-term temporal relation. The nonlinear equation of $\sigma = (1 + e^{-x)})^{-1}$is represent the mathematical expression for sigmoid function which take outputs values between zero and one. When zero value means “nothing pass,” while a value of one means “let everything through!”[34]. Parameter $(i_t)$ refer to the input gate , forget gate represented by parameter ( $f_t$), output gate $(O_t)$ and final state $(h_t)$ is defined as following[63]:

$$i_t = \sigma(W_{X_i}x_t + W_{h_i}h_{t-1} + b_i) \quad \text{(2.27)}$$

$$f_t = \sigma(W_{X_f}x_t + W_{h_f}h_{t-1} + b_f) \quad \text{(2.28)}$$

$$O_t = \sigma(W_{X_o}x_t + W_{h_o}h_{t-1} + b_o) \quad \text{(2.29)}$$

$$g_t = tanh(W_{X_c}x_t + W_{h_c}h_{t-1} + b_c) \quad \text{(2.30)}$$

$$c_t = f_t * c_{t-1} + i_t * g_t \quad \text{(2.31)}$$

$$h_t = O_t * \tanh\,(c_t) \quad \text{(2.32)}$$

The basic difference between LSTM and classical RNNs is the use of these gating functions$i_t$ , $f_t$ , $O_t$, which indicate to the input, forget, and output gate at time t respectively. Weight parameters $W_{X_i}$, $W_{h_i}$ , $W_{h_f}$, $W_{h_o}$, $W_{X_f}$ , $W_{h_c}$ , $W_{X_o}$and $W_{X_c}$ , link the different inputs and gates with the memory cells and outputs and biases $b_i$ , $b_f$, $b_c$ and $b_o$. The cell state ct is updated with a fraction of the previous cell state $c_{t-1}$that is controlled by $f_t$.

## 2.10 The Object Re-identification Process

Person re-identification in video monitoring systems has become a significant research topic in recent years. Due to the suitable cost of cameras and the development of data storage technologies, there are many establishments, schools, and government offices that have installed camera systems more than ever before[63].

This type of camera system has allowed its owner to gather intelligence information to prevent crime, in addition to analyzing previously recorded video to produce any report on each behavior. Traditionally, video surveillance has depended on individual security officers to see an increasingly large number of video feeds and make decisions. Unfortunately, as more video feeds are shown on the screen, the person responsible for monitoring finds it more difficult to make accurate evaluations of the information provided to them. Therefore, video analytics by these systems can process the information for several cameras feeds at the same time and make the same astute observations that a human can need to be developed[64].

The goal of the human re-identification model is to recognize specific people in the scene and across cameras with non-overlapping views. The model can recognize a person in one camera and extract a set of features that the model can learn. When the same person walks through a number of cameras distributed in the environment, the model must be able to compare them to the learned features and identify them. The re-identification scenario is shown in Figure (2.18), where several subjects pass through the different camera feeds but retain the same ID[65].

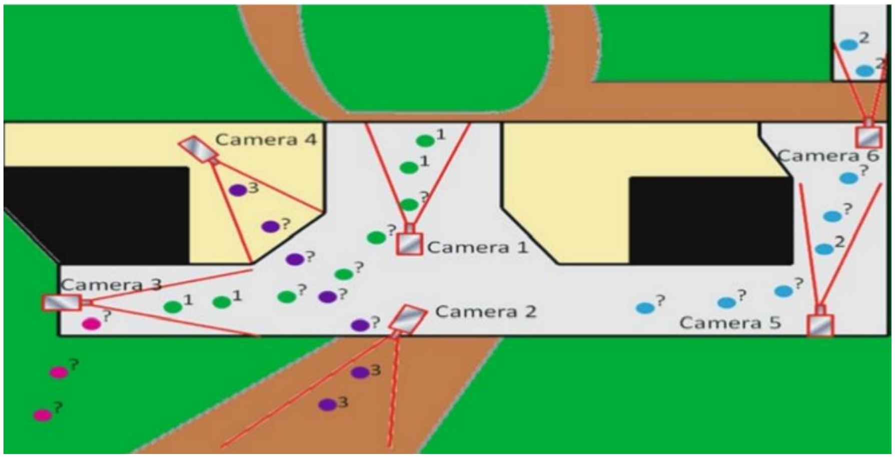


Figure (2.18): A basic human re-identification scenario[64].

Tracking multiple or single objects across non-overlapping cameras aims to predict the path of trajectories for all moving targets in the scene and keep their identity labels consistent while they move from one camera to another. Person re-identification(Reid) means how can associate images of the same person taken from the same camera or from neighbor's cameras at different points in time. However, the object (Reid) remains a notably challenging task, because the object may be changing his appearance, pose across camera views due to important changes in viewpoints, illumination, or even cluttered backgrounds. discover the Deep Neural Networks (DNN) recently support building different models of network architectures performing high-performance levels[64][65].

The methods of re-identification can be classified into two main groups. The first group comprised of the methods that try to extract signatures from texture, color and other properties of appearance in frames. This group called appearance-based methods. On the other hand, the other methods are based on features extracted from the motion and gait of persons of interest. This group called gait-based methods. Feature extraction is the first step in the Re-ID model, which is considered significant for better performance. The recent methods based on deep learning approaches (CNN, RNN) are still suitable methods to perform the Re-ID process[65].

## 2.11 Object Occlusion in Cameras Network

Many applications with different types are based on the task of object detection, such as person tracking, traffic monitoring, understanding and analysis of scenes, etc. detection process still has many limitations exist while performing detecting an object such as a change in its appearance, low visibility, angle of view, cast shadows and most importantly occlusions of the object[7].

Occlusions can take different situations that may occur when A target is totally or partially occluded by other objects in the scene. Occlusions are usually due to one part of an object is occluded by another part of a static object, like a wall, desk or a column. totally occluded occurs when other moving objects obscure the view of a target totally in a scene[60].

Depending on the topology of the network and the long period when a person occluded can classify the occlusion of a moving target into two categories (total and partial occlusion). The first kind is when the target moving is totally occluded in-camera scene for a specific time then the target may reappear. The second type represent the occluded when the target is partially occluded. The walking person across cameras produces many challenges such as many changes occurring in his pose and illumination or changing his clothes[44].

To overcome these challenges and solved the association problem for states before and after occlusion there are many approaches applied used in different ways such as deep sort, RNN, LSTM, etc depending on the unique features of some general properties represented by methods like the histogram are suitable to cope with occlusions. Also, multiple localized features can be used which contain interesting information from the target may improve the performance of a tracking task. Appearance information for the target may be is not sufficient to deal with total occlusions. There for, can be using higher-level reasoning or through multi-hypothesis approaches that maintain spreading the assumptions of tracking across time. The manner of motion and the previous knowledge about occlusion patterns can also be used to spreading the points coordinates of trajectory for a target when the valid measurements were unknown. When the target reappears from the occlusion, the modeling of appearance can deliver the necessary signs to reinitialize a track[60].

Position and speed target can be used to predict future positions for a target moving over a scene. The occlusion detection routine predicts future locations of objects, depending on current predictions of speed and position. Depending on the estimated parameters of the bounding box and the binary maps for the tracked objects, occlusion is estimated, detected and handled by estimation positions and speeds followed by applying these estimates to the image plane [62].

## 2.12 Evaluation Measures

Evaluating any model is an essential part of any system. When evaluated a model using one metric may produce a good skill score and satisfying reasonable results but when evaluated against other metrics the results are poorly. Therefore, to evaluate the results of system stages as Detection, tracking, prediction and re-identification must using suitable metric to calculate the accuracy that is used to measure the performance of a model, but this measure cannot represent the true judgment of a model. Therefore, more one evaluation metrics must be used to evaluated system performance. Using one evaluation measure to evaluation is not sufficient to valuation the model. As mentioned, the model can achieve satisfying results with one metric, such as “accuracy score” but achieve poor results with other such metrics as “logarithmic loss”. Many kinds of evaluation metrics are available like:

### A) Accuracy Measure

Accuracy is the output of divided the number of correct predictions on the total amount of input samples as shown in equation (2.49) [46].

$$\text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{total number of predictions made}}\ 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.49)$$

This measure produce well evaluation if the training set have an equal number of all classes. That's mean Classification Accuracy works well but makes false sense to achieve high accuracy.

**B) Precision:** is the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative the concluded value lies in the range [0,1] as shown in equation(2.50)[66].

$$Precision=\left(\frac{True\ Positive}{True\ positive+False\ positive}\right)\ 100\%\ \dots\dots\dots..(2.50)$$

**C) Recall:** is the average of true positive data points to sum of true positive and false negative data points and the concluded value lies in the range [0,1] , as shown in equation (2.51)[46][66].

$$Recall=\left(\frac{True\ Positive}{True\ positive+False\ negative}\right)\ 100\%\ \dots\dots\dots\dots(2.51)$$

**D) F1 Score:** F1 Score represents the Harmonic mean of precision and recall measures to measure a test's accuracy which determines the precise of how many instances it classifies correctly and robust by cannot ignoring 84 significant number of instances of the model. F1 Score has [0, 1] ranges and the greater value indicated to the better performance. F1 Score can be expressed mathematically as equation (2.54), the precision presented in equation (2.52) and Recall presented in equation (2.53)[46].

$$Precision=\left(\frac{True\ Positives}{True\ positives+False\ Positives}\right)\ 100\%\ \dots\dots\dots\dots\dots..(2.52)$$

$$Recall=\left(\frac{True\ Positives}{True\ positives + FalseNegatives}\right)\ 100\% \ldots\ldots\ldots\ldots\ldots\ldots..(2.53)$$

$$F1=2*\left(\frac{1}{\frac{1}{precision}+\frac{1}{recall}}\right)\ 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.54)$$

**E) Mean Absolute Error (MAE):** Mean Absolute Error is the average of the difference between the predicted values and actual values. It measures the distance the predictions were far from the actual output but without indicating the direction of error under or over the predicting data. Mathematically representation of MAE shown in equation (2.55) [46]:

$$MeanAbsoluteError= \frac{1}{N}\sum |y_j - \hat{y}_j\ | \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.55)$$

**F) Mean Squared Error (MSE):** Mean Squared Error (MSE) is the average of the square of the difference between the actual values and the predicted values. The easily computing of the gradient can be considered the main objective of MSE, that's allow the effect of larger errors become more powerful than a smaller error based on taking the square which causes the model focus on the larger errors. MSE represented mathematically as equation (2.56) [46]:

$$Mean\ \text{squared}\ Error= \frac{1}{N}\sum(y_j - \hat{y}_j)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.56)$$

# CHAPTER THREE

# THE PROPOSED SYSTEM

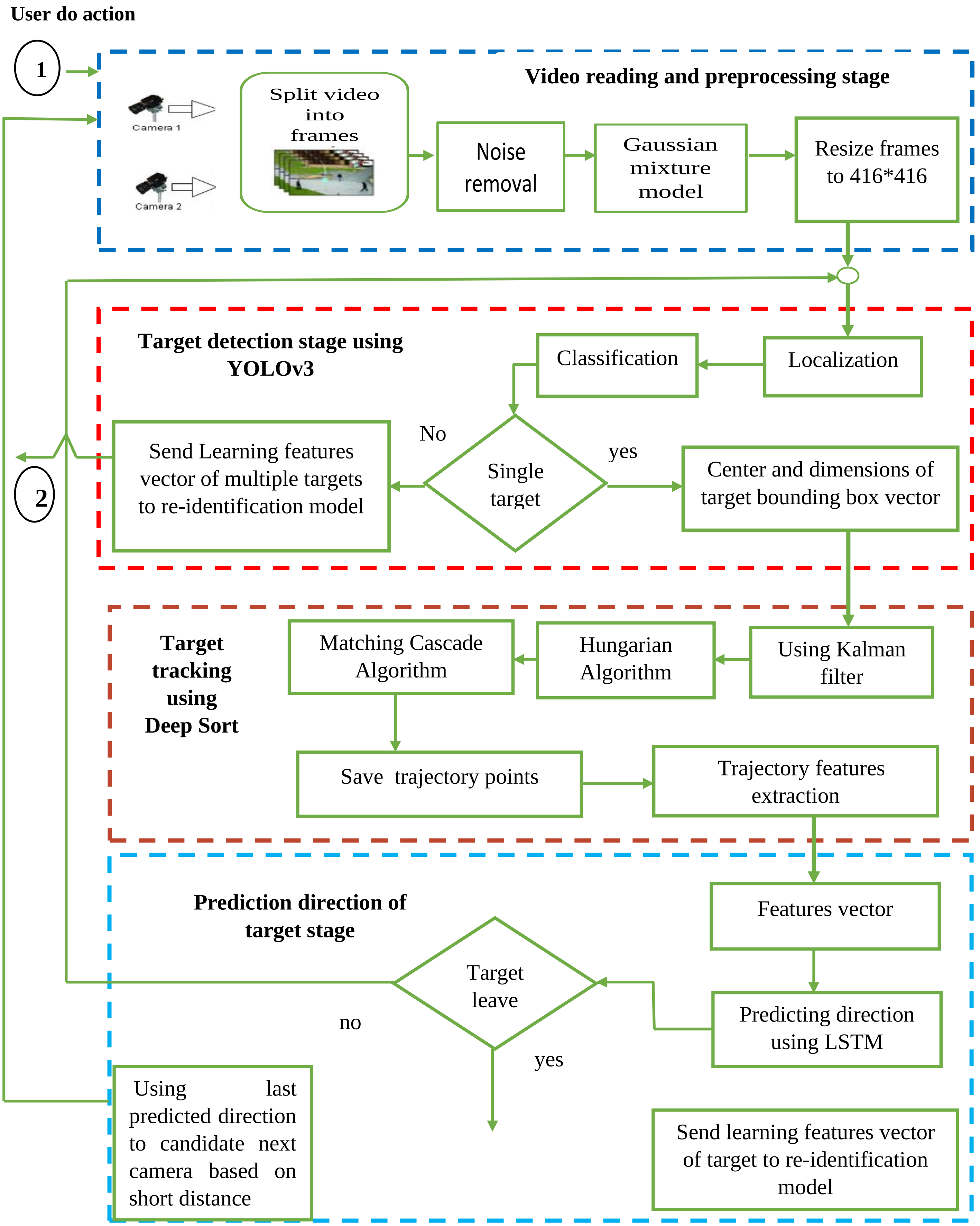
## 3.1 Introduction

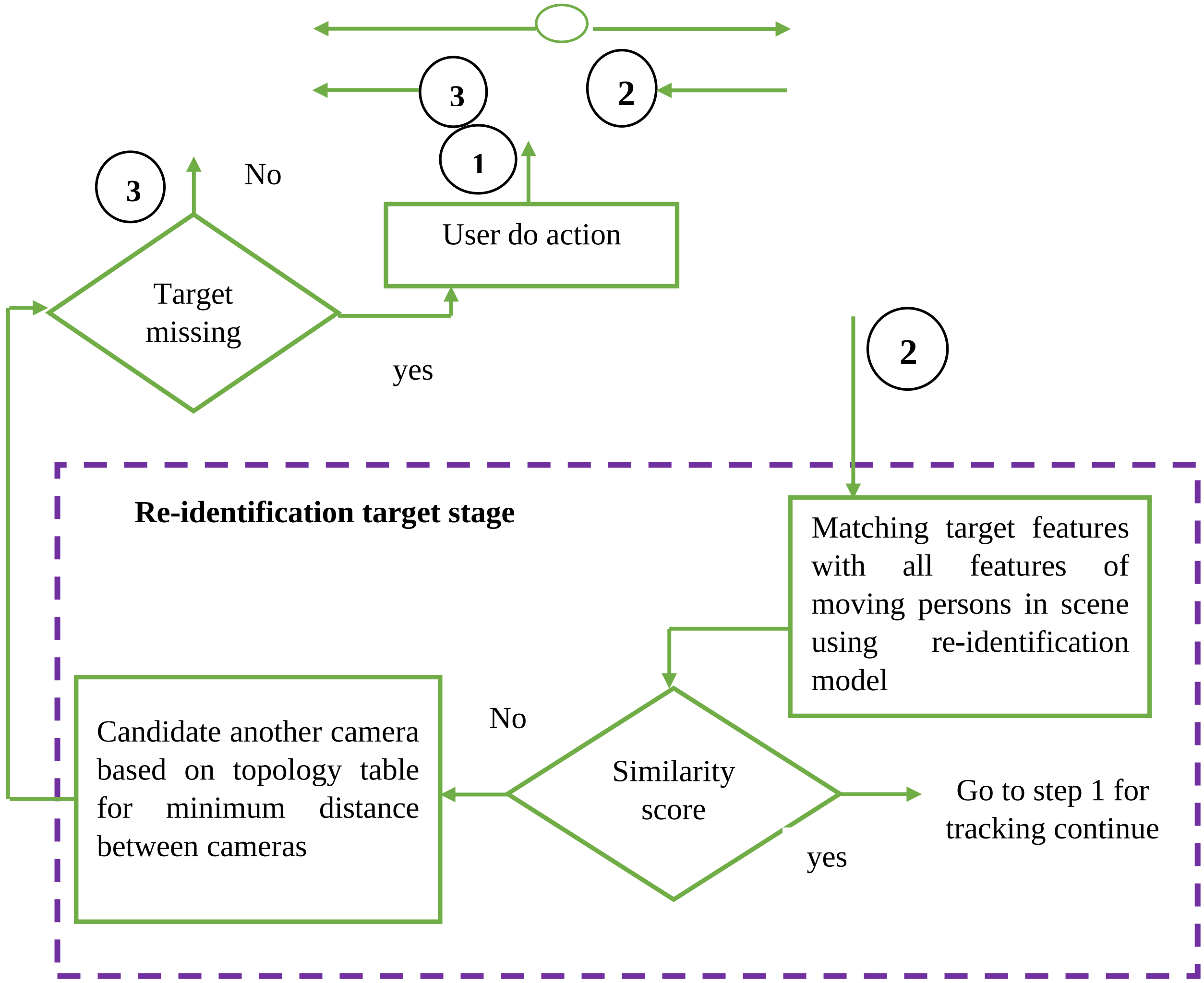
This chapter states and identifies all stages of the proposed surveillance cameras network system (SCNS) based on many methods that suggested achieving and investigating the prediction across network based on number of techniques and important object features to success and accomplishing the system requirements and its goals. SCNS is a group of video analysis, processing of image and deep learning techniques for detecting, analyzing, tracking, prediction and re-identification to moving object across camera network based on extracting learning with trajectories features, then tracking these features to make an accurate prediction about location and direction of target and where the target may be reappeared in network. The robust prediction of new locations and direction that will improving the tracking process and minimize the computation of motion cost.

The proposed system introduces and explains many used techniques and proposes and designs many algorithms to be applied. More details and concepts are mentioned with its elaborated thoughts and some results from pixel level to events level for clarifying the work as well as the experimental study that designed to apply the idea of dissertation.

## 3.2 Stages of the Proposed System

The general flowchart of the proposed SCNS includes all stages and their embedded steps. The proposed system has performed the stages as the preprocessing stage then detection stage used YOLO algorithm followed by tracking stage based on Kalman filter with deep sort algorithm then prediction stage using LSTM model. At last, re-identification stage is when each stage contained several steps performed with different functions and all stages are explained in detail in next sections. The proposed system is shown in the following Figure (3.1).


User do action
1
Video reading and preprocessing stage
Camera 1
Camera 2
Split video into frames
Noise removal
Gaussian mixture model
Resize frames to 416*416
Target detection stage using YOLOv3
Classification
Localization
No
Single target
yes
Send Learning features vector of multiple targets to re-identification model
Center and dimensions of target bounding box vector
2
Target tracking using Deep Sort
Matching Cascade Algorithm
Hungarian Algorithm
Using Kalman filter
Save trajectory points
Trajectory features extraction
Prediction direction of target stage
Features vector
Target leave
Predicting direction using LSTM
no
yes
Using last predicted direction to candidate next camera based on short distance
Send learning features vector of target to re-identification model


3
2
1
3
No
User do action
Target missing
yes
2
Re-identification target stage
Matching target features with all features of moving persons in scene using re-identification model
Candidate another camera based on topology table for minimum distance between cameras
No
Similarity score
yes
Go to step 1 for tracking continue

Figure(3.1): The general main stages of the proposed system

### 3.2.1 Dataset Description

The proposed system used two types of datasets for system construction through training and evaluated models. The first dataset is named Multi Camera Track Auto MTA, it consists of six cameras, 2,840 persons, 100 minutes, and resolution 1920*1080. These cameras are distributed in large area with overlapping and nonoverlapping topology and are divided into training and testing videos. The dataset contains different places with change in illumination and pose during moving person in scene and across cameras. Each camera has a ".dat" file, which contains an M*7 matrix, M refers to the total number of bounding boxes for this camera. Note that M varies with cameras, each line corresponds to a bounding box. It contains 7 elements denote to the camera's number, the frame number, the object label, the location and the bounding box size respectively.

This example to clarify, the line "1 1 0 90 10 24 51 " means:

Camera's Number: 1

Frame's Number: 1

Object Label : 0

The upper left point of the bounding box (x, y): (90, 10)

(Width, Height) of the bounding box: (24, 51)

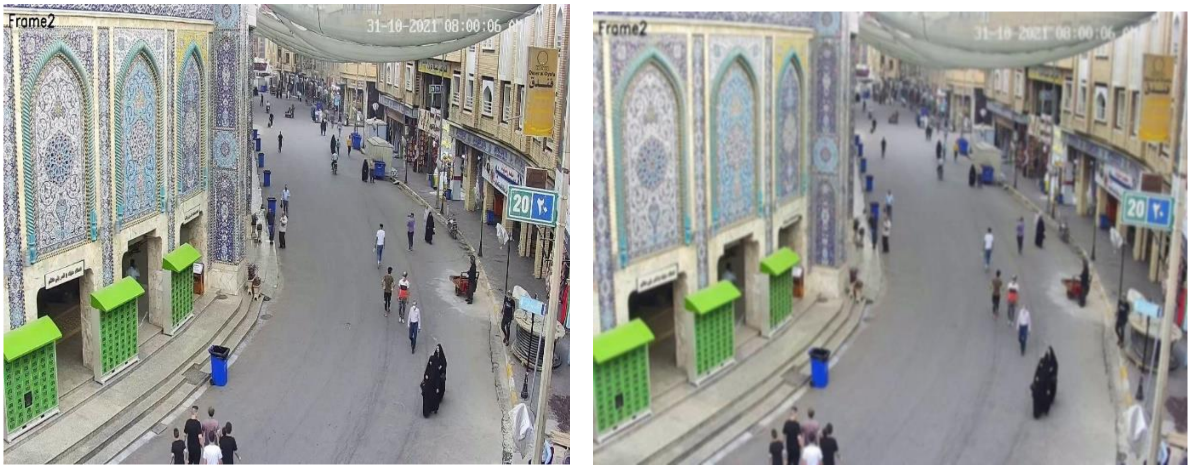
The second type of dataset is number of videos from six cameras of *Abbasid shrine in Karbala city.* Using different scenarios in each video for many persons, the time for each video about 15 minutes. Cameras topology was non-overlapping distribution.

### 3.2.2 The Preprocessing Stage

Initially, the proposed system reads video data, then divides the video into number of frames to make dealing with the type of data simple. Splitting video into frames must be an appropriate with proposed system of this dissertation through balance the frame rate per second (fps) with works proposed system.

Sometimes an image needs some operations to prepare for detection and tracking, after splitting video into frames there are some frames have resizing, blurring, or noise must these effects removal. There are many causes for appearing noise in images one of the important noisy image causes is lens optical in digital camera that captured visual information where if the camera doesn't exactly focus that give blurred image. Another issue that faced is outdoor changes like foggy weather and ambient illumination called variation appearance. Therefore, it needs techniques to enhance and filter the images. The outcomes frames should be free from noises and aberrations. There is a list of some filtering and image enhancement operations that are used in proposed system as gaussian blur filter and morphology operations.

Gaussian blur filter model is applying gaussian function on blur image, and it is typically used to reduce detail and noise. The image produced from gaussian blur is a smooth image as present in figure (3.2). It is also used with images that captures affected by lens out focuses or images that have shadows too. Gaussian blur is low-pass filter when the Gaussian function is diminished the components of high-frequency for image.



a b

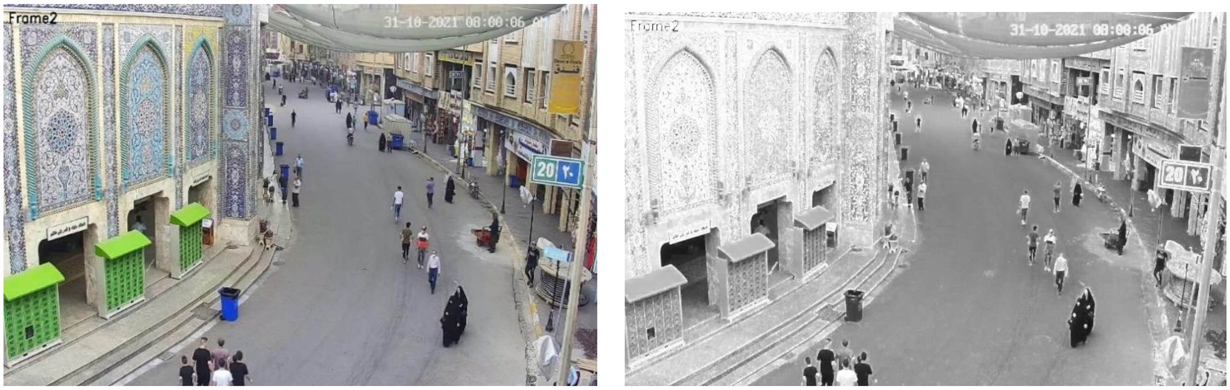
Figure(3.2): Where a. original frame and b. refer to frame after applying gaussian filter

Applying morphology operations if needed. Mathematically, morphology is an operation to eliminate imperfections in region of interest of image, and it is providing rich information pattern and image structure. In image processing concept, morphology is a method to analyze and describe the shape of digital image. Also, it is a way to describe the range of image processing approaches that cope with the shape of the features of an image.

In the proposed system, two morphological operations are used as present in figure(3.3) and figure(3.4) applying dilation and erosion on frames to improve the detection and tracking processes as well. Erosion and dilation are a set of suitable operations used image processing depending on shapes. It applies a structuring element as a filter to an input image and do some process to generate an output image such as:

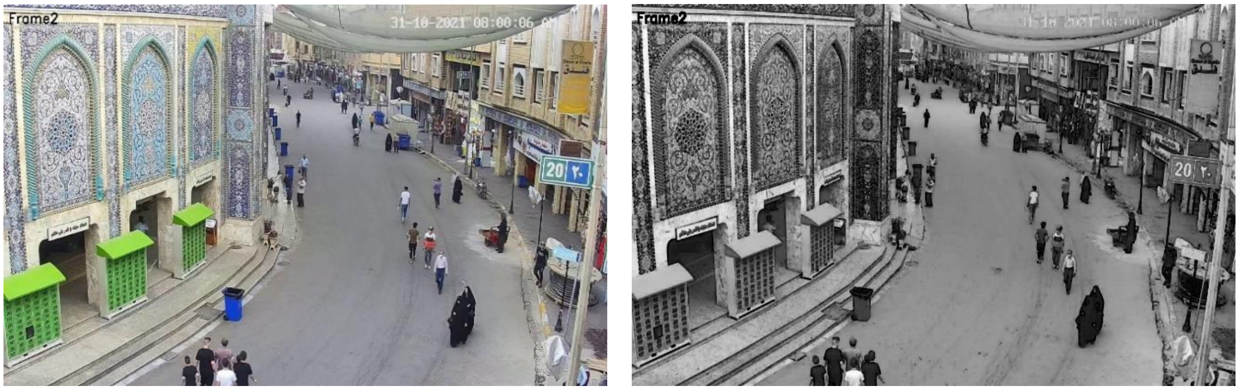
- Noise Removal
- Joining disparate elements and isolation of individual elements in an image.
- Discover any holes in an image

Figure(3.2) explains dilation operation apply on original image when figure (b) represents the result of dilation and  figure (c) represents erosion process briefly:



a b

Figure(3.3 ): Morphological operations a. original frame, b.  after apply dilation



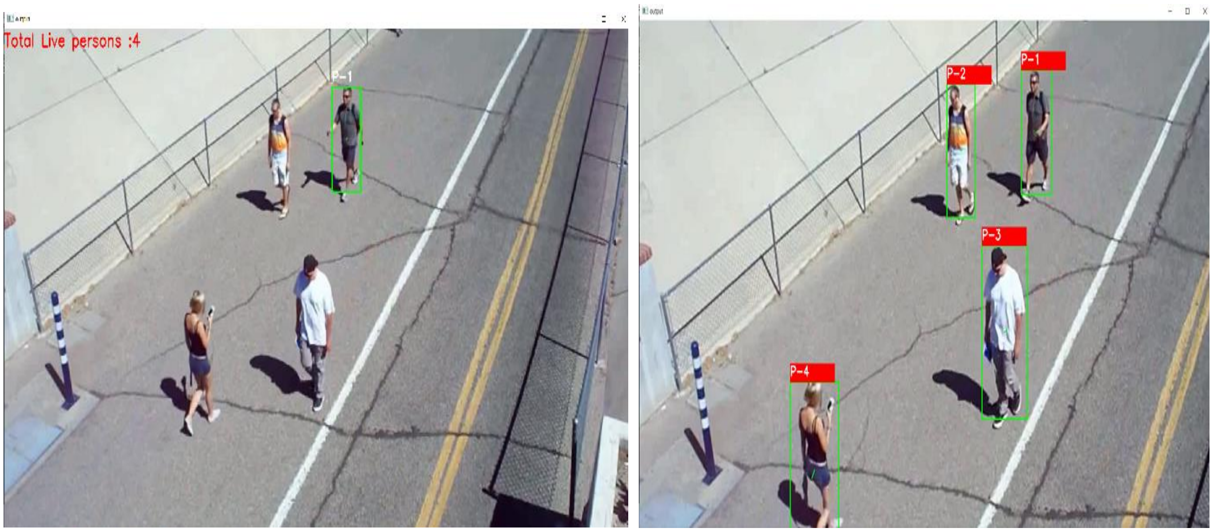
a b

Figure(3.4 ): Morphological operations a. original frame, b. after apply erosion

### 3.2.3 Object Detection Stage

After dividing video into a number of frames then the system performs preprocessing on each frame to increase accuracy of detection because the tracking depends on the result of detection. The next step is the target detection stage, detecting objects is a task that needs a complex solution, it means to process an input image (or a single frame from a sequence video) and respond with information about objects on the image and their position. These two tasks are called *localization and classification*. This stage will answer on what kind of objects are presented a given image and where exactly it located.

YOLO model is one of deep learning techniques and all details about YOLO network is explained in section (2.5.1). Speed and accuracy are the biggest advantage of using YOLO, it inputs the whole image in a one instance and predicts the location of coordinates and class probabilities for each box. The proposed system used YOLO3 for detection as shown in algorithm (3.1), each frame from sequence video input to YOLO3 algorithm to detect and classify the object in scene, classify the objects depend on number of features for shape and size and other features for describing the object, the type of object class in the proposed system is (person) as shown in Figure (3.5) when two frame represent the detection stage based YOLOv3.



a b

Figure( 3.5): Object detection by YOLOV3 a. single target, b. multi targets

## *Algorithm(3.1):YOLOV3_Object_Detection_Algorithm*

```
YOLOv3_Object_Detection_Algorithm
Input: video data , class(person)
Output: a list of bounding boxes along with the recognized class (person)
Begin
Step1:splitting video into number of frames(30 frames per second)
Step2:resize each frame into 416*416.
Step3:call_preprocessing function.
Step4:determine and detect region of interest (ROI)in each frame by using CNN by:
    ➢ Splitting each frame to regular grids and the coordinates assigned to all the grid cells.
    ➢ Check all cells to select which pixel has object then assign origin point of object.
    ➢ For each grid cell:
          •  predicts B boundary boxes, with confidence score for each box.
          • detects just one object regardless of the number of boxes.
          • predicts C conditional class probabilities (one per Class for the likeliness of
            the object class)
Step5: set single bounding box for each object by using intersection over union(IOU) and non
_max _  suppression.
   -Apply intersection over union by:
    ➢ Calculate area overlapping and area union between predicted bounding boxes with
      actual bounding box.
    ➢ Calculate IOU by dividing intersection area on union area for all bounding boxes.
    ➢ Order bounding boxes with their probabilities (IOU).
    ➢ Drop all the boxes that have probabilities less than or equal to pre-defined
      threshold(0.5 in this case).
    ➢ Pick the box with highest probability and take that as the output prediction.
    ➢ Drop the other boxes which have IOU more than the threshold value and less than
      IOU value for the output prediction boxes from the previous step.
Step6:Repeat step 2 until all the boxes are either taken as the output prediction or discarded
End
```

### 3.2.4 Object Tracking Stage

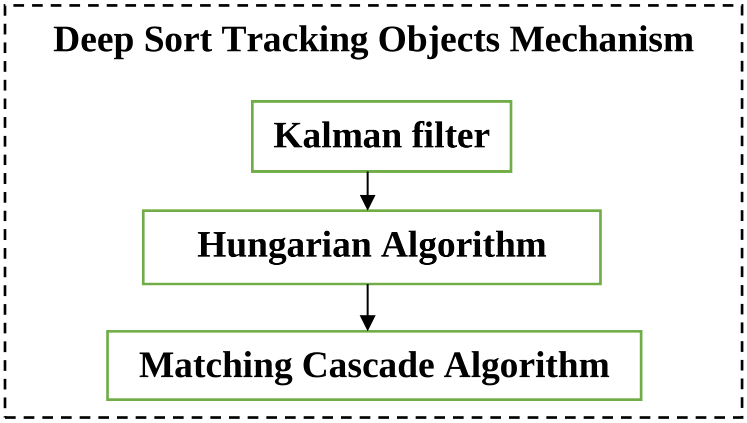
Tracking of any object in scene means how to keep the state and identification of object across frames. The tracking algorithm takes an initial set of object detections and assigns a unique (id) for each initial detection and tracks the detected objects inside bounding box with keeping unique (id) for each one for long time.

This task depends on deep sort algorithm which supported a Kalman filter by computing deep features for every bounding box and using the similarity(such as cosine metric) between deep features to improve association process between states across sequences frames.

In the proposed system, the detected objects are used for automatic object tracking by using extracted learning features from person to match an object in frame i-1 with candidate person in frame (i). The generated detections with deep sort algorithm can be used as a tracking tool to solve the assignment problem, this leads to increase the speed of tracking system reducing the number of unnecessary matches (less switch between id).

- **Deep Sort Mechanism**

The SORT is one of recent algorithms for object tracking, it is very fast, effective with occlusions and practical for object tracking. Simple Online and Realtime Tracking(deep sort) with a Deep Association Metric is an improved version of SORT to track the objects during long time of occlusion. The algorithm integrates appearance information of objects with results of predictions from Kalman filter to enhance associations between detections as present in figure(3.6). This tracking approach reduces the number of switches identities by 45% and reaches high performance and frame rate. Deep-Sort for target tracking in proposed system works in three stages.


Deep Sort Tracking Objects Mechanism
Kalman filter
Hungarian Algorithm
Matching Cascade Algorithm

Figure(3.6): Deep sort tracking Mechanism

### A. Kalman Filter(track handling and state estimation)

Firstly, define eight-variables as *(x, y, a ,h ,x’ ,y’ ,a’ and h’)* where *(x, y)* denote to the bounding box center, (*a)* refers to the aspect ratio and (*h*) represent the height of the image coordinates, the remained parameters are denoted to the velocities of the prior four variables. The next step is to use the Kalman filter based on constant velocity and model of linear observation, where it takes the parameters (*x, y, a and h*) of bounding box as the object state directly. In addition, for each track (t), it will count the number of frames for the last successful measure association ($a_t$). The track that will have a value greater than predefined max age of the tracker, it means the object leaving the scene and removed this track from the track group. Tracks that aren’t associated with measurement at the first three frames will be deleted. The algorithm (3.2) reviews the stage of deep sort, and the Figure (3.7) represents the tracking of objects using deep sort.

### B. Hungarian Algorithm(assignment problem)

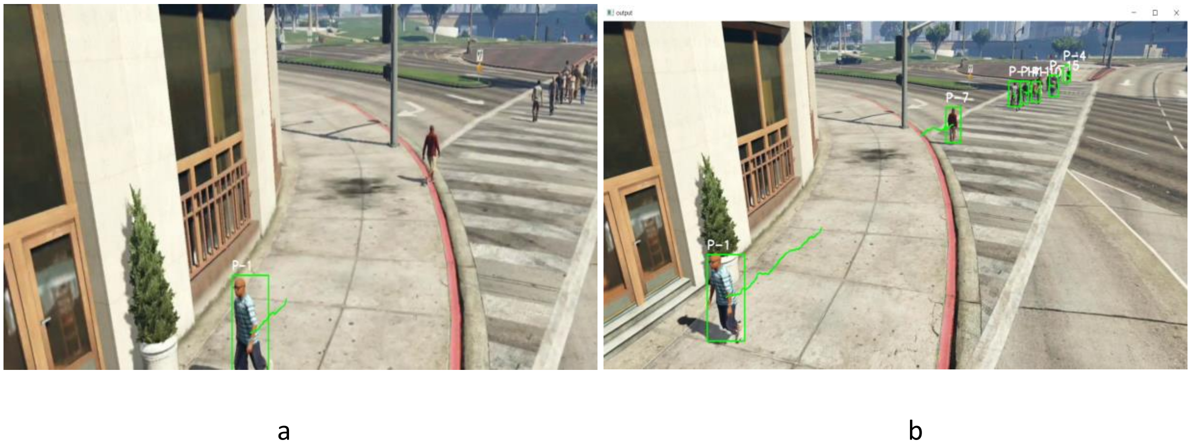
The second important step of Deep-Sort tracking algorithm is assignment problem in object tracking using the Hungarian algorithm, where it is integrated of appearance information and motion for moving targets depending on two metrics Mahalanobis distance and cosine similarity measures. Mahalanobis distance measure is an effective metric that is used to calculate the distance between a point and its predictions. Cosine similarity is metric to know whether two detected objects are same or similar by counting the similarity between two vectors (angle). Cosine angle between the features of the two vectors for detected objects is zero or close to zero, which means the detected persons are the same, even if his/her position has changed. If the cosine angle between two vectors would be 90° or close to 90° that means the detected objects are completely dissimilar. Hungarian metric used Mahalanobis distance to predict Kalman states space and new measurements that arrived, while it utilized cosine distance measure to make retrieve identities after long period of occlusion specifically when motion with less discriminative.

### C. Matching Cascade Algorithm

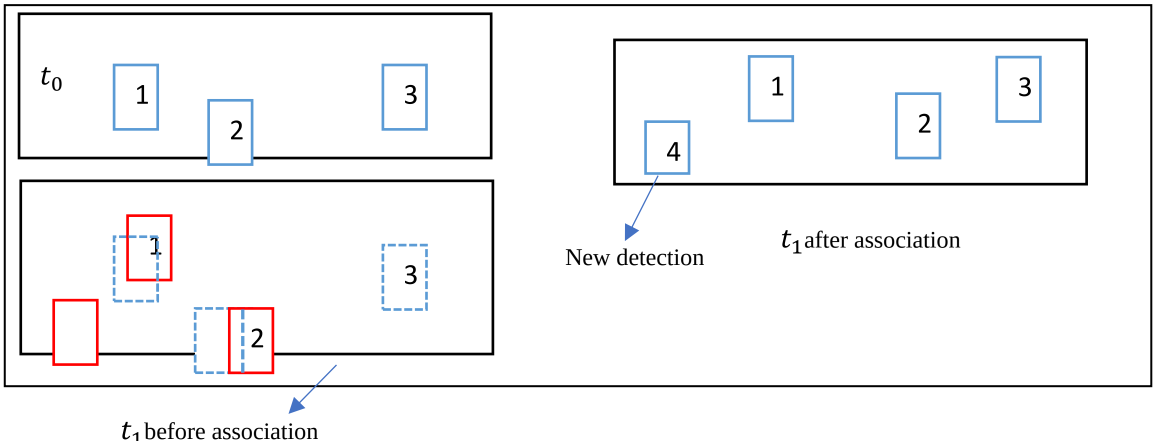
The third step is to apply matching cascade algorithm, which used intersection over union association for unexpected appearance changes; for example, partial occlusion causes and improve robustness versus initialization error. The final step in this tracking algorithm is a deep appearance descriptor. Using CNN architecture, release a large residual network and two convolution layers with six residual blocks. CNN architecture used 10 dense layers to compute global map features of 128 dimensions. Indeed, it used Deep-Sort tracking algorithm in this proposed system because it is robust through long time of occlusion with high accuracy.

A simple example to illustrate the association process by the Hungarian Algorithm is shown in Figure (3.8) At time t=0, there are three bounding boxes(1,2,3) are detected, the Hungarian Algorithm defines them at three new detections. Therefore, initialize a mean and a covariance with the values shows before, for every bounding box. At time t=1, these detections become tracking, the Kalman prediction used to predict where the boxes from time t=0 will be at time t=1. Then, these predicted boxes (dash boxes) are matched with the new boxes using the

Hungarian algorithm then, apply the update step to adjust matrices and make a better prediction at the next state.



Figure(3.7): Deep sort tracking objects algorithm a. tracking single object, b. tracking multi-objects.



Figure(3.8):Three frames with the association bounding boxes in scene using Hungarian algorithm.

### Algorithm(3.2):Deep Sort_ Object Tracking Algorithm

***Deep Sort_ Object Tracking Algorithm***

***Input:*** *objects detected (bounding box with id).*

***Output:*** *location of target.*

***Begin***

***Step1: Track Handling and State Estimation***

- *Defined on the eight-dimensional state space (x, y, a ,h ,x' ,y' ,a' ,h')that contains the bounding box center position (x, y), aspect ratio a , height h, and their respective velocities in image coordinates.*
- *Use Kalman filter algorithm with constant velocity motion and linear observation model by:*
    - ➢ *For each track k counts the number of frames since the last successful measurement association $a_K$.*
    - ➢ *If Tracks exceed a predefined maximum age (Amax), then object C leave the scene and delete from the track set.*

***Step2: Apply Hungarian algorithm for (Assignment Problem):*** *used two metrics to incorporate motion with information appearance*

- ➢ *Use the Mahalanobis distance measure between estimated Kalman states newly with arrived measurements.*
- ➢ *Use cosine similarity metric to retrieve identities after long-period occlusions when motion is less discriminative.*

***Step3: Matching Cascade***.

- ➢ *Apply intersection over union association to compute for abrupt appearance changes, such as, partial occlusion with static scene geometry, and to improve strongest versus initialization errors.*

***Step4: Deep Appearance Descriptor***

- ➢ *Using CNN architecture, to release a large residual network with two convolutional layers and 6 residual blocks.*
- ➢ *The general feature map of 128 dimensional is calculated 10 in dense layer.*
- ➢ *A final batch and the projects of normalization feature onto the unit hypersphere to be appropriate with metric of cosine appearance.*

***END***

### 3.2.5 Features Extraction

This stage is very important for object descriptor and analysis behavior of target motion in a scene. The proposed system extracted two types of features (learning features and trajectory features). In the first type learning features are extracted using CNN from each bounding box as a vector of features to match these features with other features from different classes or matching with other features of new bounding boxes at each state for identification. The second type is called trajectory features in this case the different features extracted from coordinates points of trajectory such as( position, direction and velocity).

#### 3.2.5.1 Learning Features Extraction

The learning features are extracted from bounding box of target which is detected by using YOLOv3 algorithm with many convolutions layer each layer as a descriptor for unique features. The proposed system depends on convolutional neural networks (CNN) with many kernels each one of these kernels will detect unique features when the image is input into the network, then transforms many times. Firstly, the image passes through many convolutional layers, in those layers, the network learns new and increasingly different and various features in its layers, then the image information is passed across the fully connected layers and turned into a prediction or classification.

#### 3.2.5.2 Trajectory Features Extraction

Initially, trajectory construction and normalized the trajectory points are two steps achieved before features extraction. The proposed system in the construction step presents a trajectory for each target's movement in the scene. Each bounding box has center point ($C_x$, $C_y$) in each frame, this point represents the location of the target in this state as presented in figure (3.9). All these points represent the coordinates of trajectory which present the behavior of a target's motion. The tracker is Connect the current point with the next point and continue across all frames to construct a unique trajectory for each target in the scene.

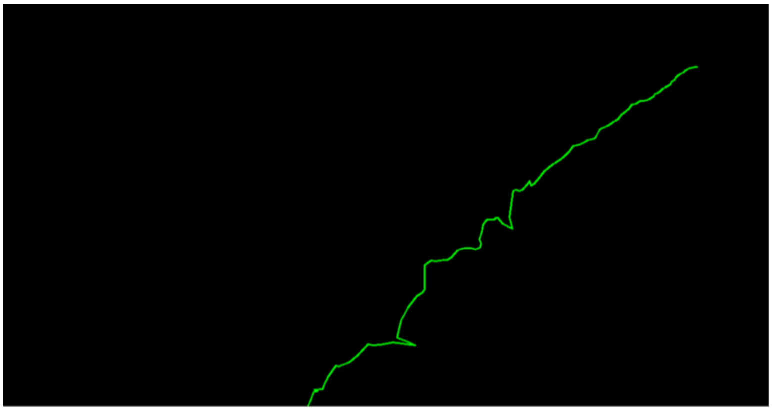


Figure (3.9 ): The trajectory construction of moving target in scene

The normalization of trajectory points step is very important to detect useful points from high numbers of trajectory points. Each trajectory represented by high numbers of location points which is present the change in a target's state with time.

Many of the points on the trajectory may have a small change in the moving of a person during the video, the system used normalization to remove these points depending on the sum of changes in the angle of direction when a target moves in a scene. After normalization and remove the less important points the remaining points of trajectory have good information about a change in direction and pose of a target, these features are used for training the LSTM model.

### *Algorithm(3.3):Reduction Trajectory Points*

```
Reduction _ Points _ Algorithm
Input: All Trajectory Points.
Output: Interested Points (points have change in direction).
Begin
Step 1: Current frame_ Current _ Point = ($X_{\text{center of bounding box}}$, $Y_{\text{center of bounding box}}$).
Step 2: Next frame _ Next _ Point = ($X_{\text{center of bounding box}}$, $Y_{\text{center of bounding box}}$).
Step 3: Calculate first _ Angle = $Tan^{-1}$ (current _ point, Next _ point).
Step 4: New _ point =Read _ New_ Point.
Step 5: Make  Current _ point value =next _ point value.
Step 6: Calculate second _ Angle = $Tan^{-1}$ (Current _ point, New _ Point)
Step 7: Result= (second angle- first angle)
Step 8: If Absolut( result) is more than practical value of threshold ( $20^{\circ}$) then take Current _
Point as a normalize point to be considered .
Else
Current _ Point removed.
Step 10: go to step 1 if not empty frame.
End
```

For predicting the new location of the target after an occluded place in a scene, the normalization of trajectory points reduces the number of points which improve the extracted features and time for processing.

The proposed system reads one point for each frame where the point represents the position of a target in the scene. At the first iteration, it reads the current point and it reads the next point in the second iteration, then calculate the angle called angle($\theta_1$) of direction between two points, then do swap to assign the next point as the current point and read the new point and recalculate the angle of direction called angle($\theta_2$) between these points then take the Absolut value for ($\theta_1, \theta_2$).

The next step applies the condition by comparting two angles if Absolut ($\theta_2 - \theta_1 > th$) which means current point is an important point and keep it in a new list, otherwise point will be deleted. The value of ($th = 20^{\circ}$) it estimated depending on try by error process. Lastly, the result list will contain the special points, this point is important and saved in list of interesting points as a normalized point. The algorithm (3.3) reviews the normalization stage.

The shape of trajectory which results from moving target (person) in the scene can be represented by various scenarios that may be direct line, curved line, or zigzag path depends on movement shape and orientation. It is construct from suitable centers points of bounding boxes across sequence of frames at each instance. The trajectory shape present the behavior for a target motion, that lead to extract many features from the target trajectory such as (Position, direction(Ɵ) and velocity(v)) such as shown in Table (3.1). The extracted features saved in (csv file) as actual dataset to represent the moving behavior of the target in a scene.

The feature of position is representing the difference value between two positions of person. The position feature can be used to calculate the amount of change in just x-coordinates or y-coordinates or both coordinates when target is moving. The change in coordinates can be used as a feature to predict the direction of motion in scene and when the target leaves the camera view to next camera across surveillance cameras network. for calculating the distance feature can be used Euclidean distance measure as show in section ( 2.7.2) equation (2.21).

The second feature (angle of direction) can be calculated for target at each location using the equation (2.22) in section(2.7.2). When the angle between two points that refer to the previous and current locations of target in view. The result value of angle of direction (Ɵ) used with the change in coordinates for (x) and (y) values to detect the orientation of moving target in video may be one of four directions ( East, West, North, South). The algorithm (3.4) reviews detected the direction of target in scene. To calculate the velocity feature by dividing the distance value on time(1/frame per second).

The features of all trajectories with values of positions for many moving persons in videos of dataset will save the features vectors in (file.csv) and will be used as (training and testing data) to construction LSTM model.

Table (3.1): The trajectory features of target

| ID | X | Y | Velocity | Direction |
|---|---|---|---|---|
| 1 | 1084 | 271 | 13.33 | North |
| 1 | 1087 | 269 | 10 | North |
| 1 | 1090 | 268 | 13.79 | West |
| 1 | 1096 | 267 | 10 | North |
| 1 | 1099 | 266 | 20 | West |
| 1 | 1105 | 267 | 17.24 | West |
| 1 | 1110 | 265 | 16.67 | West |
| 1 | 1120 | 267 | 33.33 | West |
| 1 | 1130 | 251 | 13.33 | East |
| 1 | 1135 | 252 | 22.58 | East |

***Algorithm(3.4):Detection direction of moving person***

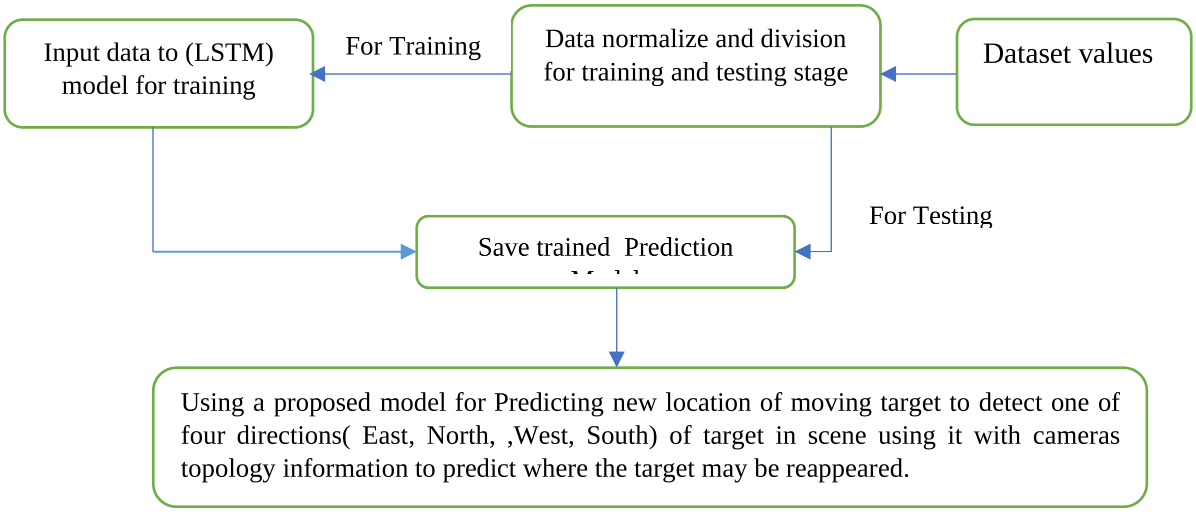
```
Detected  direction of moving person _ Algorithm
Input: center points Coordinate for bounding boxes .
Output: Direction of person in scene.
Begin
Step 1: Divided screen into four quarts(q1,q2,q3,q4) with center point(u ,v) for screen.
Step 2: for each center point of bounding box:
        • Current  Point = ($CX_{\text{center of bounding box,}}$ $CY_{\text{center of bounding box}}$) (current frame).
        • Next Point = ($cx_{\text{center of bounding box,}}$ $cy_{\text{center of bounding box}}$) (next frame).
        • Angle = $Tan^{-1}$ (current _ point, next _ point).
        • Calculate distance between two points for x-axis and y-axis by form ($d1=cx- CX$) and ($d2=cy- CY$).
        • Detect the position of person in any quart by check the position of current point and compare it with the respect to center point of screen (u ,v).
        • Calculate the direction of moving person in this quart based on the form:
              If $d1>d2$ and $CX > cx$ then the moving of person toward right.
              if $d1>d2$ and $CX < cx$ then the moving of person toward left.
              if $d1<d2$ and $CY>cy$ then the moving of person toward south.
              if $d1<d2$ and $CY<cy$ then the moving of person toward north.
        •  Assign Next  point into Current  point.
        •  End for
Step3:detect the direction of the person with Angle at each position.
End
```

### 3.2.6 Construction Prediction Model Using (LSTM)

The structure of Long Short-Term Memory (LSTM) is particularly good at learning historical patterns; therefore, they are particularly suitable for visual object tracking. This network is not very computationally expensive so it's possible to build very fast real-world trackers. Constructing robust prediction model based on LSTM used classical type of LSTM network, because there are several kinds of this model depending on the structure of network as present in Figure (3.10). The shape of data input to LSTM network needs to convert the data shape from one or two dimensions to three dimensions, because the network deals with data shape as 3-dimensions represented as (samples, time steps, features) as present in the Table (3.2).

Table (3.2): The summary representation of CONVLSTM2D prediction model

| | Layer (type) | Output shape | Params no. |
|---|---|---|---|
| | Lstm (LSTM) | (None,3,200) | 164800 |
| | lstm _1(LSTM) | (None,3,100) | 120400 |
| | lstm _2(LSTM) | (None,3,50) | 30200 |
| | dense(Dense) | (None,20) | 520 |
| | dense_1(Dense) | (None,10) | 210 |
| | dense_2(Dense) | (None,4) | 55 |
| Total params: | | 323,785 | |
| Trainable params: | | 323,785 | |
| Non-trainable params: | | 0 | |


Input data to (LSTM) model for training
For Training
Data normalize and division for training and testing stage
Dataset values
For Testing
Save trained Prediction
Using a proposed model for Predicting new location of moving target to detect one of four directions( East, North, ,West, South) of target in scene using it with cameras topology information to predict where the target may be reappeared.

Figure(3.10): The steps for Building LSTM Model

Constructing LSTM model for predicting new state for occluded moving targets in scene depending on trajectory features consists of three stages (training, testing and prediction stage).

### A. Training Stage of LSTM Model

This stage represents how can training LSTM network on actual values from different scenario obtains from six videos, each video contains many places when target occluded and present specific trajectory features (position, direction and velocity) for moving target in scene. Construction robust model for predicted new state for moving target required extracted features (trajectory features) and saved in (file.csv) as dataset.

The dataset is obtained from the different scenarios of persons motion in scene and save the interesting points of coordinates for each trajectory. The total number of samples in file.csv is 16000 points for x-coordinate and y-coordinate obtains from MAT dataset and another file.csv contains 12400 points for x-coordinate and y-coordinate obtains from *Abbasid shrine in Karbala city* dataset .

Initially the dataset in (file.csv) is divided into two parts as (training data) contains 80% of total features vectors of dataset and (testing data) with remaining vectors. Training data have many vectors of specific features for many targets moving in different direction and walking through occluded places, some of these vectors present the actual values which represent the behavior of moving target before occluded and after reappears from occlusion. The training process for LSTM with structures of network includes number of layers each layer has number of neurons used to practice the layers of network, based on previous information (before occluded target) of trajectory and remember it by forgetting cell in LSTM to predict the state of target after occlusion with one of four directions of moving person in scene like (Right, left, North, South).

### B. Testing Stage of LSTM Model

The next stage to building prediction model is testing and evaluating the performance of model by inputting the testing data into model and computing the results of new states of moving target in scene. In this stage calculate the performance of model by checking the values of new features predicted by model with actual value after person reappear from occluded area, if the results were robust that means the model is able to predict the new state of moving object in scene, that will support the deep sort Algorithm to solve the assignment problem between detections and new predictions when target reappear in scene after many steps of time when person was occluded.

### C. Prediction Stage of LSTM Model

After completing the activation of model, the proposed system is calling the model to predict new state for the moving person in scene. The LSTM model depends on lookback for previous information of specific time steps to predict the new state. Number of time steps selected during constructing model using try by error method in this case is (three steps).

In proposed system, the LSTM prediction can help to improve detection and tracking tasks depend on trajectory features through two situations, the first situation called short-term occlusion when the target occluded and disappears more than three frames in scene for specific short time by another object may be person or wall ….etc. The second situation is called long-term occlusion when the target leaves the boundary of FOV, in this case there are high change in target state such as illumination and pose where target disappear for long period from the scene.

The proposed system after each tracking step checks the target if leaves view or not, if the result test was not leaving view, the system sends the trajectory feature vector to LSTM model to predict new state for moving target in scene based on memory cell of LSTM to support deep sort in tracking process to improve association between states.

On the other hand, if the target left view, in this case the system used the previous predicted features vector of target with its orientation to select which next camera the target may reappear depending on the spatial relations between cameras in the priority table, then send the learning features vector for next camera to complete matching process between the target with all objects appear in the next camera for doing target re-identification.

Depending on the special-temporal relation among cameras to predict the direction of the target across a blind area, the process of prediction will continue for target when it leaves the boundary of view for the camera and depending on the distance between the current camera with its neighbors located in that direction to re-identify the target which may be reappear as illustrated at Algorithm (3.5).

***Algorithm(3.5):Algorithm of LSTM model***

```
Algorithm of LSTM model
Input: input csv file of trajectories points, (number of layers), number of epochs, learning rate(ή), pitch size and lookback steps.
Output: learned prediction model using LSTM.
Begin
Step1: preprocessing csv file of data frame.
Step4: divided data frame to 80%  for training and 20% for testing data.
Step5: input training data to RNN(LSTM) for training model.
Step2:intinlazed weights for each layer(W1, W2).
Step3: for each epoch:
        For each cell state:
          • forget gate in cell passes information of previous hidden state with current input through sigmoid function to decide what information should be thrown away or kept.
          • Input gate is updating the cell state,  passes the previous hidden state and current input into a sigmoid function. The result will decide which information is important to keep from the tanh output.
          • Calculate the cell state. the cell state gets pointwise multiplied by forgets vector. Then it take the output from the input gate and do a pointwise addition which updates the cell state to new values relevant.
          • Output gate decides what the next hidden state should be. Remember that the hidden state contains information on previous inputs. The hidden state is also used for predictions.
          • Calculated mean square error between predicted values with actual values.
          • If result is less than threshold value, return.
Step6:After specific number of epochs for training LSTM net, using test data to evaluate model.
Step7:Calculate the accuracy and losses of prediction model.
END
```

### 3.2.7 Object Re-identification Stage

Re-identification stage represents the last stage in  the proposed system. The stage starts when the target leaves (FOV) for current camera and enters the boundary of blind area between non-overlapping cameras network, then the proposed system used re-identification model with previous target features to predict where the target may reappear to do matching between target features with all person's features moving in next camera scene to re-identify the same target in next camera as presented in algorithm(3.6). This stage needs many steps for constructing re-identification model based on CNN model for features matching for target blob.

## 3.2.7.1Create Object Re-identification Model

This stage represented how to build the robust model for re-identifying the moving target when it moves across camera network. There are many steps required to create the model such as building dataset, training and testing model. The next sections will explain each step of building model.

**A. Building Dataset**

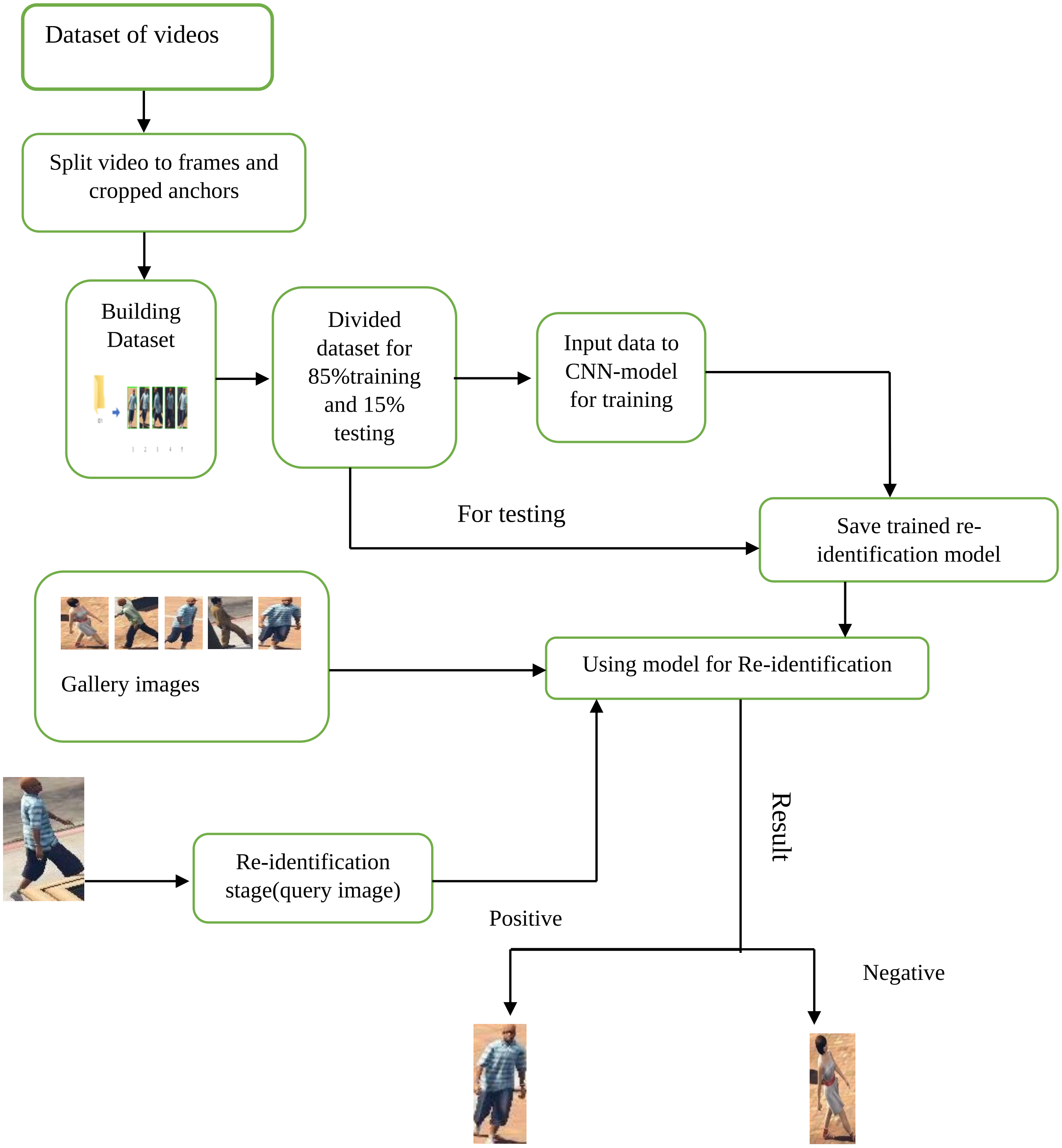
This stage is necessary for target re-identification to build suitable dataset for training and evaluated model for reidentification across cameras network. First step is read videos from MTA- dataset and detected each person by cropped many states of bounding boxes for same person and saved these cropped images in label file as shown in Figure (3.11). Each person has unique file contains many images with different scenarios of state for that person. The same steps apply on dataset of *Abbasid shrine in Karbala city.* After preparing dataset can build re-identification model.

Figure(3.11): The samples of cropped images from dataset which building for re-identification.

## B. Training Stage of RIED Model

Training stage is the first step for building model based on dataset created in the previous step. The dataset will be divided into two parts, first part contains 85% of total samples and 15% for second part for constructed model. To training CNN for recognize image and classify image as a positive image when target image matching another image from gallery images or as a negative image if no matching any image from gallery images. The convolution neural network consists of many layers for convolution, pooling, net fully connected layer and dens layer, each layer uses the weights which saved from previous iteration to obtain the output of this layer to be input to the next layer and continues until the input to last layer for the classification step to make the decision and produce the output of the model such as presented in in Figure (3.13) and Table(3.3). The target image will pass across CNN to extracted features vector, the step implements CNN model to extract and track the salient features of the input sample. Next step is pass query image from gallery to CNN to extracted features vector, the feature maps that outputs from the feature extraction step is Flatten before inters to the last step. At end the model will matching between two vectors for two images for identification.

This process is repeated to matching features of target image with all gallery images for comparative to training model to select the image may be similar as present in Figure (3.12).


Dataset of videos
Split video to frames and cropped anchors
Building Dataset
Divided dataset for 85%training and 15% testing
Input data to CNN-model for training
For testing
Save trained re-identification model
Gallery images
Using model for Re-identification
Result
Re-identification stage(query image)
Positive
Negative

Figure(3.12): The steps for building a Re-identification Model

**Algorithm(3.6):Human Re-identification algorithm**

```
Human Re-identification algorithm

Input: target prob, labeled dataset of gallery images(all instances of same person have same label), number of layers, number of epochs, learning rate, pitch size.
Output: model construction for person re-identification.
Begin
Step1: divided data set for 85%training and 15% testing stage.

Step2:training stage.
    • For each epoch:
        • Read new target prob and resize to 224*224
        • Pass image of target to CNN2D to extracted features vector.
        • For all images in gallery:
            • Pass query image from gallery to CNN2D to extracted features vector.
            • Using cosine metric to compute similarity between features vector of target image with features vector of query images from gallery to retrieval.
            • If similarity score more than specific threshold two images is similar.
              Else
              Two image not similar, return.
    • Return
Step2:select max similarity value
Step3:evaluated model by comparting each image from test data with all gallery images for retrieval.
Step4:complete building model.
End
```

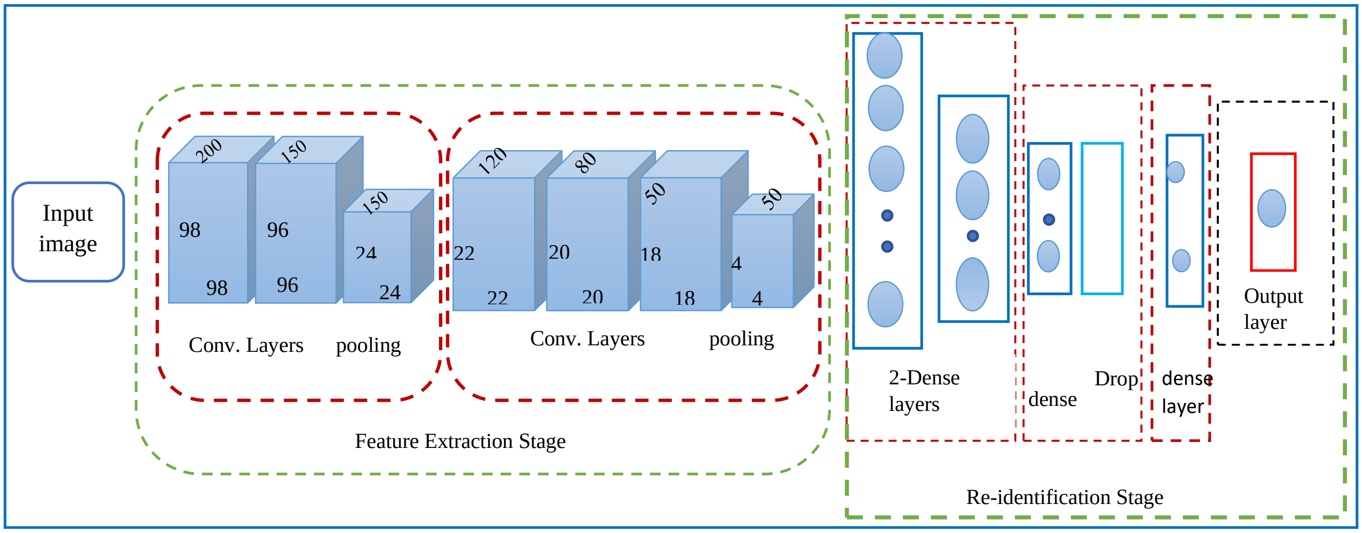


Figure (3.13): The Architecture of re-identification model

Table (3.3): The summary representation of re-identification model

| Block no. | Layer (type) | Output shape | Params no. |
|---|---|---|---|
| 1 | conv2d (Conv2D) | (None,98,98,200) | 5600 |
| | conv2d-1 (Conv2D) | (None,96,96,150) | 270150 |
| | max-pooling2d (Maxpooling2D) | (None,24,24,150) | 0 |
| 2 | conv2d-2 (Conv2D) | (None,22,22,120) | 162120 |
| | conv2d-3 (Conv2D) | (None,20,20,80) | 86480 |
| | conv2d-4 (Conv2D) | (None,18,18,50) | 36050 |
| | max_pooling2d-1 (Maxpooling2D) | (None,4,4,50) | 0 |
| 3 | flatten (Flatten) | (None,800) | 0 |
| | dense_(Dense) | (None,120) | 96120 |
| | dense-1 (Dense) | (None,100) | 12100 |
| 4 | dense-2 (Dense) | (None,50) | 5050 |
| | dropout (Dropout) | (None,50) | 0 |
| 5 | dense-3 (Dense) | (None,4) | 204 |
| Total params: | | 673,874 | |
| Trainable params: | | 673,874 | |
| Non-trainable params: | | 0 | |

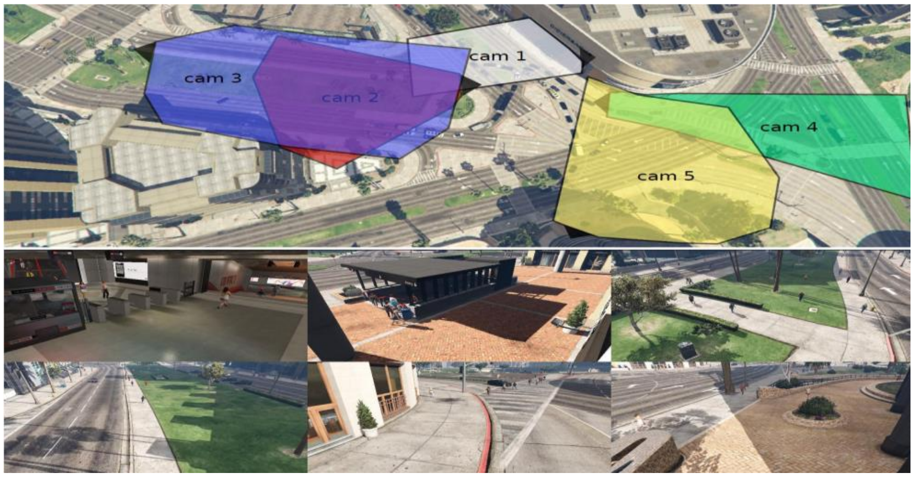
Learning phase includes the (training and validation process), trains the proposed model on all the training dataset for learning on the class of samples by input this samples with its corresponding labels to model. Adjusting the saved weights corresponding to each layer through the training process based on computing the total difference value between the actual and desired output of the network. The weights updating is to go on until the convergence of network reaching to the minimum error; based on using mean square error and loss function. After completing the training step and the network is stable, the validation process implemented to evaluate the model (the validation data is considered part of the training data used 20% from total training data). The weights which result from the training process using to prove the performance and accuracy of the trained proposed model and to decide the model preparation to predict classification with new data. The output shape (None,98,98,200), where None refer to default patch size of data and 98 refer to size of output frame then number of 200 refer to number of filters.

### B. Testing Stage of RIED Model

In this stage, the trained proposed model implements to predict the output class on unseen (untrained samples) inputs and represents the application of the system. The part test of images from dataset will be input to model to check the performance of model to re-identify the target image based on matching two images to decide if these images is same or different. When high accuracy refers to better performance for model.

### 3.2.7.2 Spatial and Temporal Relation of Cameras Network

The relation between cameras(spatial and temporal) represents the method of connection between multiple cameras as shown in Tables(3.4) and (3.5). Figure (3.14) show these relations in network.



Figure(3.14): The topology of network for  MTA dataset

Table (3.4): The topology of  camera network each row in table refers to the camera and its neighbors in each side (east, west, south, north), (MTA dataset).

| CAMERA NUMBER | EAST SIDE | WEST SIDE | SOUTH SIDE | NORTH SIDE |
|---|---|---|---|---|
| C0 | C4,C5 | C1 | ------ | ----- |
| C1 | C0 | C2,C3 | C4,C5 | ----- |
| C2 | C1 | C3 | C4,C5 | ----- |
| C3 | ---- | ----- | C1,C0<br>C4,C5 | ----- |
| C4 | C5 | C0 | ---- | C1,C2<br>C3 |
| C5 | ----- | C4 | ----- | C1,C2<br>C3 |

Table (3.5): The relation distances and time between cameras in network (MTA Dataset)

| CAMERAS NUMBER | DISTANCE BETWEEN CAMERAS | TIME BETWEEN CAMERAS |
|---|---|---|
| C1 ↔ C0 | 144M | 96 sec |
| C1 ↔ C4 | 250M | 166 sec |
| C1 ↔ C5 | 200M | 133 sec |
| C2 ↔ C1 | 125M | 83 sec |
| C2 ↔ C3 | 120M | 80 sec |
| C2 ↔ C0 | 275M | 183 sec |
| C4 ↔ C3 | 200M | 133 sec |
| C2 ↔ C4 | 375M | 250 sec |
| C2 ↔ C5 | 285M | 100 sec |
| C0 ↔ C4 | 150M | 100 sec |
| C0 ↔ C5 | 200M | 133sec |
| C4 ↔ C5 | 120M | 80 sec |

Table (3.6): The topology of camera network each row in table refers to the camera and its neighbors in each side (East, West, South, North), ( *Abbasid shrine in Karbala city* dataset).

| CAMERA NUMBER | EAST SIDE | WEST SIDE | SOUTH SIDE | NORTH SIDE |
|---|---|---|---|---|
| C0 | ----- | C1,C2 | ---- | ----- |
| C1 | ----- | C2 | C3 | C0 |
| C2 | C1 | C0,C3 | ---- | ----- |
| C3 | C4 | C1 | ----- | ----- |
| C4 | C5 | C3 | ---- | ------ |
| C5 | ----- | ----- | ----- | C4 |

Table (3.7): The relation distances and time between cameras in network, (*Abbasid shrine in Karbala city* dataset).

| CAMERAS NUMBER | DISTANCE BETWEEN CAMERAS | TIME BETWEEN CAMERAS |
|---|---|---|
| C1 ⟷ C0 | 288M | 360 sec |
| C1 ⟷ C4 | 115M | 75 sec |
| C1 ⟷ C5 | 200M | 133 sec |
| C2 ⟷ C1 | 80M | 53 sec |
| C2 ⟷ C0 | 110M | 73 sec |
| C2 ⟷ C4 | 140M | 93 sec |
| C2 ⟷ C5 | 280M | 185 sec |
| C2 ⟷ C3 | 115M | 75 sec |
| C0 ⟷ C4 | 150M | 100 sec |
| C0 ⟷ C5 | 230M | 153sec |
| C4 ⟷ C5 | 80M | 53 sec |

### 3.2.7.3 The Adjacency Matrix

Adjacency matrixes represent the spatial relationship between cameras in network. Where each number in matrix refers to the weight (distance) between two cameras. Depending on this matrix and with priority (minimum distance), the system selects the candidate camera to search about target in that scene. The figure (3.12) refers to spatial distance between cameras networks.

### 3.2.8 The Occlusion

Occlusion represents any obstacle that may be blocking the target, different occlusions may appear such as wall , tree or building…etc. Disappearance the target from view is considered as one of the difficult challenges in object tracking. Reidentification stage present how can re-identify the target after period time when the target disappears as result for occlusion, can classify the occlusion based on the re-identification process to two types short and long occlusion.

Table (3.8): **Adjacency Matrix** contains the distances between cameras, (MTA dataset).

| | $C0$ | $C1$ | $C2$ | $C3$ | $C4$ | $C5$ |
|---|---|---|---|---|---|---|
| $C0$ | 0 | 144 | 275 | 0 | 150 | 200 |
| $C1$ | 144 | 0 | 125 | 0 | 250 | 200 |
| $C2$ | 275 | 125 | 0 | 0 | 375 | 285 |
| $C3$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $C4$ | 150 | 250 | 375 | 0 | 0 | 120 |
| $C5$ | 200 | 200 | 285 | 0 | 120 | 0 |

Table (3.9): **Adjacency Matrix** contains the distances between cameras, (*Abbasid shrine in Karbala city* dataset).

| | $C0$ | $C1$ | $C2$ | $C3$ | $C4$ | $C5$ |
|---|---|---|---|---|---|---|
| $C0$ | 0 | 288 | 110 | 0 | 150 | 230 |
| $C1$ | 288 | 0 | 80 | 0 | 115 | 200 |
| $C2$ | 110 | 80 | 0 | 0 | 140 | 280 |
| $C3$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $C4$ | 150 | 115 | 140 | 0 | 0 | 80 |
| $C5$ | 230 | 200 | 280 | 0 | 80 | 0 |

### 3.2.8.1 Re-identification with Short Occlusion

Short occlusion term used for describing the target state when the moving person suddenly disappears for a short time (specific numbers of frames) in scene, such a person occluded by another one or by any obstacle causes disappear the target from view for specific moments. Due to the short-term occlusion that appear in scene for short time in single static camera, that mean little change in illumination or pose of target and using Deep sort algorithm with LSTM model for prediction new location and re-tracking the same target after it reappear in scene. Two models corporative to get on a correct association for target states in scene based on its features. The proposed system depends on deep sort and LSTM to solve the association problem for target states across short-term occlusion for re-identification.

#### 3.2.8.2 Re-identification with Long Occlusion

Long occlusion term represents the targe state when the person leaves the boundary of camera view and disappears for long time occluded during the blind area between cameras as shown in Figure(3.12). The proposed system used the structure of LSTM model to predict orientation of moving target and send the previous features vector of target to the next camera which selected depending on the priority tables of cameras topology. The priority table shows the special relationships between cameras of network as present in Table(3.5) and (3.7).

The proposed system depends on the priority index to candidate the next camera, can illustrate this process in example when the target leaves the boundary of view for camera number one then moving toward the left side, the system detected the number of neighbor’s cameras in the left side depends on the topology table then the system candidate the camera which has high priority (the short distance or the nearest camera to current camera). The system calculates the total time which requires for moving target to reach next camera.

The total time is equal to summation of time between two cameras (current camera and next camera) and the time detected with first twenty frames of next camera video. The first twenty frames is estimation time which need to complete a matching task between the features vector for target with features for all persons in candidate camera for re-identification then check if the result was matching the system assign same (id) for target and continue of tracking in next camera, else if the result was no matching the system repeat search process on other camera in the left side based on priority list to candidate another camera which have second degree in priority list to check it if the target may be go to this camera or no and repeat same steps for matching and re-identification.

## 3.9 Summary

This chapter states and identifies each stage of the diagram of the proposed SCNS based on LSTM, and CNN-model for re-identification with other algorithms such as yolov3 and deep sort which is used to succeed the system. All used techniques and proposed algorithms are mentioned and explained in detail with some abstracted results of scenes' events to clarifying the work, the experimental study that ware used two types of datasets contains different scenarios for person occluded with many challenges and to prove the idea of dissertation.

# CHAPTER FOUR

# EXPERIMENTAL RESULTS AND DISCUSSIONS

## 4.1 Introduction

This chapter shows and identifies the results and discussion of the proposed SCNS stages in general starting with the proposed algorithms to the main stages of system, the evaluation methodologies of the research, the system test, the performance measures used for evaluation, the evaluation results of the proposed system with discussions, the general evaluation of the SCNS and/or its algorithms individually based on the ground truth and evaluation criteria. The effectiveness of the proposed system illustrated in the previous chapter is tested with different parameters values and the results of implementing will be analyzed in this chapter.

It is worth to mention that the evaluation of the proposed system stages is implemented separately. The dataset contains several video categories from six cameras distributed in nonoverlapping topology in environment. The categories are selected to evaluate the proposed system of person re-identification across camera network includes some challenges such as (change illumination , pose , short occlusion and long occlusion when person moving from camera to another across blind area). A public dataset is applied as a case study to determine the behavior of each model. The datasets that are used with the proposed system, as well as the system requirements are presented in the next sections. Other sections describe the results obtained from each stage of the proposed system.

## 4.2 System Requirement

First, to perform deep learning and machine learning on any dataset spatially when the type of dataset is videos , the software/program involves a computer system powerful enough to handle the computing power necessary. Therefore, the proposed system is implementing using the following:

- **Hardware:**

1. Processor Intel i7, Ram 8 GB, Storage 500 GB, Freq. 2.2 GHz.

2. Central Processing Unit (CPU): Intel(R) Core(TM) i7-10750H CPU.

3. RAM: Samsung 16 GB.

4. Hard Disk: 2 TB + 256 GB.

5. Graphics Processing Unit (GPU): NVIDIA GeForce GTX 1060 Ti 4 GB

- **Operating System:** Windows 10 64 bit.

- **Programming Language:** Python.

## 4.3 Evaluation Measures

Precision and Recall metrics are two popular measures used for objective evaluation in recent detection, tracking, prediction and re-identification techniques. The precision measure is especially used to specify the relation of appearance with spatial properties between the moving objects and the change in object state as the small spread pixels which has noisy details (not objects) or predicted new position has distance about actual location. These measures have good criteria for describing the results in terms of frames, bounding boxes, trajectories , …etc. These criteria refer to parameters as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) bounding box . On the opposite, subjective evaluation is based on human vision observations of resulted frames with bounding boxes. For such a tested frame, TP is the bounding boxes that represent a moving person's body truly and considered finally as same person with same (id) detection correctly across occlusion while FP is the bounding boxes that represent a moving person's body falsely (it is switch id for another object not same person) and considered finally as same person with same (id) detected by mistake. The parameter TN is the bounding boxes that represent wrong person detected truly and considered finally as assign different (id) correctly while FN is the bounding boxes that represent a wrong person detection falsely and considered finally as person by mistake as shown in Figure (4.1).

## 4.4 Evaluation of Proposed System Stages

The proposed system contains a number of stages each stage comprises specific algorithms to complete this task, evaluate the performance for the proposed system required evaluate each stage and determine the efficiency for each algorithm depending on many of metrics explained in section (2.12), the evaluation results of the proposed system with discussions. The general evaluation of the SCNS and/or its algorithms individually based on the ground truth and evaluation criteria.

### 4.4.1 Evaluation of Object Detection Stage

Detection stage is the second process after preprocessing task in steps of the proposed system. Evaluation for this stage means applying several metrics to determine the performance of yolov3 algorithm to detect and classify the moving objects in scene. Depending on IOU algorithm for calculate the ratio of overlapping between two bounding boxes(predicted bounding box, truth bounding box), when truth box represents the fit dimensions of box around the boundary of object body while the predicted bounding box represents the output of yolo algorithm when it detected the object. The ratio of IOU is obtained from dividing the intersection of area by union area of two bounding boxes (predicted and truth), acceptance the result value depends on the threshold value, if the result value was more than the threshold value, this value is considered acceptable, and the result of detection is correct but if the result value was less than threshold value, this value is considered mistake detection in that position. The predetermine threshold used in the detection algorithm are chosen depending on the ratio of overlapping between boxes.

#### 4.4.1.1 Objective Evaluation Results

The algorithm of YOLOv3 and the others are implemented under core i7, 8 GB memory, and OpenCV version (4.1.25) library of python (3.8) on PyCharm editor and the video’s dimensions(1280×720) for *Abbasid shrine in Karbala* and (1920×1080) for MAT dataset.

For detection stage the parameters of(TP,FP,FN,TN) means:

- **TP** is referred to the bounding boxes that represent a moving person's body (Foreground) truly and considered finally as same person(IOU > 0.5).
- **FP** is the bounding boxes that represents a moving person's body falsely when (IOU=<0.5).
- **FN** mean missing detection for target.
- **TN** refers to bounding box that may appear on background space or another object.

For more clarification, the evaluation results of detection stage are organized by choosing (838) frames as in Table(4.1) and present the results in the term of Performance Metrics( confusion matrix) in Table(4.2) based on (precision and recall).

| | Total frames 838 | | Predicted boxes ( Left up and right down points) | | Ground truth boxes ( Left up and right down points) | | IOU | Foreground detection | Wrong detection | Missing detection | Background detection |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Frame no | ID | X1,Y1 | X2,Y2 | X1,Y1 | X2,Y2 | IOU>0.5 | TP | FP | FN | TN |
| C0 V1 | 3 | 1 | 975,460 | 1008,545 | 970,455 | 1010,553 | 0.72 | 1 | 0 | 0 | 0 |
| | 15 | 1 | 975,460 | 1008,547 | 971,455 | 1008,555 | 0.83 | 1 | 0 | 0 | 0 |
| | 22 | 1 | ------ | ------- | 971,468 | 1007,566 | --- | 0 | 0 | 1 | 0 |
| | 25 | 1 | 972,460 | 1009,570 | 972,475 | 1010,572 | 0.86 | 1 | 0 | 0 | 0 |
| | 143 | 1 | 895,590 | 940,708 | 890,590 | 940,711 | 0.88 | 1 | 0 | 0 | 0 |
| | 153 | 1 | 883,605 | 930,711 | 880,630 | 930,719 | 0.86 | | | | |
| | 167 | 1 | 868,620 | 910,720 | 868,621 | 911,721 | 0.95 | | | | |
| | | | | | | | | | | | |
| C1 V1 | 4 | 1 | 89,316 | 180,422 | 90,315 | 170,427 | 0.84 | 1 | 0 | 0 | 0 |
| | 5 | 1 | 109,321 | 195,423 | 111,315 | 179,429 | 0.72 | 1 | 0 | 0 | 0 |
| | 6 | 1 | 119,321 | 205,422 | 119,312 | 206,425 | 0.89 | 1 | 0 | 0 | 0 |
| | 7 | 1 | 122,321 | 215,425 | 121,315 | 210,425 | 0.89 | 1 | 0 | 0 | 0 |
| | 8 | 1 | 130,320 | 220,425 | 125,310 | 219,425 | 0.84 | 1 | 0 | 0 | 0 |
| | 18 | 1 | 199,275 | 315,410 | 199,290 | 285,405 | 0.63 | 1 | 0 | 0 | 0 |
| | 27 | 1 | -------- | --------- | 1011, 201 | 998, 385 | ---- | 0 | 0 | 1 | 0 |
| | 42 | 1 | 319,249 | 409,370 | 350,250 | 390,368 | 0.42 | 0 | 1 | 0 | 0 |
| | 43 | 1 | 333,255 | 421,375 | 360,260 | 412,371 | 0.55 | 1 | 0 | 0 | 0 |
| | | | | | | | | | | | |

Table (4.1): The samples of evaluation detection stage using YOLOv3

Table (4.2): The confusion matrix for actual and predicted detection

| Total frame detected = 838 | | Actual detection | |
|---|---|---|---|
| Prediction detection | | Positive | Negative |
| | Positive | TP=743 | FP=7 |
| | Negative | FN=88 | TN=0 |

**Accuracy** =(TP+TN)/(TP+TN+FP+FN)=743/838=**0.89%**

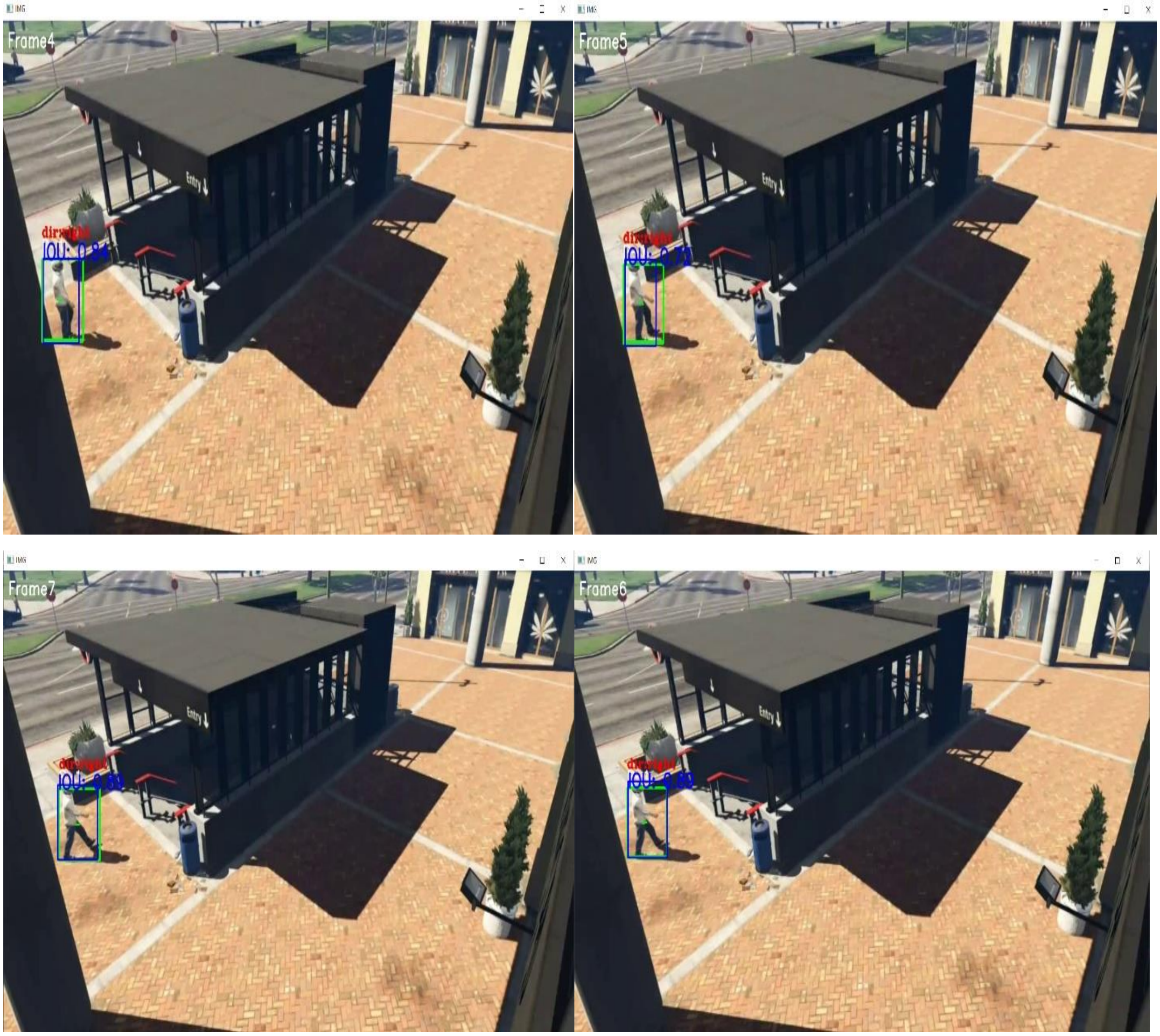
**Precision** =TP/(TP+FP)=743/(743+7)=**0.99%**

**Recall** =TP/(TP+FN)=743/(743+88)=**0.90%**

**F _ Score** =2*(precision*recall / Precision + Recall)=2(0.99*0.9/0.99+0.9)=**0.94%**

#### 4.4.1.2 Subjective Evaluation Results

The objective measure in the practical side needs to be more explained by a fair subjective evaluation that can help in understanding and trusting the outcomes as compared to the ground-truth frames as in Figures (4.1) and (4.2). The detection algorithm of YOLOv3 has highest scores for target detection during sequence of frames for two datasets (MTA, *Abbasid shrine in Karbala city* )that reflect a promising and good place in future use over the others for tracking systems.



To be continued

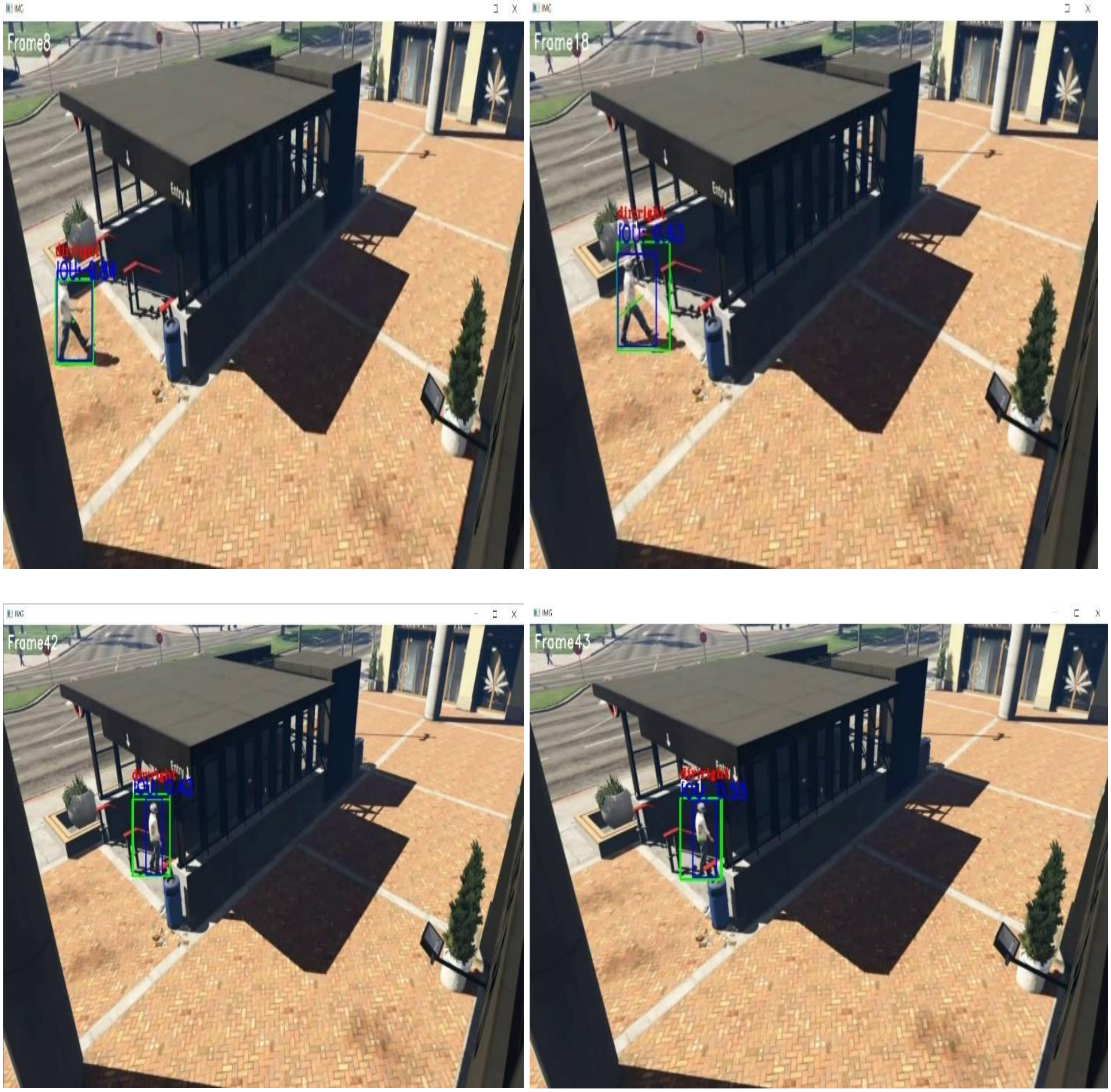
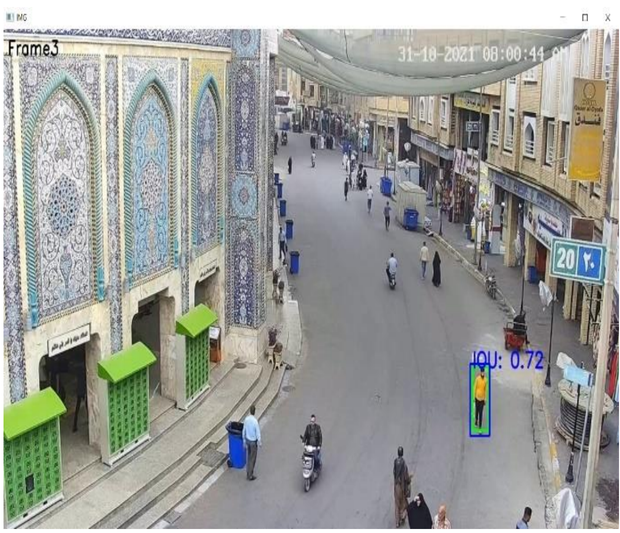
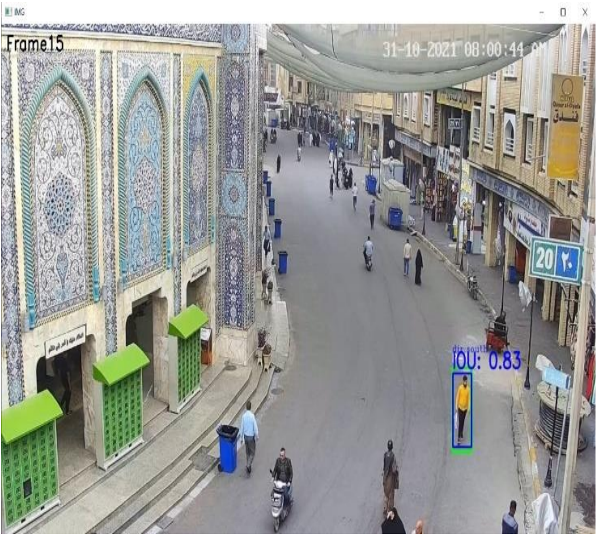
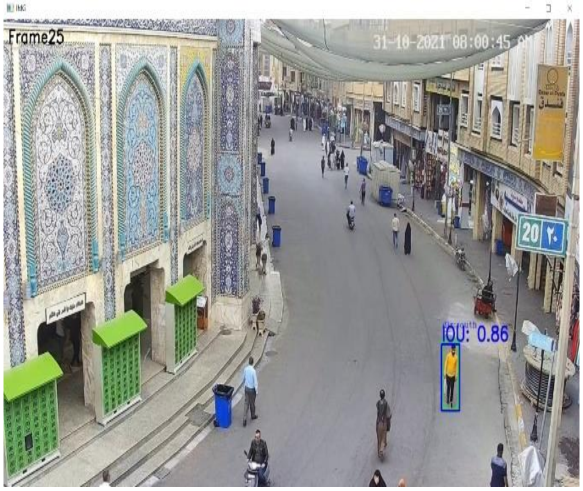
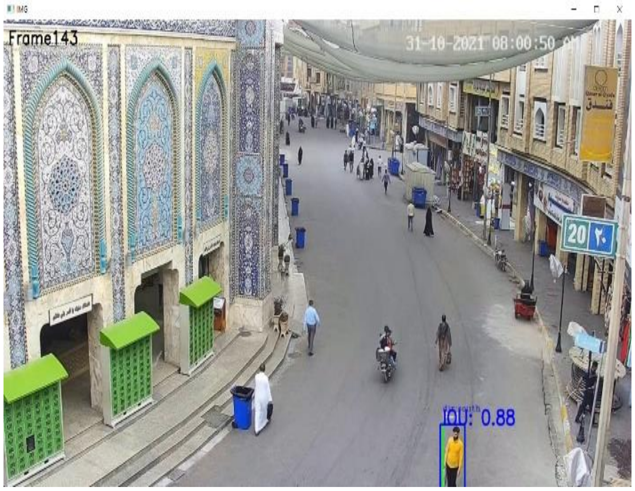


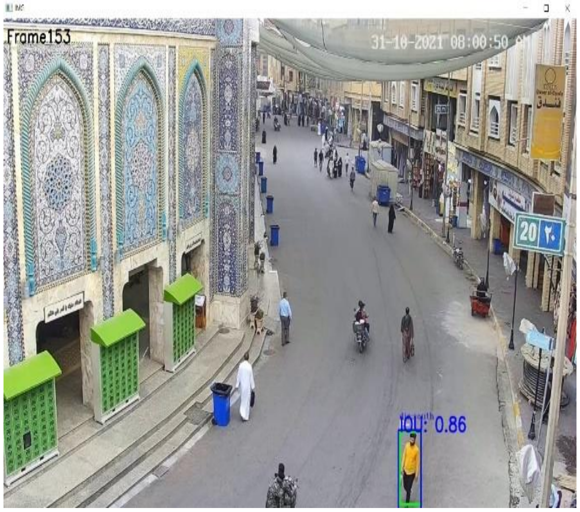
Figure (4.1): The subjective results for a number of samples from (C1.V1) from( MTA dataset) present the evaluation of detection stage.


IMG
Frame3
31-10-2021 08:00:44
IOU: 0.72


IMG
Frame15
31-10-2021 08:00:44
IOU: 0.83


IMG
Frame25
31-10-2021 08:00:45
IOU: 0.86


IMG
Frame143
31-10-2021 08:00:50
IOU: 0.88


IMG
Frame153
31-10-2021 08:00:50
IOU: 0.86

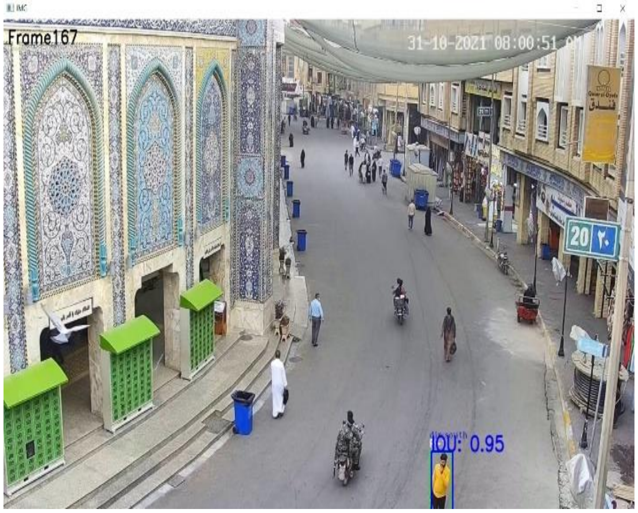

IMG
Frame167
31-10-2021 08:00:51
IOU: 0.95

Figure (4.2): The subjective results for a number of samples from (C0.V1) from(*Abbasid shrine in Karbala city* dataset) present the evaluation of detection stage.

### 4.4.2 Evaluation of Tracking Stage

In this stage, evaluating the performance of tracking algorithm depends on the ability of algorithm to keep the identity for moving target across video frames. Many obstacles cause occluded for target some of them was static occlusion as wall and building, but the others are dynamic occluded like moving cars or clutter by another object. Re-identify target after each occlusion presents the efficiency of tracking. Next steps present the( objective and subjective) results to evaluate the tracking task.

#### 4.4.2.1 Objective Evaluation Results

For Tracking stage, the parameters of (TP,FP,FN,TN) as shown in table(4.3) means:

- **TP** is referred to the bounding boxes that represent a moving person's body truly and considered finally as the same person with the same id detection correctly across video frames.
- **FP** is the bounding boxes that represent a moving person's body with ID falsely.
- **FN** is the bounding boxes that represent a missing track for target.
- **TN** refer to bounding boxes that represent a switch id for another object not same person.

The evaluation results of tracking stage are organized by choosing (990) frames as in table( 4.3) and present the results in the term of performance metrics( confusion matrix) in table(4.4) based on (precision and recall).

Table (4.3): The samples of evaluation  tracking stage using deep sort with Kalman filter

| C3 V1 | Total Frames 990 | | Success Tracking | Wrong Tracking | Missing Tracking | Different Object Tracking |
|---|---|---|---|---|---|---|
| | Frame no. | ID | TP | FP | FN | TN |
| | 1 | 1 | 0 | 0 | 1 | 0 |
| | 2 | 1 | 0 | 0 | 1 | 0 |
| | 3 | 1 | 1 | 0 | 0 | 0 |
| | 4 | 1 | 1 | 0 | 0 | 0 |
| | 5 | 1 | 1 | 0 | 0 | 0 |
| | 6 | 1 | 1 | 0 | 0 | 0 |
| | 7 | 1 | 1 | 0 | 0 | 0 |
| | 8 | 1 | 1 | 0 | 0 | 0 |
| | 9 | 1 | 1 | 0 | 0 | 0 |
| | 10 | 1 | 1 | 0 | 0 | 0 |
| | 11 | 1 | 1 | 0 | 0 | 0 |
| | 12 | 1 | 0 | 0 | 1 | 0 |
| | 13 | 1 | 0 | 0 | 1 | 0 |
| | 14 | 1 | 0 | 0 | 1 | 0 |
| | 15 | 1 | 0 | 0 | 1 | 0 |
| | 16 | 1 | 0 | 0 | 1 | 0 |
| | 17 | 1 | 1 | 0 | 0 | 0 |
| | 18 | 1 | 1 | 0 | 0 | 0 |
| | 19 | 1 | 1 | 0 | 0 | 0 |
| | 20 | 1 | 1 | 0 | 0 | 0 |

Table (4.4): Represent the confusion matrix for actual and predicted ID

| Total frame tracked =990 | | Actual ID | |
|---|---|---|---|
| Prediction ID | | Positive | Negative |
| | Positive | TP=955 | FP=10 |
| | Negative | FN=21 | TN=4 |

**ID Precision** = TP/(TP+FP)=955/(955+10) = **0.98 %**

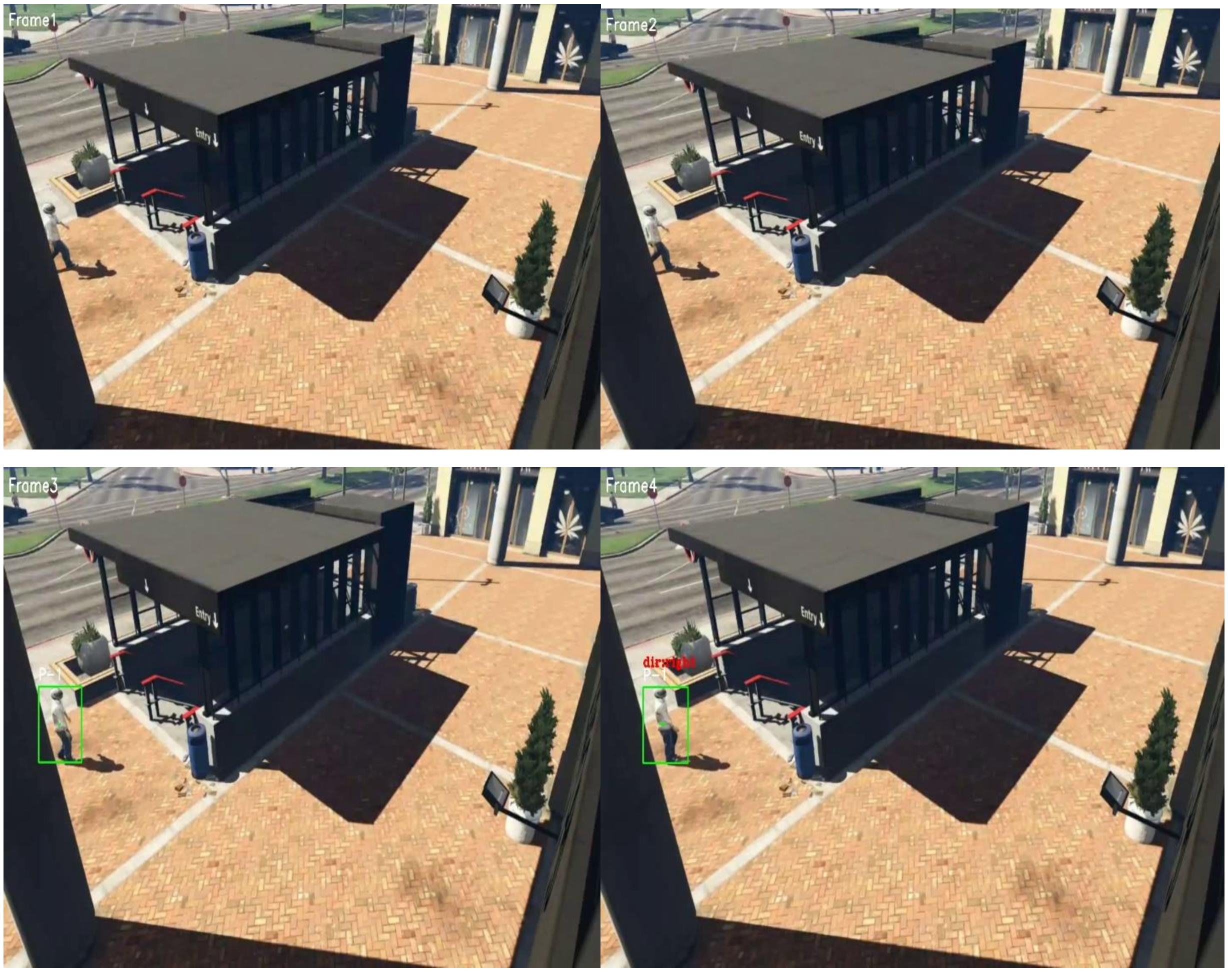
**ID Recall** = TP/(TP+FN) = 955/(955+21) = **0.97 %**

**F _ score** =2*(precision*recall/precision + recall) =2(0.98*0.97/0.98+0.97) = **0.97%**

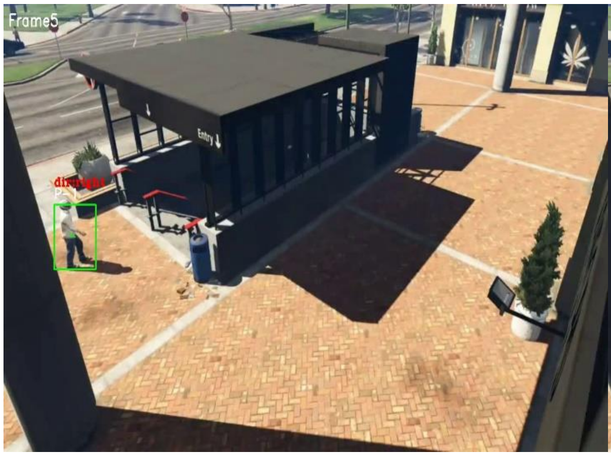
**Accuracy** = (TP+TN)/(TP+TN+FP+FN) = (955+4)/(955+10+21+4) = **0.96%**

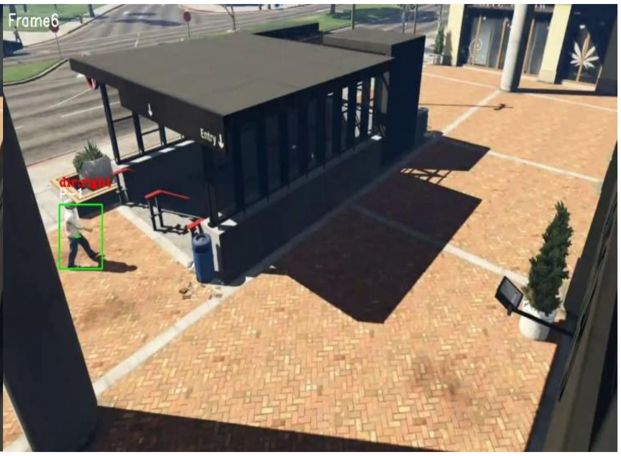
### 4.4.2.2 Subjective Evaluation Results

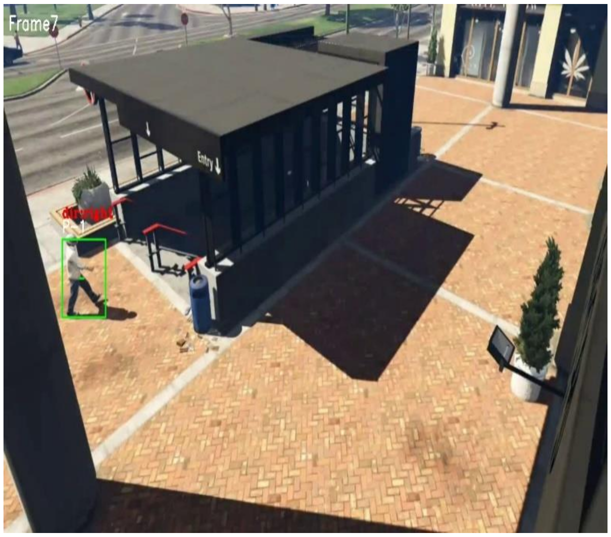
Several of frames(F1-21) from(C1,V1) of MTA dataset) shown in Figure(4.3) present tracking task for subjective evaluation. There is no tracking in (F1,F2 and F12-14).

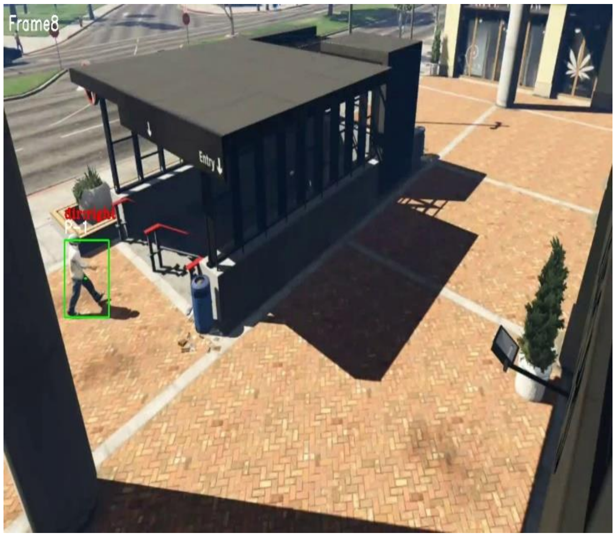


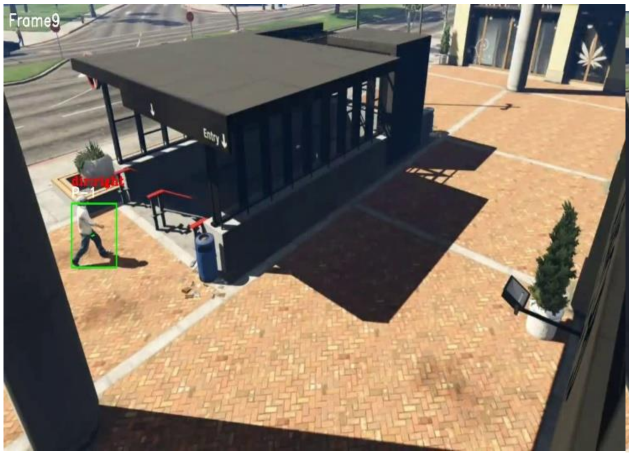
To be continued

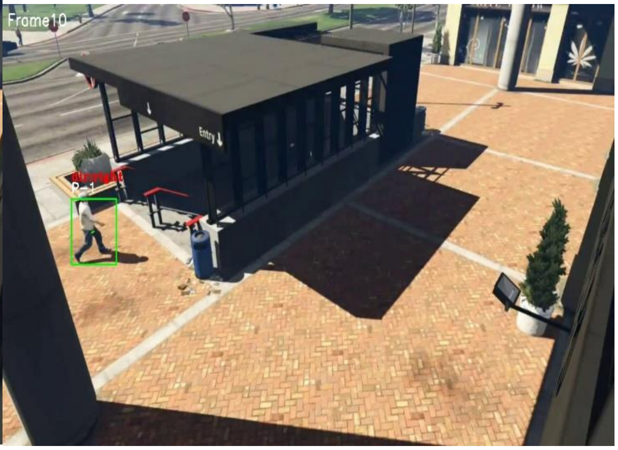












To be continued

Figure(4.3): General frames result of target tracking using deep sort with Kalman filter (MTA dataset)

To be continued

Figure(4.4): The frames present target tracking using deep sort with Kalman filter(*Abbasid shrine in Karbala city dataset)*

### 4.4.3 Evaluation of Prediction Stage

Creating the proposed model to predict direction of moving target based on the position feature for that target at each frame step required learn mode on the behavior of moving persons. The values of trajectory points represent the coordinates of position points for target which represent the behavior for moving target in scene. After extracting these points from trajectory and saving them in csv file, then using these sequence points for learning mode to predict the next position for moving target. Table(4.5) presents sample of data as (csv file).

Table (4.5): The sample of position feature for moving target

| ID | X | Y |
|---|---|---|
| 1 | 647 | 167 |
| 1 | 646 | 167 |
| 1 | 645 | 169 |
| 1 | 643 | 169 |
| 1 | 641 | 170 |
| 1 | 638 | 169 |
| 1 | 636 | 169 |

Two datasets are used in dissertation according to the training requirements implemented in some stages. The total samples of position features for various direction of many persons for each dataset is divided into 80% from original dataset for training and 20% for testing dataset as present in table(4.6) used for testing the system on unseen data.

Table (4.6):The summary of dataset division ratio of LSTM

| No. | Data name | Total samples 100% | Training dataset(train &valid)80% | Actual training set 90% | Validation set 10% | Testing dataset 20% |
|---|---|---|---|---|---|---|
| 1 | MAT | 18000 | 14400 | 12960 | 1440 | 3600 |
| 2 | *shrine in Karbala city* | 12400 | 9920 | 8928 | 992 | 2480 |

The performance of the training and evaluation modes is evaluated with two key points (metrics), which are accuracy and loss functions. A quick way to understanding the behavior for the learning of the proposed model on a specific dataset by evaluating the training and a validation dataset for specific epoch and plot the results as shown in figure (4.5), the figure present the relationship between the accuracy and loss of model for training stage and validation stage of prediction model, the curve shape refer to the better values of accuracy and losses after 20 epochs, the results values were (epoch:20, loss:0.0026, accuracy:0.9831, val_loss:0.0017, val_accuracy:0.944).

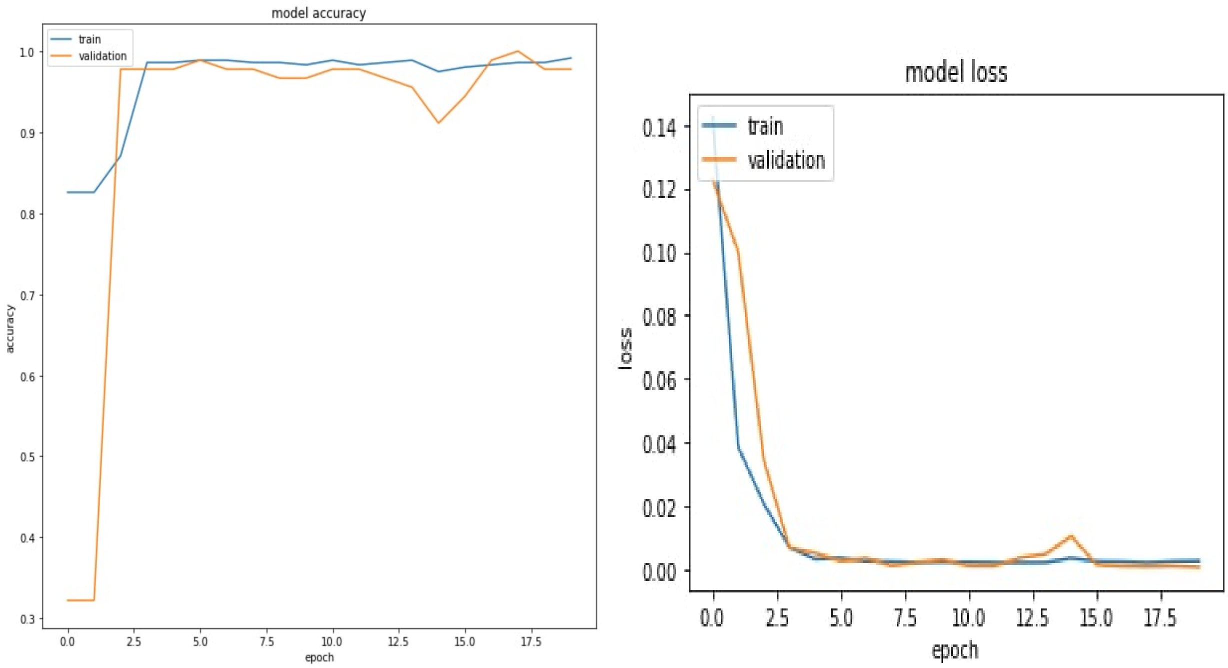


Figure (4.5): Accuracy and Loss Function of Proposed Model Training

### 4.4.3.1 Objective Evaluation Results of Prediction Stage

In this stage, various metrics are used for evaluation the results to predict the direction for moving target as confusion matrix, average displacement error (ADE) and final displacement error (FDE). For prediction stage to estimate the direction of moving person, the parameters of(TP,FP,FN,TN) means:

- **TP** is referred to the bounding boxes that represent a truly direction for a moving person in scene.
- **FP** is the bounding boxes that represent a falsely direction of moving person's body.
- **FN** is the bounding boxes that represent a missing detection with direction for target.

- **TN** refer to bounding boxes that represent assign same direction for another object not same person.

The evaluation results of the prediction stage are organized by choosing (530) frames as in Table(4.7) and present the results in the term of performance metrics( confusion matrix) in Table(4.8) based on (precision and recall).

Table (4.7): The samples of evaluation prediction stage

| **C3** | **Total frames = 530** | | | | **Succeed prediction** | **wrong prediction** | **Missing prediction** | **Different object predicted** |
|---|---|---|---|---|---|---|---|---|
| | **Frame no** | **ID** | **Actual direction** | **Predicted direction** | **TP** | **FP** | **FN** | **TN** |
| | 1 | 1 | East | Wrong pred | 0 | 1 | 0 | 0 |
| | 2 | 1 | East | East | 1 | 0 | 0 | 0 |
| | 3 | 1 | East | No pred | 0 | 0 | 1 | 0 |
| | 4 | 1 | East | No pred | 0 | 0 | 1 | 0 |
| | 5 | 1 | West | West | 1 | 0 | 0 | 0 |
| | 6 | 1 | East | East | 1 | 0 | 0 | 0 |
| | 7 | 1 | East | East | 1 | 0 | 0 | 0 |
| | 8 | 1 | East | South | 1 | 0 | 0 | 0 |
| | 9 | 1 | East | East | 1 | 0 | 0 | 0 |
| | 10 | 1 | East | East | 1 | 0 | 0 | 0 |
| | 11 | 1 | North | Wrong pred | 0 | 1 | 0 | 0 |
| | 12 | 1 | East | Wrong pred | 0 | 1 | 0 | 0 |

Table (4.8): Represent the confusion matrix for actual and predicted direction

| **Total frame detected =530** | | **Actual direction** | |
|---|---|---|---|
| **Prediction** | | Positive | Negative |
| **Direction** | Positive | **TP=475** | **FP=38** |
| | Negative | **FN=17** | **TN=0** |

**Precision** = TP/(TP+FP) =475/(475+38) = **0.93%**

**Recall** = TP/(TP+FN)=475/(475+17) = **0.97 %**

**F _score** = 2*(precision*recall/precision + recall) =2(0.93*0.97/0.93+0.97) = **0.95%**

**Accuracy** = (TP+TN)/(TP+TN+FP+FN)=475/530 = **0.9%**

Another test based on two metrics used for evaluated prediction results called:

- **Average displacement error:-**apply Euclidean distance between the actual trajectory and the predicted trajectory averaged over all times-steps.
- **Final displacement error:-** apply Euclidean distance between the actual trajectory point and the predicted trajectory point at end of n time-steps as shown in table(4.9).

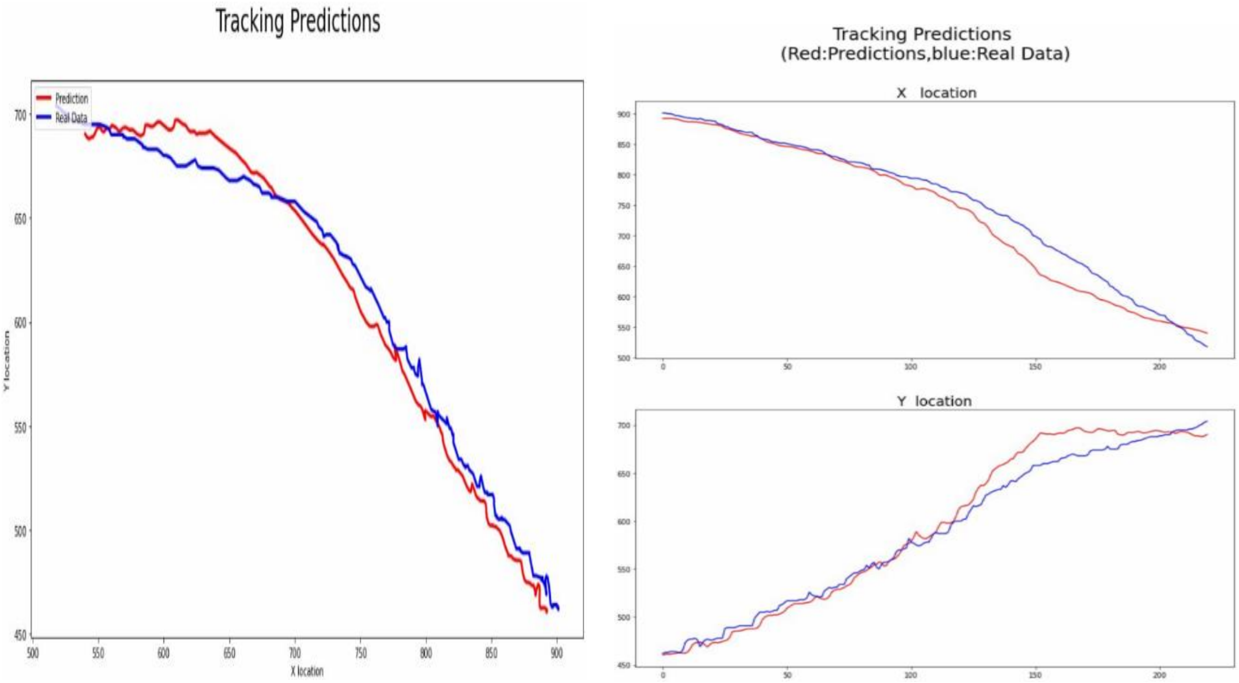
Table (4.9): Represent using two metrics(ADE , FDE) for evaluated predicted stage

| | | **ADE , FDE Evaluation Metrics** | | |
|---|---|---|---|---|
| **C4,V1** | | **Actual Trajectory points** | **Predicted Trajectory points** | **Distance between two points** |
| Frame | ID | X1,Y1 | X2,Y2 | Dist<50 px |
| 1 | 1 | 452,399 | 433,409 | 21 |
| 2 | 1 | 448,402 | 430,409 | 19 |
| 3 | 1 | 441,406 | 427,411 | 14 |
| 4 | 1 | 439,407 | 424,413 | 16 |
| 5 | 1 | 433,412 | 418,417 | 15 |
| 6 | 1 | 432,415 | 412,421 | 20 |
| 7 | 1 | 424,419 | 409,424 | 15 |
| 8 | 1 | 420,420 | 403,429 | 19 |
| 9 | 1 | 416,421 | 398,433 | 21 |
| 10 | 1 | 409,422 | 391,436 | 22 |

The results of predicted new positions for moving target and detect the direction depending on the change in coordinates to estimate the where the target may reappear. All the Figures( 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11) represent the evaluation for predicted direction with actual direction for many targets from different videos of many cameras. Using two metrics ADE and FDE to evaluate the results of predictions. Some results are presented in Table (4.9).

Table(4.10): Represent the actual and predicted trajectory points for moving target from (C0,V1)

| Actual x | Actual y | Predict x | Predict y |
|---|---|---|---|
| **901** | **462** | **892.1** | **460.6** |
| **901** | **463** | **892.3** | **461.1** |
| **900** | **463** | **892.4** | **461.4** |
| **900** | **464** | **891.3** | **461.7** |
| **899** | **464** | **890.9** | **462.5** |
| **897** | **465** | **889.5** | **462.4** |
| **896** | **473** | **888.7** | **463.4** |
| **895** | **476** | **887.5** | **466.5** |



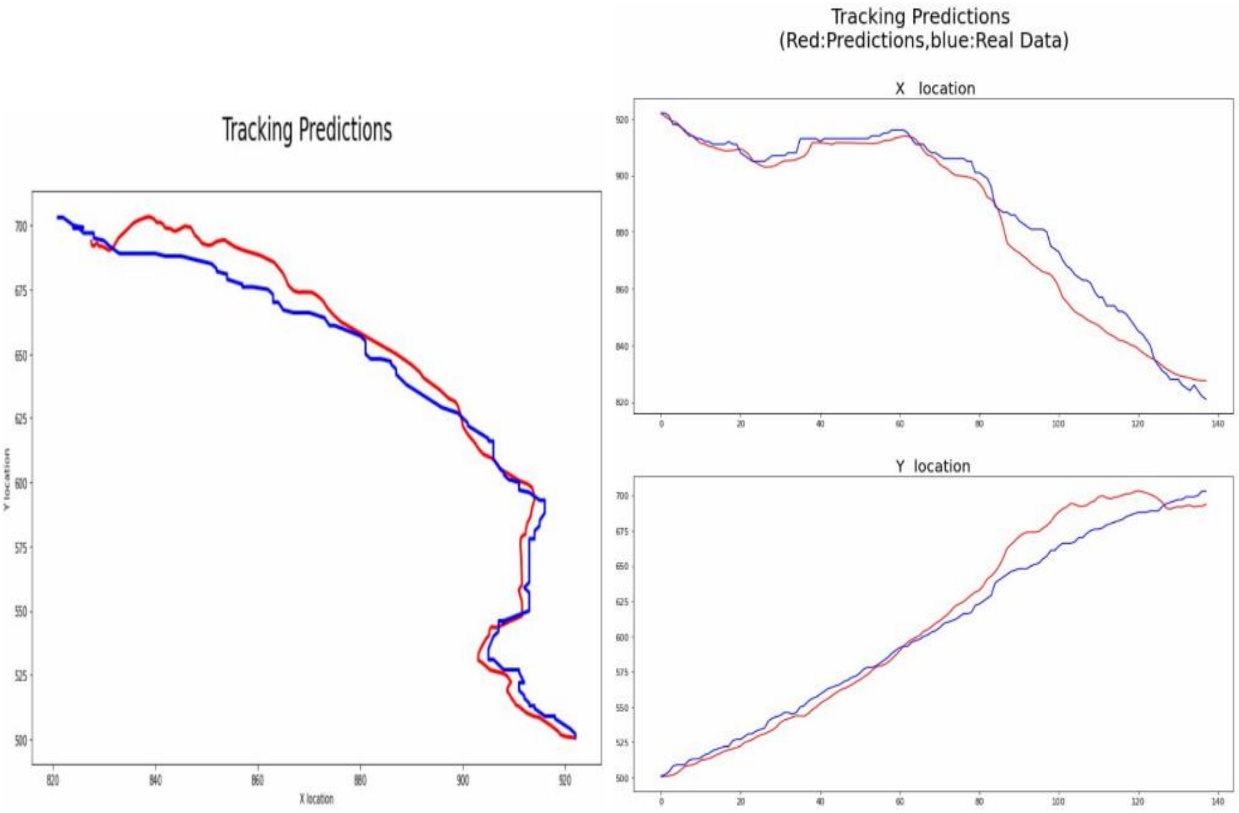
a b

Figure (4.6): a. trajectory points , b. trajectory points for x ,y values separately when blue color represent actual trajectory points while red color represents predicted trajectory points from (C0,V1).

Table(4.11): Represent the actual and predicted trajectory points for moving target from(C1,V1).

| Actual x | Actual y | Predict x | Predict y |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **922** | **501** | **921.9** | **500.5** |
| **922** | **502** | **920.8** | **500.8** |
| **921** | **504** | **919.8** | **501.1** |
| **918** | **508** | **919.1** | **501.7** |
| **918** | **509** | **917.1** | **503.1** |
| **917** | **509** | **915.3** | **505.3** |
| **916** | **512** | **914.9** | **507.8** |
| **914** | **513** | **913.7** | **508.4** |

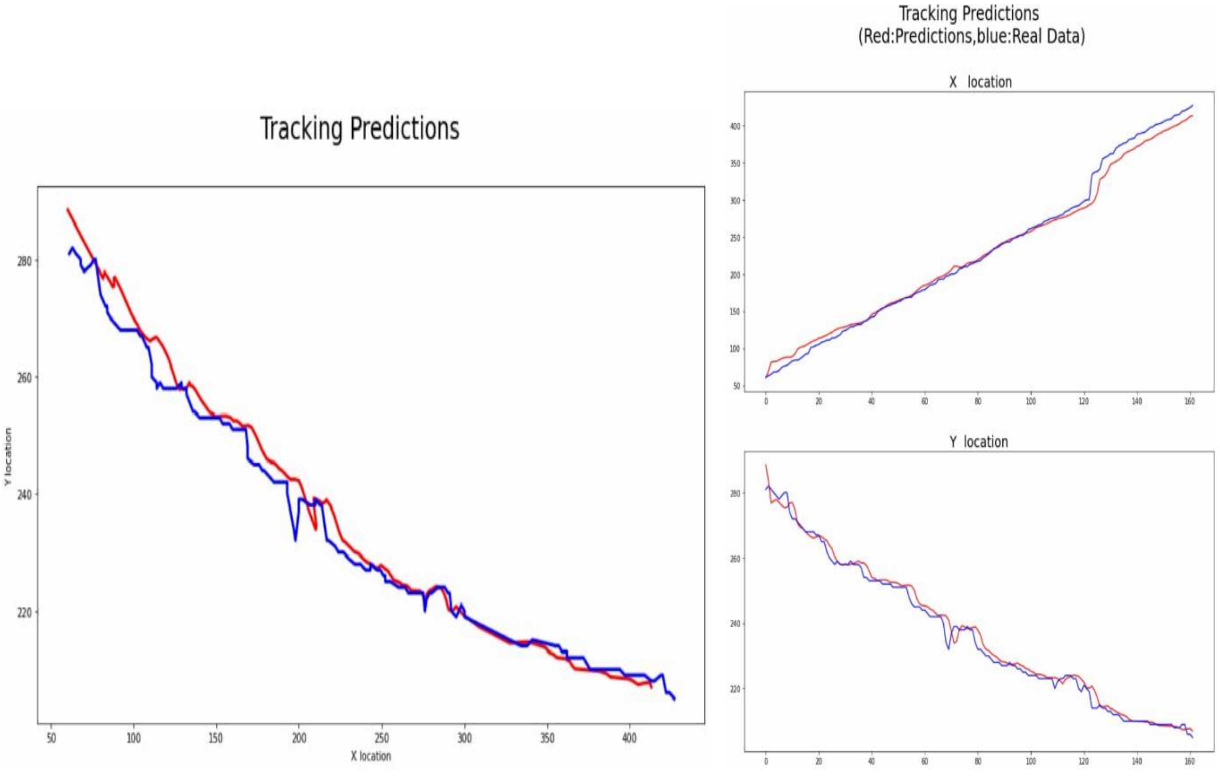


a b

Figure (4.7): a. trajectory points , b. trajectory points for x ,y values separately when blue color represent actual trajectory points while red color represents predicted trajectory points from(C1,V1).

Table(4.12): Represent the actual and predicted trajectory points for moving target.

| Actual x | Actual y | Predict x | Predict y |
|---|---|---|---|
| 61 | 281 | 60.1 | 288.4 |
| 63 | 282 | 68.7 | 283.6 |
| 65 | 281 | 81.4 | 276.8 |
| 68 | 280 | 82.1 | 277.1 |
| 68 | 279 | 82.4 | 277.4 |
| 70 | 278 | 84.2 | 276.8 |
| 74 | 279 | 86.1 | 276.9 |
| 76 | 280 | 67.3 | 275.4 |

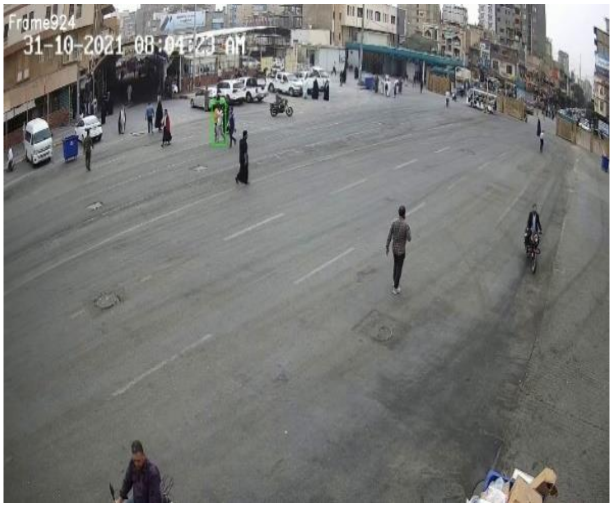


a b

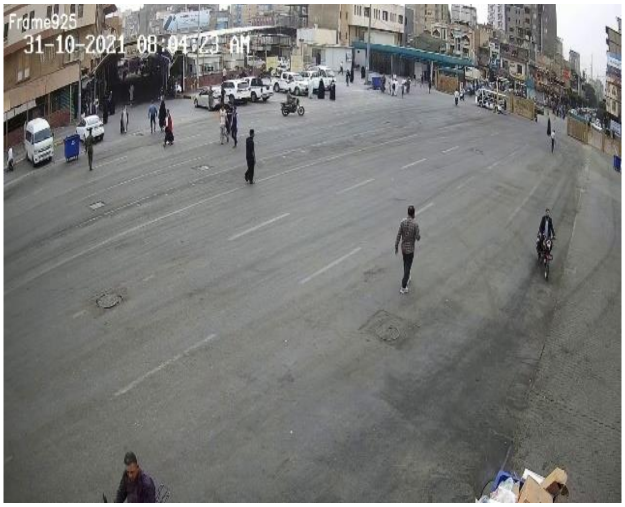
Figure (4.8): a. trajectory points , b. trajectory points for x ,y values separately when blue color represent actual trajectory points while red color represents predicted trajectory points from (C0,V1).

#### 4.4.3.2 Subjective Evaluation Results of Prediction Stage

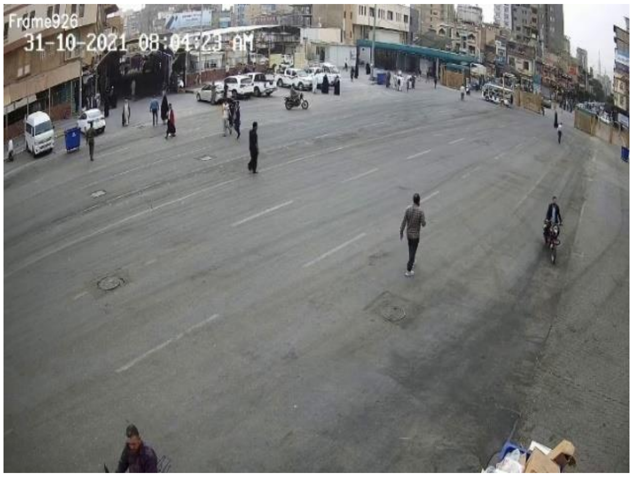
In this scenario, the system tracking target and the predicted direction of moving target as shown in Figure(4.9), can see the target appears at frame no.925, and the target is occluded from frame no.925 to frame no.964 when the target disappears behind other moving persons in scene. In frame no.965, the target re-appeared in scene. The system predicted the direction of moving target using previous information based on LSTM model which support the deep sort with Kalman filter to association correct states of moving target.



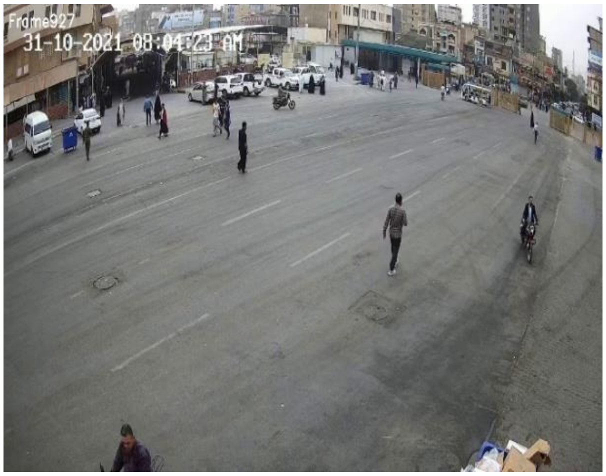
C3.V1.F 924



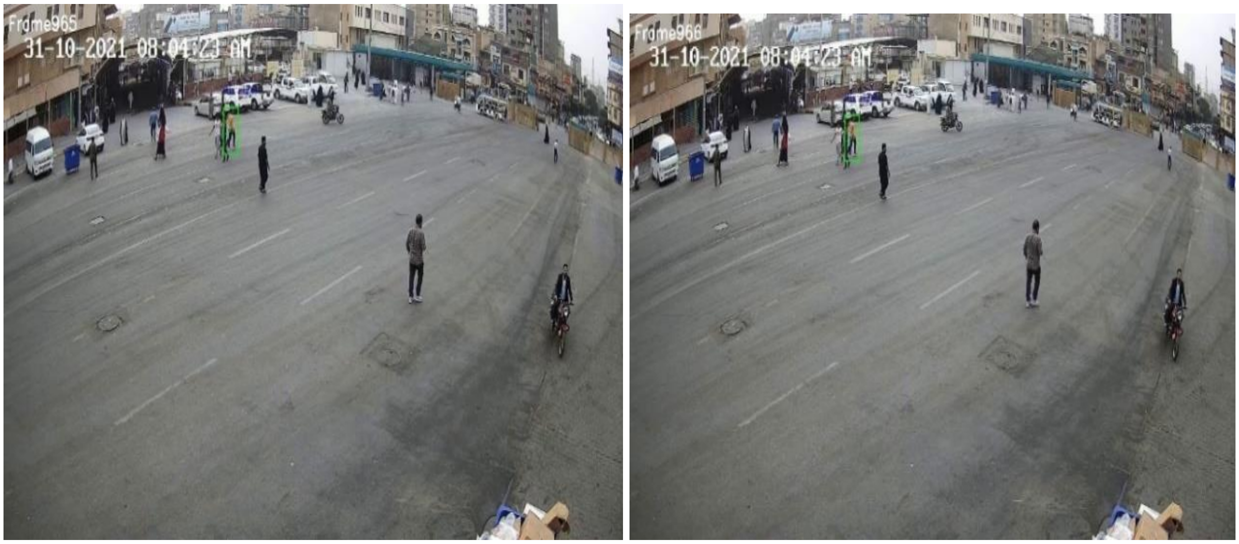
C3.V1.F 925



C3.V1.F 926



C3.V1.F 927



C3.V1.F 965 C3.V1.F 966

Figure (4.9):General frames results (F925-F966) show tracking and predicted direction of the target across occlusion keep same id and re-identify the same object.

Other scenarios to predict the direction of moving target in different videos from dataset (MTA) as shown in Figures(4.10) and (4.11). Different scenarios present many persons moving in scene and predicted the direction for each moving person.

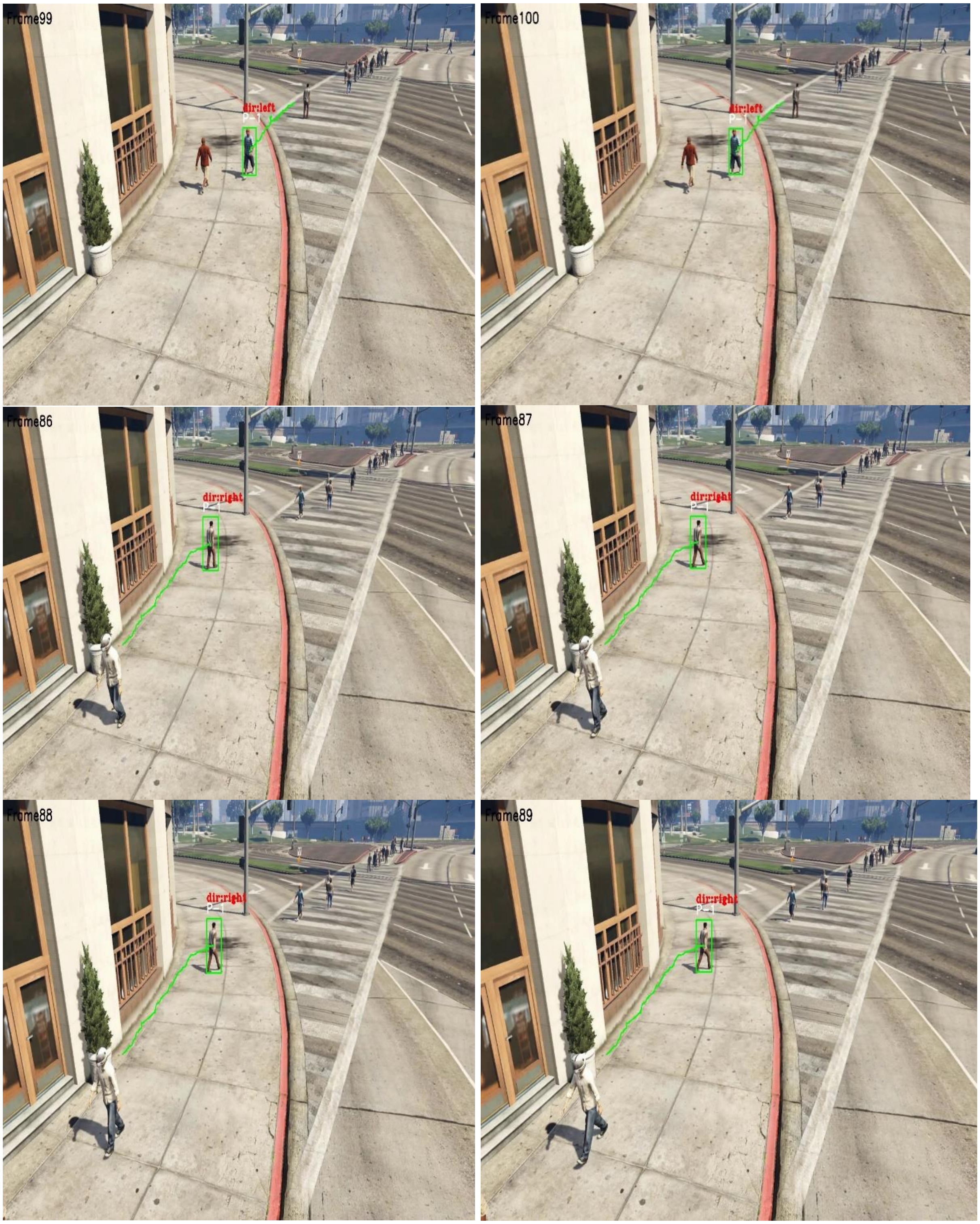


Figure (4.10): Predicted direction of moving target from dataset(MTA,C4,(V1,F 99-100),(V2,86-89))

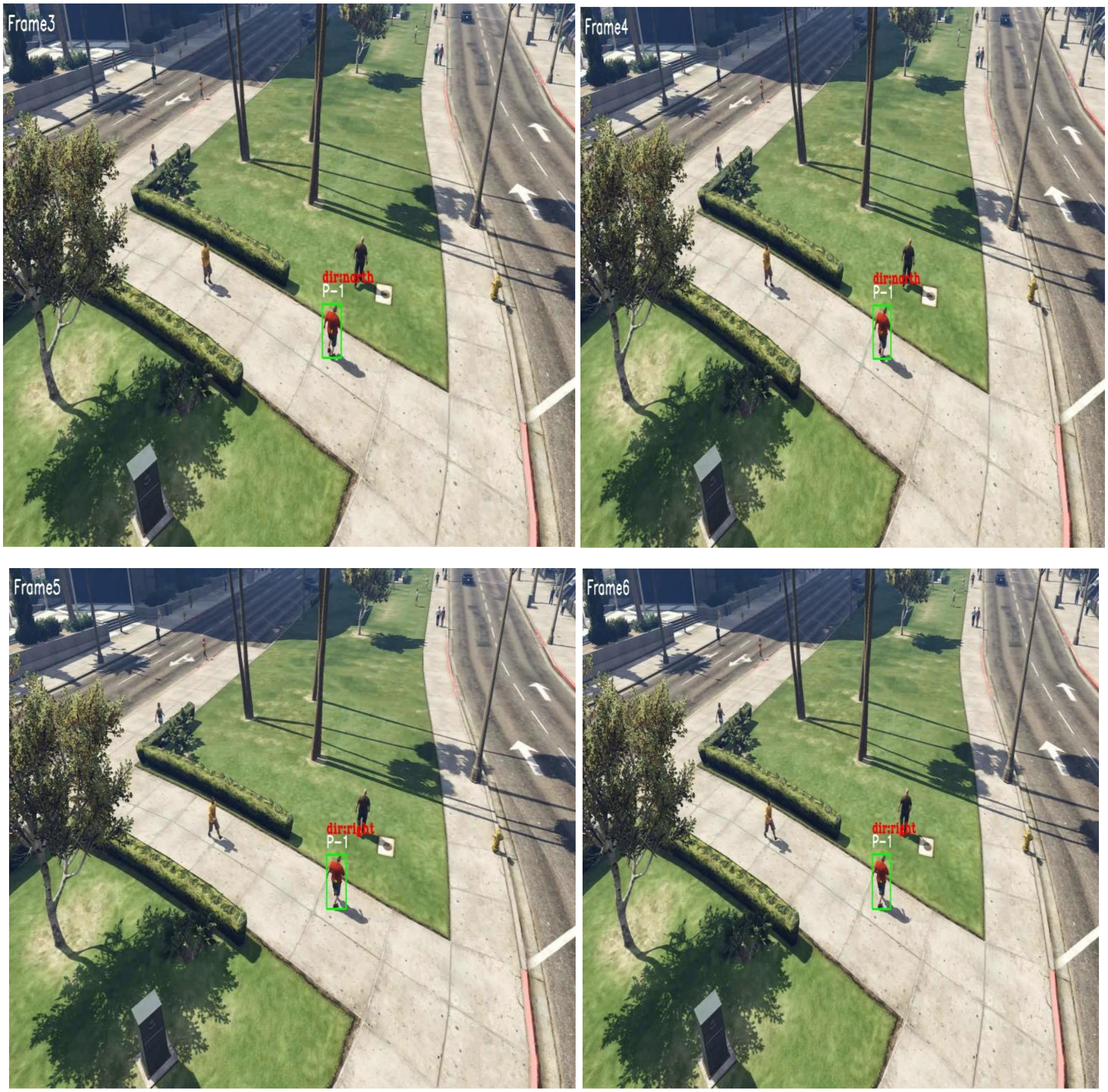


Figure (4.11): Predicted direction of the moving target from dataset(MTA,C2,V1,F3-6)

### 4.4.4 Evaluation of Re-identification Stage

Evaluation stage for proposed model based on CNN for reidentification target using two datasets each one used to build proposed model. There are many steps as building dataset then learning mode using for building the proposed model.

#### 4.4.4.1 Building of Dataset

The first step for create re-identification model is building of dataset, different scenarios for images of persons cropped from frames of many videos and using for training CNN network to building model. Two datasets are use in dissertation according to the training requirements implemented in some stages. The total samples of cropped images of persons for each dataset is divided into 85% from the original dataset as a training data and 15% for testing dataset used for testing the system on unseen data. The training dataset is also divided into 80% actual training dataset and 20% validation dataset. The summary of datasets division ratio represented in Table (4.13).Two models created for re-identification each one depends on one dataset, first model use MAT dataset and the other model is based on *Abbasid shrine in Karbala city* dataset, sample of frames presented in Figures (4.12) and (4.13) refers to cropped frames for each dataset. The next step is learnt mode, in the learning step the proposed re-identification model learned on the training dataset which contains cropped frames of persons, and the weights of each layer are adjusted during the training process. After the model reached convergence with the minimum error rate, it must validate work by simulation of this model with validation data as input to the trained network (model). The network uses the saved weights and calculates the output class of each input image and if the output result is same as the class of the actual label of the input images, then the network is validation. In this case, the proposed model becomes ready to use for re-identifying the target in scene.

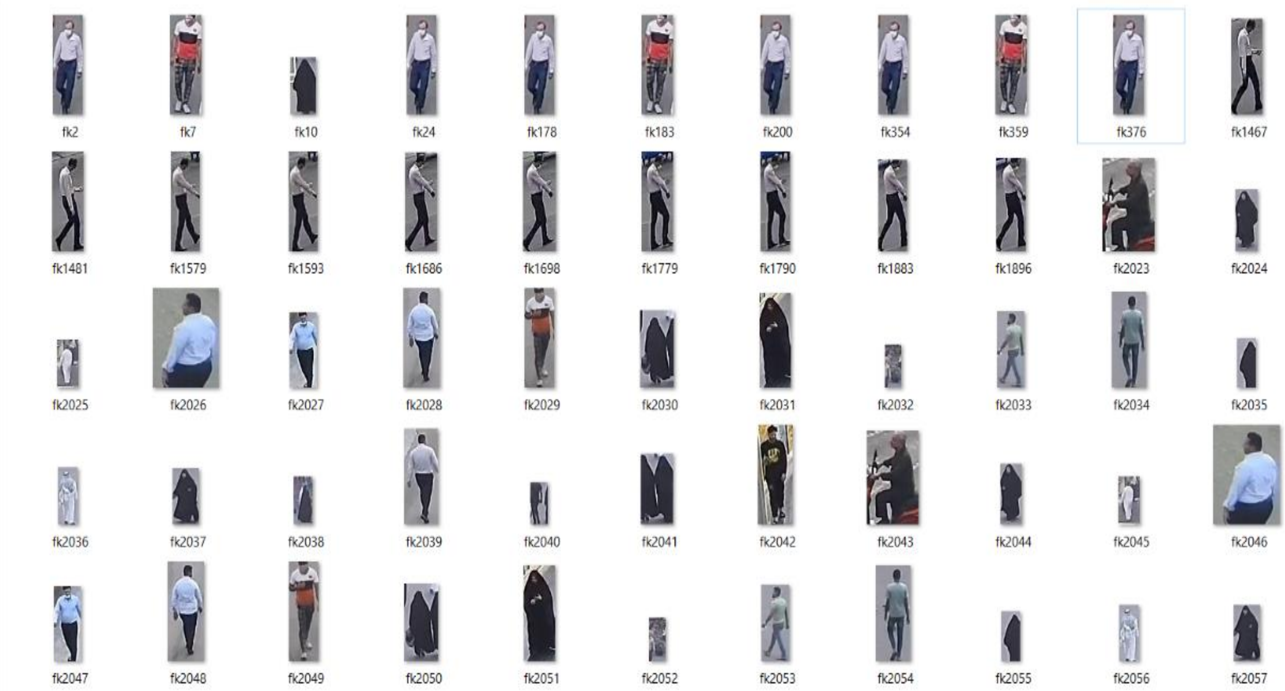


Figure (4.12): The samples for cropped images from videos of *shrine in Karbala city* dataset.

Table (4.13):The summary of dataset division ratio

| No. | Data name | Total subject 100% | Training dataset(train &valid)85% | Actual training set 80% | Validation set 20% | Testing dataset 15% |
|---|---|---|---|---|---|---|
| 1 | MAT | 16000 | 13600 | 10880 | 2720 | 2400 |
| 2 | *shrine in Karbala city* | 12000 | 10200 | 8160 | 2040 | 1800 |

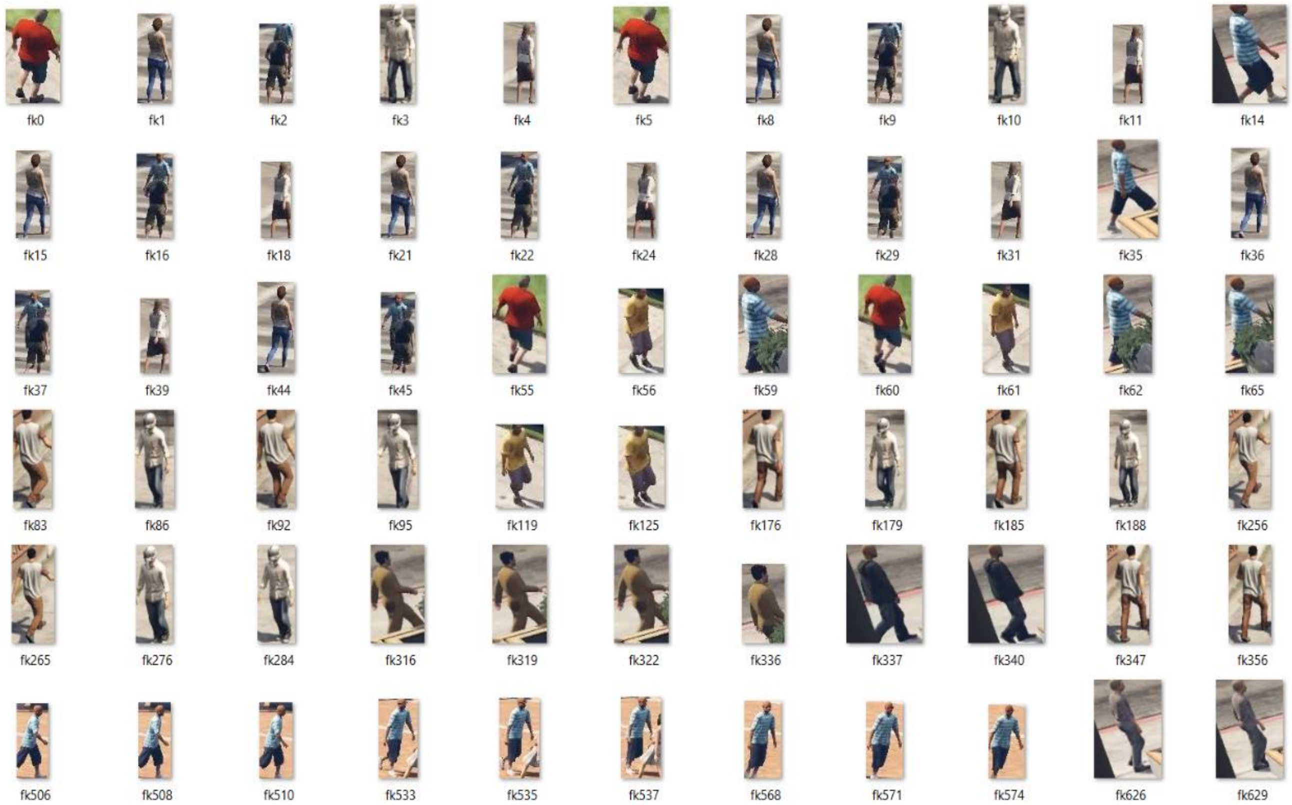


Figure (4.13): The samples for cropped images from videos of MTA dataset

The performance of the training and evaluation modes is evaluated within two key points (metrics), which are accuracy and loss functions. A quick way to understanding the behavior for the learning of the proposed model on a specific dataset by evaluating the training and a validation dataset for specific epoch and plot the results as can be shown in Figure (4.14), the figure present the relationship between the accuracy and loss of model for training stage and validation stage of re-identification model, the curve shape refer to the better values of accuracy and losses after 30 epochs, The results of evaluation of model were (all epoch=30, validation=0.25, batch size=50, accuracy= 0.9877 at 25 epoch).

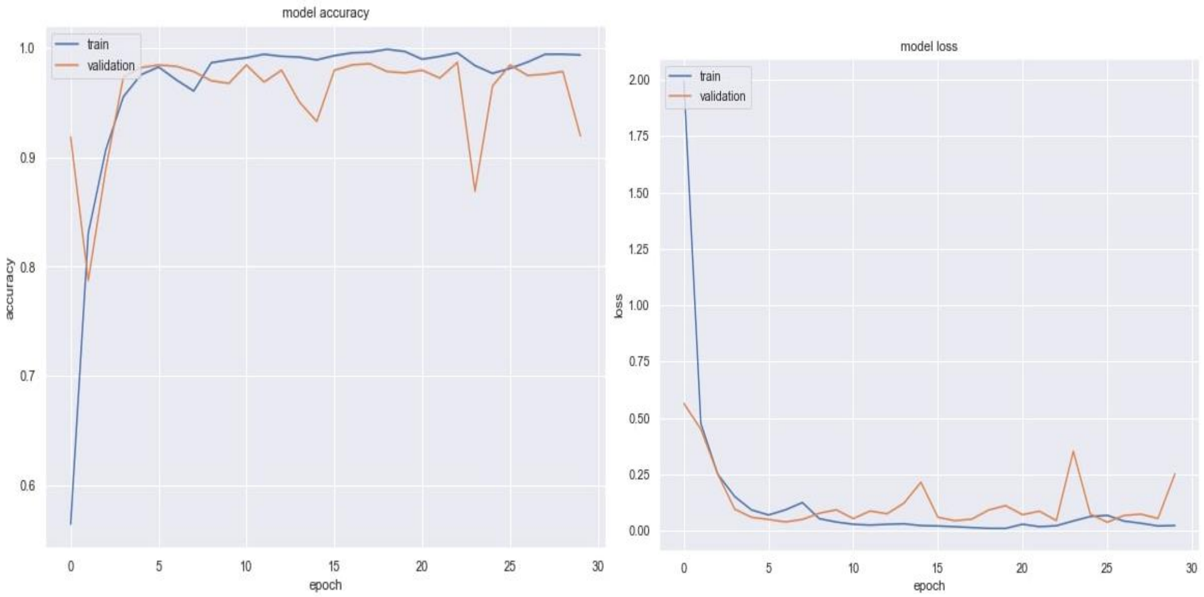


Figure (4.14): Accuracy and Loss Function of Proposed Model Training

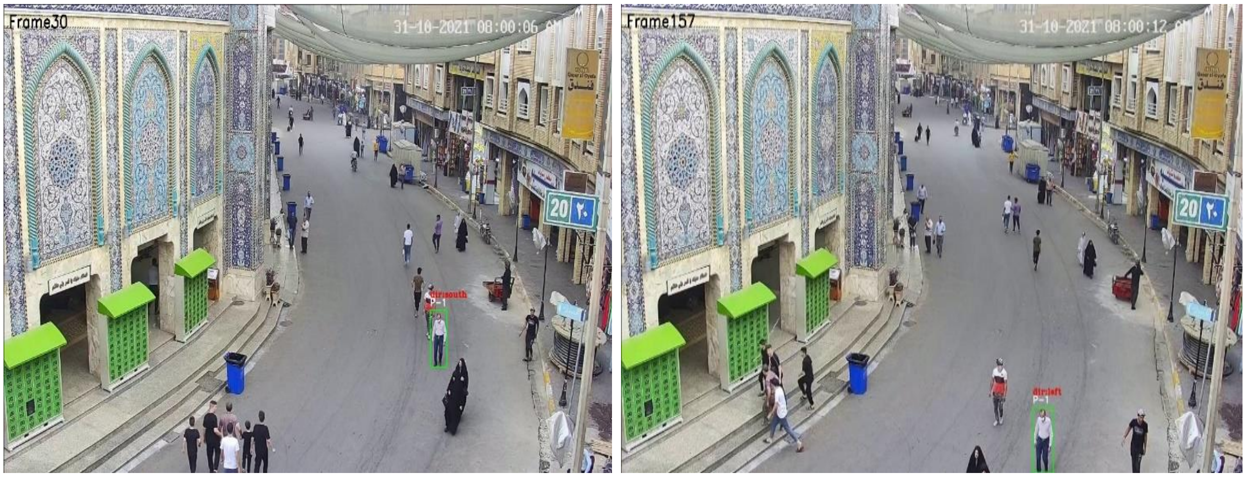
#### 4.4.4.2 Objective Evaluation Results of Re-identification Stage

The first step is preparing the videos from cameras (dataset) to apply the re-identification model on them, for each video that record from specific camera can select number of persons in scene each one of them have unique behavior and consider as a target. Next step at each process, the system detects and track single target (person) for each one of them in view. The system uses LSTM model to support deep sort algorithm and predict the direction of single target moving in scene. For each person in video can predict the last direction when the target leaves the view, if the system succussed for selecting the next camera in the same side where the target is moving toward and apply the re-identification model to identify the same target then the case is considered true positive (TP) because the system is capable to predict where the target may reappear and identify the same target in the neighbor’s camera. While if the system is selecting the wrong camera or predict the wrong direction of target that cause lost the target, this case can be considered as false negative(FN), but if the system selects the correct next camera in correct side where the target is walking toward but the rei-dentification model re-identify another person (not target), this case is considered true negative(TN). The last case is considering a false positive when the system predicted the wrong direction that lead to select the wrong neighbor's camera, but the re-identification model re-identifies the same target in this camera view. There are many cases represent different scenarios from many cameras of dataset presented in this chapter to evaluate the performance of re-identification model in two stages ( objective and subjective stages).

### Case Study 1.

In this case, evaluating the re-identification model using sample videos from dataset of *Abbasid shrine in Karbala city*. The scenario presents the target appearing at first in camera number(0) when the system uses the YOLOv3 algorithm to detect the target and then starts the tracking process depending on the Kalman filter with the deep sort algorithm, at the same time the system is based on LSTM to predict the direction of the target at each next step (new location)in the scene. The prediction process continues for the target motion until the target may leave the boundary of the field of view with a specific side(right, left, north, south). In this case, the target leaves the boundary of the camera in the left direction as shown in Figure(4.15).

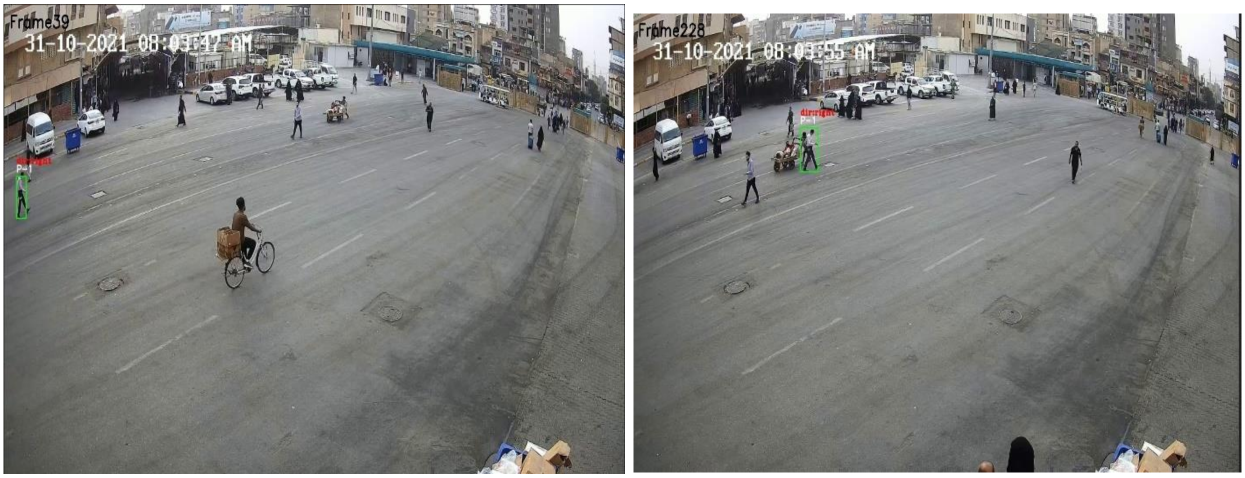
When the target leaves the scene and the last direction is toward the left side, the system checks the topology table to candidate the cameras that are located on the left side and checks the distance between the current camera with its neighbors on the left side to select the camera (has high priority in topology table) that means the candidate camera has a short distance from the position of camera which target leaved it. In this scenario, the system expects the target to appear in the candidate camera



C0.V1.F30 C0.V1.F157

Figure (4.15): (C0,V1, *Abbasid shrine in Karbala city* dataset ) presents the scenario when the target walking in scene in frame no.30 then the target leaves the camera view in frame no.157 at left direction

If the target appears in this camera the system will be using the re-identification model to re-identify the target and continues in the tracking task, but if the target did not appear in the candidate camera scene after about (20 frames), the system switches to another camera that has a second degree in priority for cameras located in the left side. In this case, the target walking toward the left side, and there are two cameras in that side (c1, c2). The information of the topology table refers to c2 has a shorter distance(114m) to c0 compared with c1 (125) the system tries to detect the target in c2 at first to search about target but if the target does not appear in a limited period, then the system switch to camera c1 and repeats the re-identify process to detect the target. The moving target appears at c1 as shown in Figure(4.16).

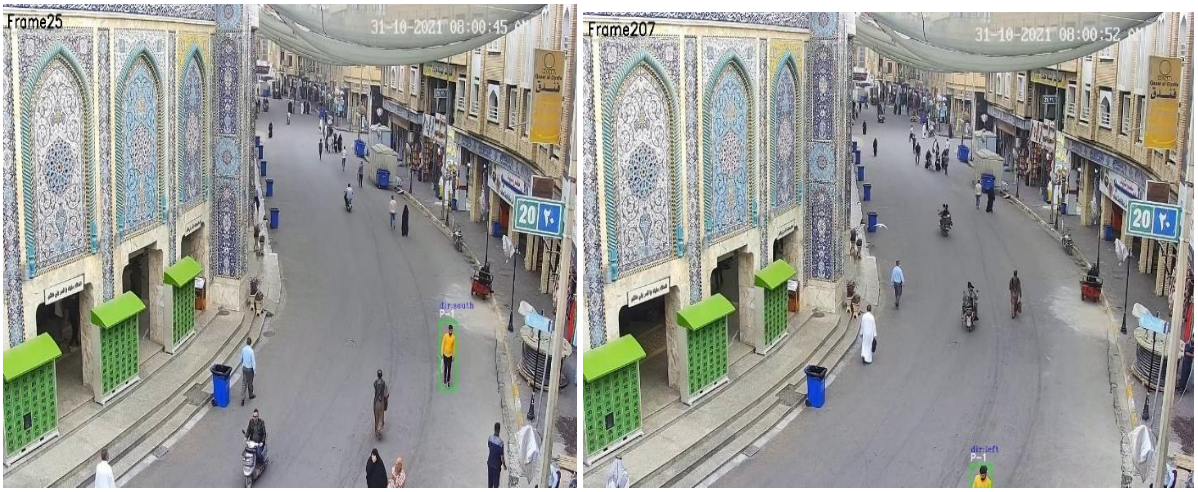


C1.V1.F39 C1.V1.F228

Figure(4.16): Frame(39) presents the scenario when target appears in the camera view(C1) on the left side

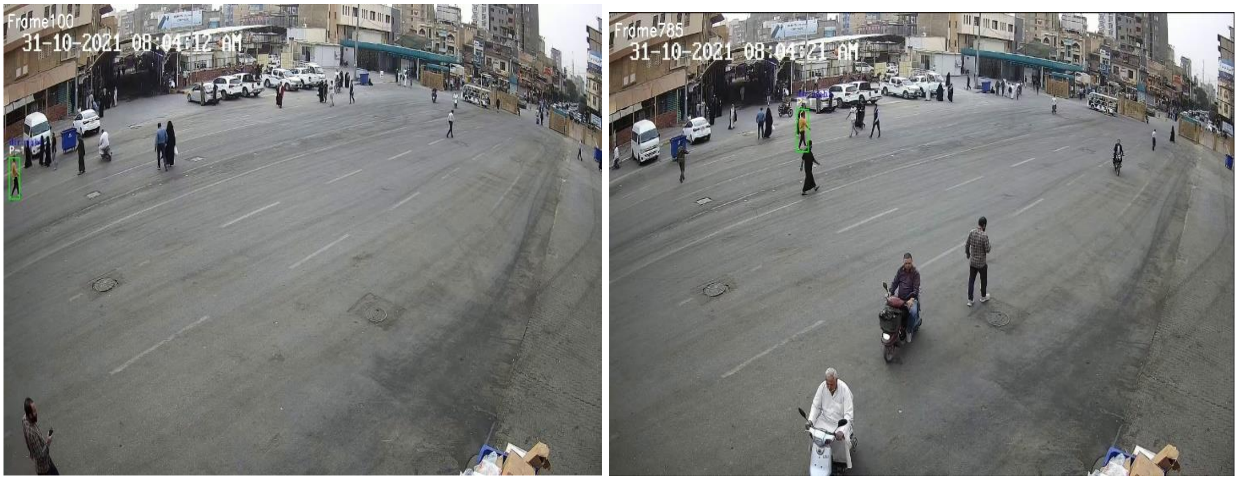
## Case Study 2

In this case, evaluating the re-identification model using sample videos from dataset of *shrine in Karbala city* dataset. When the system using YOLOv3 to detect the target(another person) then tracking based on Kalman filter with deep sort algorithm in scene. In this case, the target appears at first in camera (C0), the prediction result of direction for target refers to the left side, there are two cameras appear in topology table in that side. The system starts to select one of these two cameras, it selects the camera which has the short distance between it and the current camera. Then the system applies the re-identification model to recognize if the target may appear in this camera or not. Figure(4.17) presents the C0 when the target detected and tracked in scene and Figure(4.18) presents the C1 when the reidentification model recognizes the target in a neighbor’s camera.



a.(C0.V1.F25) b.(C0.V1.F207)

Figure( 4.17): Frame(no.25) a. presents the scenario when target appears in (C0) walking to the left side and b. presents the target when leaving the camera view in frame(no.207).



a.(C1.V1.F100) b.(C1.V1.F785)

Figure(4.18): Where a. frame(no.100) presents the scenario when the target appears in neighbor camera(C1) and b. frame (no. 785) the target is walking in the scene with the predicted direction to the right side.

Table (4.14): Presents a sample of persons in many cameras (different videos lengths, all videos' dimensions 1280×720) for evaluation re-identification stage(*Abbasid shrine in Karbala city* dataset).

| NO. | Direction | Neighbor's cameras | Last direction of target before leave view | Candidate cameras depend on short distance | Dose target appear in scene | Second priority | Dose target appear in next camera | True prediction and identify target TP | False prediction but true target FP | True predicted but wrong target TN | False prediction Misses target FN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **C 0** | **L** | **C1,C2** | **L** | **C2** | **no** | **C1** | **yes** | **1** | **0** | **0** | **0** |
| **V 1** | | | **L** | **C2** | **yes** | **----** | **----** | **1** | **0** | **0** | **0** |
| **S 5** | | | **L** | **C2** | **yes** | **-----** | **----** | **1** | **0** | **0** | **0** |
| | | | **S** | **C2** | **yes** | **-----** | **----** | | **1** | **0** | **0** |
| | **R** | **------** | | | | | | | | | |
| | **S** | **------** | **S** | **---** | **No** | **----** | **----** | **0** | **0** | **1** | **0** |
| | **N** | **------** | | | | | | | | | |
| | | | | | | | | | | | |
| **C 1** | **L** | **C2** | | | | | | | | | |
| **V 1** | **R** | **-----** | | | | | | | | | |
| **S 1** | **S** | **C3** | **S** | **C3** | **Yes** | **-----** | **-----** | **1** | **0** | **0** | **0** |
| | **N** | **C0** | | | | | | | | | |
| **C2** | **L** | **C0,C3** | **L** | **C0** | **No** | **C3** | **yes** | **1** | **0** | **0** | **0** |
| **V1** | **R** | **C1** | | | | | | | | | |
| **S1** | **S** | **-----** | | | | | | | | | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | ----- | | | | | | | | | |

Table (4.15): Represent the confusion matrix for re-identification model from *Abbasid shrine in Karbala city* dataset

| **Total samples(persons) =7** | | **Actual re-identify** | |
|---|---|---|---|
| **Prediction re-identifies** | | **Positive** | **Negative** |
| | **Positive** | **TP=5** | **FP=1** |
| | **Negative** | **FN=0** | **TN=1** |

**Precision** = TP/(TP+FP) =5/(5+1) = **0.83%**

**Recall** = TP/(TP+FN)=5/(5+0) = **100%**

**F _score** = 2*(precision*recall/precision + recall) =2(0.83*1/0.83+1) = **0.91%**

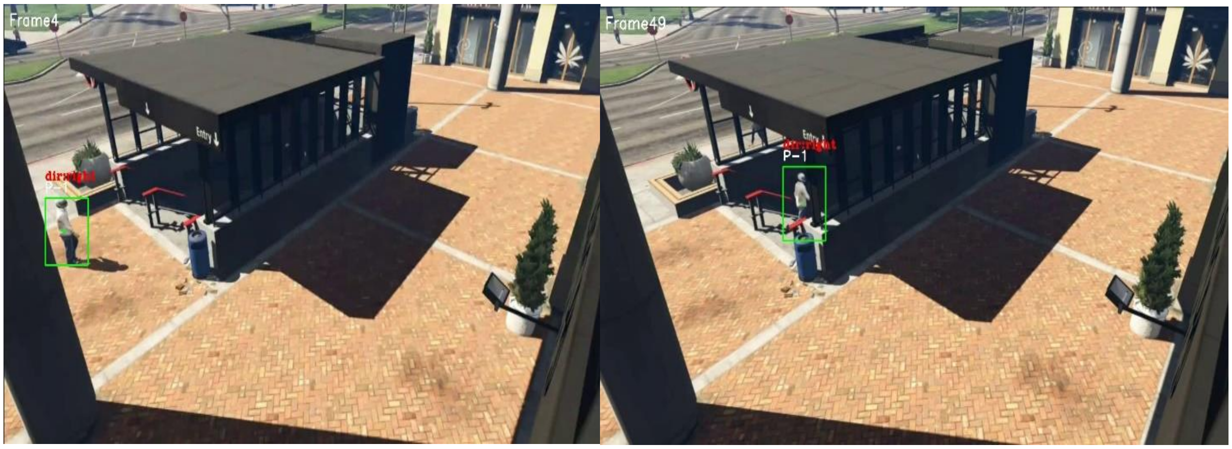
**Accuracy** = (TP+TN)/(TP+TN+FP+FN)=6/7 = **0.86%**

From MAT dataset select samples with number of scenarios to evaluate the proposed system such as follow:

### Case Study 3

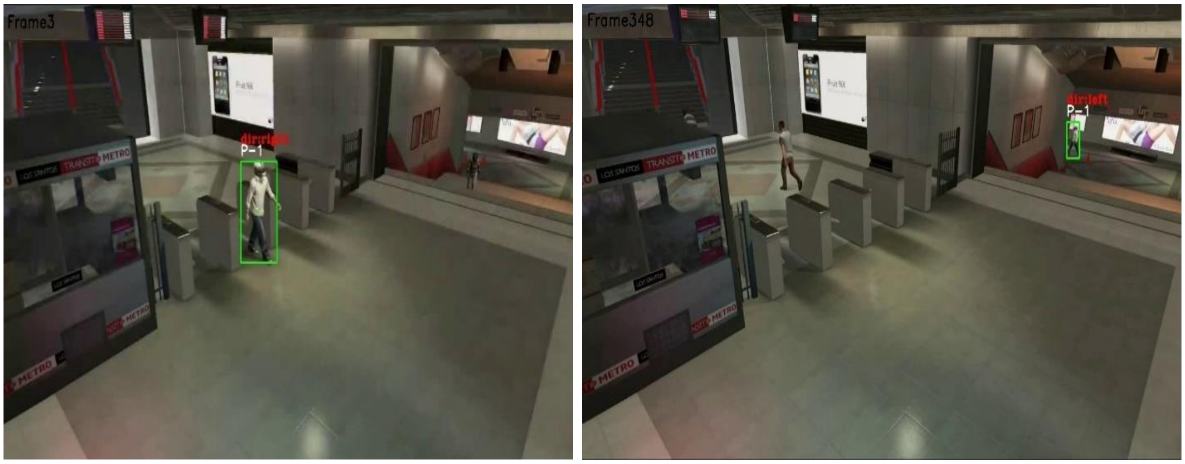
In this case, evaluating the re-identification model using sample videos from dataset of MTA dataset. When the system using YOLOv3 to detect the target(person) then tracking based on Kalman filter with deep sort algorithm in scene. In this case, the target appears at first in camera (C1). The prediction result of direction for target refers to the right side, there are one camera appear in topology table in that side with distance (144m) between two cameras. The system applies the re-identification model to recognize if the target appears in this camera or not.

Figure(4.19) presents the C1 when the target appears for first once and Figure(4.20) present the C0 when the reidentification model recognize the target in a neighbor's camera.



a.(C1.V1.F4) b.(C1.V1.F49)

Figure(4.19): Where a. frame(no.4) presents the scenario when the target appears in neighbor camera(C1) and in b. frame (no.49) the target walking in the scene predicting last direction to the right side.

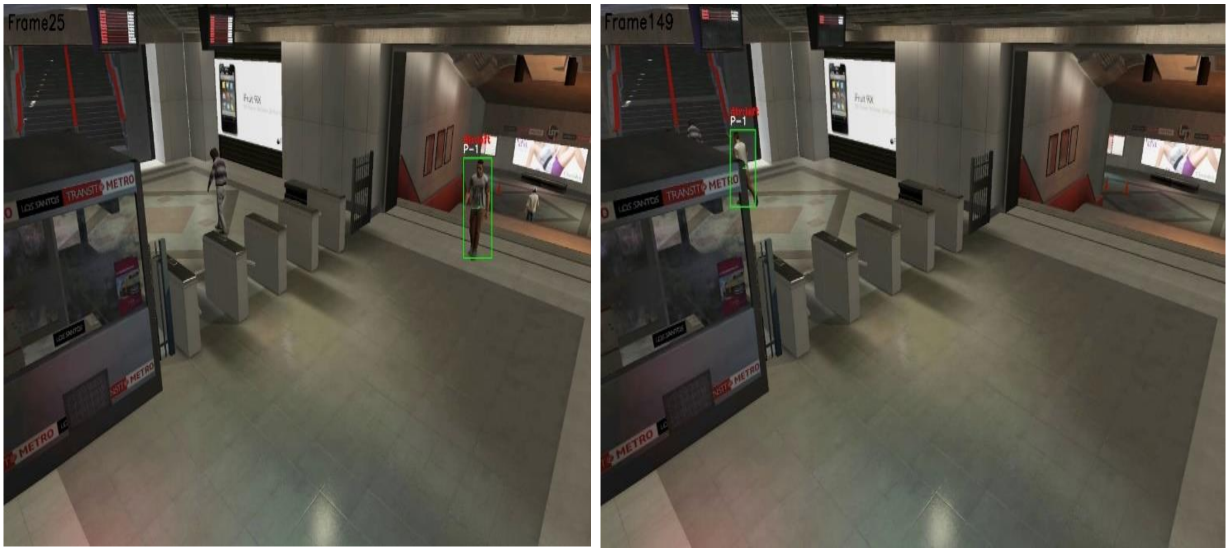


a.(C0.V1.F4) b.(C0.V1.F49)

Figure(4.20):Where a. frame(no.3) presents the scenario when re-identifying the same target in neighbor camera(C0) and in b. frame (no.348) the target walking to leave the scene of same camera predicting last direction to the left side.

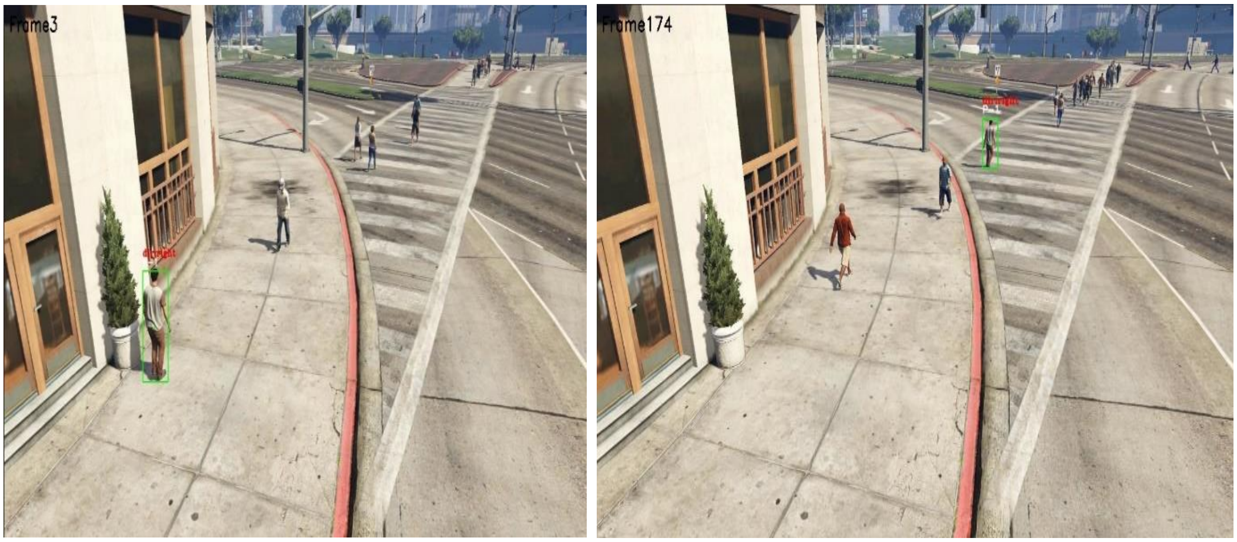
### Case Study 4

In this case, evaluating the re-identification model using sample videos from dataset of MTA dataset. When the system using YOLOv3 to detect the target(person) then tracking based on Kalman filter with deep sort algorithm in scene. In this case the target appears at first in camera (C0), the prediction result of direction for target refers to the left side, there are one camera appears in topology table in that side with distance (144m). The system applies the re-identification model to recognize if the target appears in this camera or not. In this case the target does not appear in the camera no.1 the last choice for system in this case is search on target in all neighbors' cameras for re-identifying the target in this scenario the target appears in camera no.4. Figure(4.21) presents the C1 when the target detected and tracked in the scene and Figure(4.22) present the C4 when the reidentification model recognizes the target in a neighbor's camera.



a.(C0.V1.F25) b.(C0.V1.F149)

Figure(4.21): a. frame(no.25) presents the scenario when target appears in (C0) walking to the left side and b. presents the target when leaves the camera view in frame(no.149) to the left side.



C4.V1.F3 C4.V1.F174

Figure(4.22): Frame(no.3) presents the scenario when re-identifying the target in neighbor camera(C4) and in b. frame (no.174) the target walking in the scene with predicted last direction to the right side.

Table (4.16): Presents a sample of persons in many cameras (different videos lengths, all videos' dimensions 1920×1080) for evaluation re-identification stage(MTA dataset).

| No. | Direction | Neighbor’s cameras | Last direction of target before leave view | Candidate cameras depend on short distance | Dose target appear in scene | Second priority | Dose target appear in next camera | True prediction true identify target TP | False prediction but true target FP | True predicted but wrong target TN | False prediction Misses target FN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | L | C2,C3 | L | C2 | no | C3 | no | 0 | 0 | 1 | 0 |
| V1 | | | L | C2 | yes | ---- | ---- | 1 | 0 | 0 | 0 |
| S= | | | L | C2 | no | C4 | yes | 1 | 0 | 0 | 0 |
| 6 | | | S | ---- | no | ----- | ---- | 0 | 1 | 0 | 0 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | R | C0 | R<br>R | C0<br>C0 | yes<br>yes | -----<br>----- | -----<br>------ | 1<br>1 | 0<br>0 | 0<br>0 | 0<br>0 |
| | S | C4,C5 | | | | | | | | | |
| | N | ------ | | | | | | | | | |
| | | | | | | | | | | | |
| C0<br>V1<br>S=<br>1 | L | C1 | | | | | | | | | |
| | R | C4,C5 | R | C4 | no | C5 | yes | 1 | 0 | 0 | 0 |
| | S | ----- | | | | | | | | | |
| | N | ----- | | | | | | | | | |
| C2<br>V1<br>S=<br>2 | L | C3 | | | | | | | | | |
| | R | C1 | R | C1 | yes | ---- | ---- | 1 | 0 | 0 | 0 |
| | S | C4,C5 | S | C5 | yes | ----- | ----- | 1 | 0 | 0 | 0 |
| | N | ----- | | | | | | | | | |
| C4<br>V1<br>S=<br>1 | L | C0 | | | | | | | | | |
| | R | C5 | | | | | | | | | |
| | S | C5 | | | | | | | | | |
| | N | C1,C2<br>C3 | N | C3 | no | C2 | yes | 1 | 0 | 0 | 0 |

Table (4.17): The confusion for re-identification model of MTA dataset

| **Total samples(persons) =10** | | **Actual re-identify** | |
|---|---|---|---|
| **Prediction re-identifies** | | **Positive** | **Negative** |
| | **Positive** | **TP=8** | **FP=1** |
| | **Negative** | **FN=0** | **TN=1** |

**Precision** = TP/(TP+FP) =8/(8+1) = **0.88%**

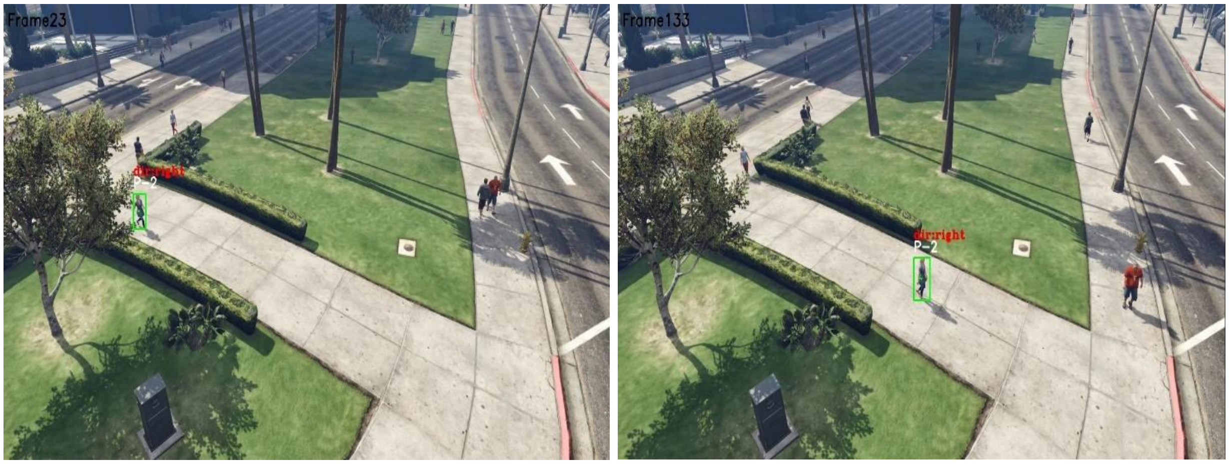
**Recall** = TP/(TP+FN)=8/(8+0) = **100%**

**F _score** = 2*(precision*recall/precision + recall) =2(0.88*1/0.88+1) = **0.94%**

**Accuracy** = (TP+TN)/(TP+TN+FP+FN)=8+1/10 = **0.90%**

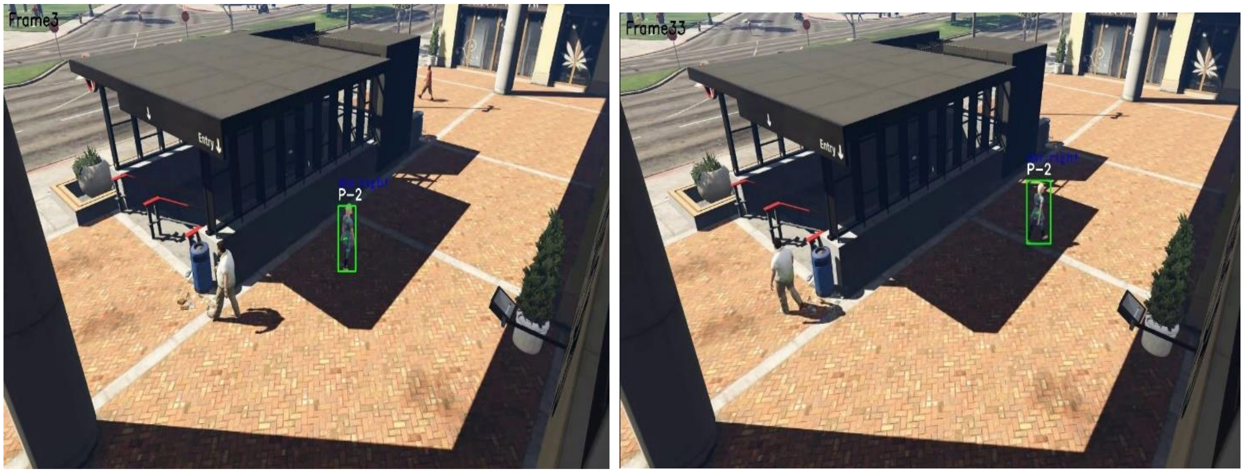
### 4.4.4.3 Subjective Evaluation Results of Re-identification Stage

The subjective results represented in several video frames from two cameras, where the target appears in camera no.2 in frame no.23 then the target walking in scene to the right side as shown in Figure(4.23). The distance between two cameras are 125 m and the time about(83sec).The proposed system will check the cameras on side right with respect to camera no.2 from topology table there are one camera(C1) on that side. When target enters the felid of view for camera no.1 the re-identification model will identify the same target. Figure (4.24) presents the target when appears in C1 at frame no.3 then continues walking in scene.



a.(C2.V1.F23) b.(C2.V1.F133)

Figure(4.23): a. frame(no.23) presents the scenario when target appear in (C2)walking to the right side and b. presents the target is walking in the camera view in frame(no.149) to the right side.



a.(C1.V1.F3) b.(C1.V1.F33)

Figure(4.24): a. frame(no.3) presents the scenario when the target appears in neighbor camera(C1), in b. frame (no.33) the target is walking in the scene with predicting last direction to the right side

## 4.5 Execution Time

The software of proposed system applied under Windows 10 operating system and python language(3.8) using with computer specifications as mentioned in section 4.2 , the items below explain the time complexity of proposed system:

The time complexity when apply CPU with MTA dataset was 0:05:55.013.

The max memory usage in megabytes is :922.

The time complexity when apply GPU with MAT dataset was 0:03:48.022.

The max memory usage in megabytes is:1170.

The time complexity when apply CPU with *Abbasid shrine in Karbala city* dataset was 0:06:11.013.

The max memory usage in megabytes is: 1000.

The time complexity when apply GPU with *Abbasid shrine in Karbala city* dataset was 0:03:17.427

The max memory usage in megabytes is: 1196.

## 4.6 Time Complexity

Time complexity can be calculated for each stage of the proposed system:

For object detection stage when YOLOv3 used which is consider a kind of CNN models. In a CNN, the number of features in each feature map is at most a constant time the number of input pixels n (typically the constant is < 1). Convolving a fixed size filter across an image with n pixels takes O(n) time, since each output is just the sum product between k pixels in the image, and k*k weights in the filter, and k doesn't vary with n. Similarly, any max or avg pooling operation doesn't take more than linear time in the input size. Therefore, the overall runtime is still linear. The stage is identifying persons in YOLO, the time of complexity is linear, where the image is passed once, and all objects are discovered once by using (YOLOv3)

$O(1) + O(1) + O(N) + O(N) + 6*O(1) + O(N) = O(N)$

The previous complexity is in the case for a single frame, and thus the complexity of the video increases O(N*M). Where M is the number of frames in the video.

In the tracking stage the system used Deep-SORT Tracking, before knowing the complexity calculation, we must discuss how does it work:

1- First all the objects are detected in the image.

2- Second, existing tracks positions are updated using a Kalman filter.

3- Then, they cluster the tracks by age (how long the tracks as not been associated with a detection) and run the Hungarian algorithm on each of the cluster in increasing age order All the remaining unmatched and unconfirmed tracks of age 1 are processed using the original SORT algorithm.

4- Finally, un-matched detection is set as new tracks.

5- The cost used for the first matching step is set as a combination of the Mahalanobis and the cosine distances. The Mahalanobis distance is used to incorporate motion information and the cosine distance is used to similarity between two objects. To compute the cosine distance, the rely on a CNN to compute the appearance descriptors (no precision on to how to train the network, but weights are provided).

6- The greatest time complexity of the conventional neural network.

O(N*L*Fps) where:

N = number of pixels

L = length video

Fps = number of frames per second

In this stage, the objects are tracked with the previous frames and this complexity is on the level of one frame.

O(N*K) + O(K*K) + O(K) + O(N) = O(N*K)

The previous complexity is in the case of a single frame, and thus the complexity of the video increases, O(N*K*M),Where M is the number of frames in the video and k number of tracks.

And K can be neglected where K<<M & K<<N : thus O(N*M)

In the algorithm of reduction trajectory points the complexity is linear where all steps cost O(1) and all steps are repeated for all trajectory points (T).

O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(T) = O(N)

For prediction stage LSTM was used at this stage and the complexity is explained in steps follows. The complexity for LSTM in real time(Test Time Complexity) is O(I*K*H) where:

I = the number of inputs

K = the number of outputs

H = the number of cells in the hidden layer

I<<H & K<<H ➔ O(H)

The complexity construction for LSTM (Train Time Complexity) is O(S*I*K*H) where :

S = the number of samples

I = the number of inputs

K = the number of outputs

H = the number of cells in the hidden layer, I<<H & K<<H ➔ O(S*H)

Knowing the object's orientation within the frame is computed by the difference between the two centers of bounding boxes for previous and next position of the object, so the complexity is O(1).The step is repeated for all objects as: O(1) + O(P) + O(1) = O(P)

Thus, the complexity of the video is: O(P*M) and P can be neglected as P<<M then the complexity is O(M)

For re-identification stage the time complexity can be calculated for training and testing model as:

Training: O(N) + O(1) + O(N) + O(epochs)*[O(in*H) + O(H*Out)] + O(N) = O(N*N*N)

Testing: O(N) + O(1) + O(N) + O(in*H*) + O(H*Out) + O(N) = O(N*N)

The complexity of the whole system during execution is the greatest complexity obtained is square : O(N*M)

And the complexity during training neural networks is cube : O(N*N*N) and changing the number of hidden layers or the structure of the neural network will change the complexity based on the chosen neural network.

## 4.7 Summary

This chapter summarizes and shows all evaluation results subjectively and objectively based the performance measures for all the proposed algorithms that have been applied on the SCNS' stages and hence, an excellently scores have been outed on. Moreover, the SCNS is evaluated in general, and it is achieved that it is possible to apply it for indoor and outdoor area as to the purpose of its design.

# CHAPTER FIVE

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 Conclusions

Occluded and disappear the moving object(person) from the scene in cameras network is considers one of the major causes of poor performance of surveillance cameras system because the target occluded by another object or may leave the scene to enter the blind area between cameras and that needs search on the target in all neighbor's cameras that will need more processing and more consume time.

During the period of this study, there are several notes that can be concluded as an outcome of this study:

1- The selected detection algorithm and tracking algorithm were applied in the proposed system are giving satisfy performance, where the algorithms are deal with short and long period of occlusions through keeping the track of target to reduce the identity switches. They give excellent result to next stages in the proposed system.

2- LSTM is a suitable choice for keeping a history of states to predict the motion of a moving single object that supports the Deep sort and Kalman filter to track the object across the camera's network because the prediction task improved the tracking process and select the correct direction in the scene and across cameras network.

3- Normalize the trajectory points depends on change of angle (threshold value) through keep just the interested points that increased the performance of prediction model, because the interested point gives better index on the change of direction for moving target in scene.

4- Increase the lookback steps for previous positions of moving target effect on the performance of LSTM model because the previous positions values which has distance location about the current position will effect on the prediction result (high change in direction of target moving after more numbers of steps).

5- Predicting the direction of the target's movement in the scene or when it leaves the camera's preview boundary, depending on the LSTM model and the Detection direction of the moving person Algorithm helped a lot in detecting which next camera may be the target reappear.

6- Training of re-identification model on different scenarios, change in illuminations and pose that increase the performance of the model to identify the same target where moving across multiple cameras in the network.

7- The topology of cameras represented by spatial relations between cameras in the network supports to detected of the neighbor's cameras based on the predicted direction of the target to solve the handoff problem between cameras.

## 5.2 Recommendations for Future Works

Therefore, there are many directions that can be suggested in the future work. The following are some of these directions.

1- Monitoring target used tracking by detection method , there is a suggestion to use new techniques for object detection which may have faster and more efficient with accurate results.

2- Applying the proposed model at nighttime also.

3- There is a suggestion to apply the system with moving multiple objects also.

4- There is a suggestion to apply the system with another class of moving object also such as animals or cars etc.

5- There is a suggestion to add feature such as gait or face recognize feature to increase the identification in situations when the target is walking toward the viewfinder

# References

[1] Quoc Cuong Le and Donatello Conte, “Online Multiple View Tracking: Targets Association Across Cameras. Workshop on Activity Monitoring by Multiple Distributed Sensing (AMMDS)”, HAL Id: hal-01880374, 2018.

[2] Pawan Kumar Mishra and G. P. Saroha, “A Study on Video Surveillance System for Object Detection and Tracking”, IEEE, (INDIACom), ISSN 0973-7529, ISBN 978-93-80544-20-5,2016.

[3] Olly Styles, Tanaya Guha and Victor Sanchez, “Multi-Camera Trajectory Forecasting: Pedestrian Trajectory Prediction in a Network of Cameras”, IEEE, https://DOI 10.1109/CVPRW50498.2020.00516, 2020.

[4] Malik Souded, “People Detection, Tracking and Re-identification Through a Video Camera Network”, PHD Thesis, University Sophia Antipolis, 2014.

[5] Xurshedjon Farhodov, Kwang-Seok Moon, Suk-Hwan Lee, “LSTM Network with Tracking Association for Multi-Object Tracking” Journal of Korea Multimedia Societ , Vol.23, Issue 10, Pages.1236-1249 , 2020.

[6] Yongyi Lu, Cewu Lu and Chi-Keung Tang, “Online Video Object Detection using Association LSTM”, IEEE,2017.

[7] Fanar Ali and Tawfiq A. Al-assadi, “Human Body Identification and Tracking based on Local Texture and Shape Features”, Dissertation , University of Babylon, 2019.

[8] Hasan Thabit and Israa Hadi, “Intelligent Video Surveillance System for Traffic Violation Detection based on User Defined Rules in Multi-Camera Network”, Dissertation, University of Babylon, 2021.

[9] Kristina Host a , Marina Ivašić-Kos b and Miran Pobar c , “Tracking handball players with the Deep SORT algorithm”, DOI:10.5220/0009177605930599, Published in ICPRAM 2020.

[10] Tuan Linh Dang , Gia Tuyen Nguyen and Thang Cao, “Object Tracking Using Improved Deep Sort Yolov3 Architecture”, ICIC Express Letters, Volume 14, Number 10, October 2020.

## References

[11] Yang Jie , Lilian Asimwe Leonidas and Munsif Ali, “Ship Detection and Tracking in Inland Waterways Using Improved YOLOv3 and Deep SORT, https://doi.org/10.3390/sym130203082021, 2021.

[12] Harish Chaandar and Avnish Kumar, “Learning and Predicting Sequential Tasks Using Recurrent Neural Networks and Multiple Model Filtering”, Association for the Advancement of Artificial Intelligence (www.aaai.org), 2016.

[13] Pu Zhang, Wanli Ouyang and Pengfei Zhan, 2017, “SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction”, IEEE, DOI: 10.1109/CVPR.2019.01236, 2020.

[14] Ehtesham Hassan, “Multiple object tracking using feature fusion in hierarchical LSTMs” , Journal of Korea Multimedia Society, Vol. 23, No. 10, October 2020 (1236-1249).

[15] Xurshedjon Farhodov, Kwang-Seok Moon, “LSTM Network with Tracking Association for Multi-Object Tracking ”,The Journal of Engineering, doi: 10.1049/joe.2020.0111 www.ietdl.org, 2020.

[16] Yeong-Jun Cho, Su-A Kim and Jae-Han Park, “Joint Person Re-identification and Camera Network Topology Inference in Multiple Cameras” , Elsevier Inc, 2019.

[17] F´abia Isabella Pires Enembreck, Erikson Freitas de Morais and Marcella Scoczynski “Person re-identification using Convolutional Neural Network and Autoencoder embedded on frameworks based on Siamese and Triplet networks” , DOI:10.21203/rs.3.rs-117671/v1, November 2020.

[18] Mianjin Wei and Jihong Pei, “Pedestrian Tracking Combined with Deep Learning and Camera Network Topology in Non-overlapping Multi-camera Surveillance”, IEEE, 2020

[19] Mingli Song**,** Dacheng Tao and Stephen J, “Sparse Camera Network for Visual Surveillance -- A Comprehensive Survey", IEEE,2013.

[20] Dimitrios Makris, Tim Ellis and James Black, “Bridging the gap between cameras", IEEE Computer Vision and Pattern Recognition, 2004.

[21] Shu Zhang, Yingying Zhu and Amit Roy-Chowdhury, “Tracking multiple interacting targets in a camera network” journal homepage: www.elsevier.com/locate/cviu, 2015.

## References

[22] Anton Thelandersson and Ólafur Már Óskarsson, "Topology Inference for Non-Overlapping Camera Networks", Thesis, Lund university, 2019.

[23] Youlu Wang, "Distributed Multi-object Tracking with Multi-camera Systems Composed of Overlapping and Non-overlapping Cameras", Dissertation, 2013.

[24] Amanjot Singh1 and Jagroop Singh, "Comparative Analysis of Gaussian Filter with Wavelet Denoising for Various Noises Present in Images ", *Indian Journal of Science and Technology*, Vol 9(47), December 2016.

[25] Zhong-Qiu Zhao, Peng Zheng and Shou-tao Xu, "Object Detection with Deep Learning: A Review", IEEE, 2019.

[26] Licheng Jiao , Fan Zhang, Fang Liu and Shuyuan Yang, "A Survey of Deep Learning-based Object Detection", IEEE, 2019.

[27] Zhengxia Zou, Zhenwei Shi and Yuhong Guo, "Object Detection in 20 Years: A Survey", IEEE, 2019.

[28] Joseph Redmon, Santosh Divvala and Ross Girshick, "You Only Look Once: Unified, Real-Time Object Detection", IEEE, 2016.

[29] Joseph Redmon, Ali Farhadi University of Washington , "YOLOv3: An Incremental Improvement", Computer Vision and Pattern Recognition (cs.CV), 2018.

[30] Dr. S.V. Viraktamath and Madhuri Yavagal, "Object Detection and Classification using YOLOv3", International Journal of Engineering Research & Technology (IJERT), Vol. 10, Issue 02, February-2021

[31] Liquan Zhao and Shuaiyang Li , "Object Detection Algorithm Based on Improved YOLOv3", Electronics MDPI, https://doi.org/10.3390/electronics9030537, 2020.

[32] Rojasvi G.M and Dr Anuradha S G, "Object Detection and Tracking for Computer Vision Applications" International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), Vol 5, Issue 4, April 2018.

# References

[33] Himanshu Chandel and Sonia Vatta, “Occlusion Detection and Handling: A Review”, International Journal of Computer Applications, Vol. 120, No.10, June 2015.

[34] Caglayan Dicle, Octavia I. Camps and Mario Sznaier, “The Way They Move: Tracking Multiple Targets with Similar Appearance”, IEEE, DOI 10.1109/ICCV.2013.286, 2013.

[35] Suresh Kumar K Dr, “Occlusion Detection with Background Elimination and Moving Object Tracking”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Vol.8 Issue-4, November 2019.

[36] Changlin Xiao, “Visual Tracking with an Application to Augmented Reality”, The Ohio State University, PhD Dissertation, 2017.

[37] Zahra Soleimanitaleb, Mohammad Ali Keyvanrad and Ali Jafari, “Object Tracking Methods:A Review”, 9th International Conference on Computer and Knowledge Engineering (ICCKE), October 24-25 2019.

[38] Shixion Xia, Jiqi Ziaqizhao and Yong Zhou, “ Video Object Segmentation and Tracking: A Survey” , Article, 39 pages. DOI: 0000001.0000001,2019

[39] Afef Salhi, Fahmi Ghozzi and Ahmed Fakhfakh, “Estimation for Motion in Tracking and Detection Objects with Kalman Filter”, DOI: 10.5772/intechopen.92863, 2020.

[40] Hui Li, Yun Liu, Chuanxu Wang and Shujun Zhang, “Tracking Algorithm of Multiple Pedestrians Based on Particle Filters in Video Sequences”, Computational Intelligence and Neuroscience, https://doi.org/10.1155/2016/8163878, 2016.

[41] Jos Elfring ,Elena Torta and René van de Molengraft , “Particle Filters: A Hands-On Tutorial”, MDPI, https://doi.org/10.3390/s21020438 , 2021.

[42] Claudio Urrea and Rayko Agramonte, “Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation”, *Journal of Sensors,* 2021.

[43] Liana Taylor, Midriem Mirdanies and Roni Permana, “ Optimized object tracking technique using Kalman filter”, *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, 2016.

## References

[44] Emilio Maggio, Andrea Cavallaro, “Video Tracking Theory and Practice”, book, 2011.

[45] Gioele Ciaparrone1,2, Francisco Sánchez and Siham Tabik “ Deep learning in video multi-object tracking : A survey”, Neurocomputing, https://doi.org/10.48550/arXiv.1907.12740, 2019.

[46] Axel Nyström, “Evaluation of Multiple Object Tracking in Surveillance Video”, Master Thesis, Linköping University, 2019.

[47] Nicolai Wojke , Alex Bewley and Dietrich Paulus, “Simple Online and Realtime tracking with a Deep association Metric”, https://doi.org/10.48550/arXiv.1703.07402, 2017.

[48] Yu Zheng and Xiaofang Zhou, “Computing with Spatial Trajectories”, Book, 2011.

[49] Dieter Pfoser, Christian S. Jensen and Yannis Theodoridis, “Novel Approaches to the Indexing of Moving Object Trajectories”, Proceedings of the 26th VLDB Conference, Cairo, Egypt, 395-406, September 10-14, 2001.

[50] Minshi Liu, Guifang He and Yi Long, “A Semantics-Based Trajectory Segmentation Simplification Method”, *Journal of Geovisualization and Spatial Analysis* , 2021.

[51] Adnan Brdjanin, Nadja Dardagan and Dzemil Dzigal "Single Object Trackers in OpenCV: A Benchmark", IEEE, 2020.

[52] Nadeem Anjum and Andrea Cavallaro, “Multi-feature object trajectory clustering for video analysis” IEEE, 2008.

[53] R.Manikandan1 and R.Ramakrishn, “Human Object Detection and Tracking using Background Subtraction for Sports Applications”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 10, October 2013.

[54] Zhiquan He, Hao Sun and Wenming Cao, “Multi-level context-driven interaction modeling for human future trajectory prediction”, Neural Computing and Applications, 2022

[55] Daniel Amigo, David Pedroche and Jesu´ s Garcı´a, “Review and classification of trajectory summarization algorithms: From compression to segmentation”, *International Journal of Distributed Sensor Networks*, Vol. 17 (10), 2021.

## References

[56] Omar Moured, “evaluation of deep learning based multiple object trackers”,  Master Thesis, Middle east technical university, 2020.

[57] Noor Dhia and Israa Hadi, “Driver Drowsiness Detection Based on Spatiotemporal Features with 3D Convolutional Neural Networks” , Dissertation, University of Babylon,2020.

[58] Peter Ghavami, “Machine Learning, Deep Learning—Artificial Neural Networks”, Book ,Chapter 8, 2020.

[59] Xurshedjon Farhodov, Kwang-Seok Moon and Suk-Hwan Lee “LSTM Network with Tracking Association for Multi-Object Tracking” , *Journal of Korea Multimedia Society* Vol. 23, No. 10, October 2020(pp. 1236-1249), 2020.

[60] Mohammad Hossein Nasseri, Hadi Moradi and Reshad Hosseini, “ Simple Online and Real-Time Tracking with Occlusion Handling”, Corpus ID: 232147229,  2021.

[61] Ruben Villegas and Jimei Yang, “Decomposing motion and content for natural video sequence prediction”, Published as a conference paper at ICLR 2017.

[62] Martina Tichá , “Long-Term Person Re-Identification in Video”, Master Thesis, Czech Technical University in Prague, 2019.

[63] Asmat Zahraa , Nazia Perwaiza and Muhammad Shahzad “Person Re-identification: A Retrospective on Domain Specific Open Challenges and Future Trends” , Elsevier, 2022.

[64] Kevin C and Krucki, “Person Re-identification in Multi-camera Surveillance Systems”, Master Thesis, Dayton University, 2015.

[65] Wenyu Wei, Wenzhong Yang and Enguang Zuo a , “Person Re-identification based on Deep Learning — An overview”,  *Journal of Visual Communication and Image Representation*, Vol.82, January 2022.

[66] Marta de Oliveira Barreiros, Diego de Oliveira Dantas, Luís Claudio de Oliveira Silva, “Zebrafish tracking using YOLOv2 and Kalman flter”, Scientifc Reports |https://doi.org/10.1038/s41598-021-81997-9, 2021.

# الخلاصه

يعتبر مجال أنظمة كاميرات المراقبة مجالًا مهمًا للبحث في السنوات الأخيرة لأنه يستخدم لأغراض مختلفة مثل مراقبة المباني الكبيرة والشوارع والأماكن المثيرة للاهتمام وما إلى ذلك. يمكن لأنظمة المراقبة تحليل محتوى الفيديو بذكاء واكتشاف السلوكيات غير العادية ، يمكن لهذه التقنيات الجديدة أن توفر مراقبة أكثر دقة وأمانًا من خلال دعم الشرطة والأمن في حالات التحقيق. لا يزال التنبؤ باتجاه جسم متحرك يمثل تحديًا لأن الكائن يغير مظهره أو وضعه أوحجبه. يؤدي اختفاء الهدف (الشخص) من المشهد إلى ضعف أداء نظام المراقبة لأن النظام يفقد المسار. ستحتاج إعادة تحديد  نفس الهدف في شبكة الكاميرا إلى البحث عن الهدف في جميع الكاميرات المجاوره وسيحتاج ذلك إلى مزيد من المعالجة واستهلاك المزيد من الوقت. يتمثل حل هذه المشكلة في زيادة الارتباط بين الحالات لاعادة تحديد نفس الهدف بعد الحجب والتنبؤ بالاتجاه الصحيح للهدف لتعيين الكاميرا التالية المحددة حيث قد يظهر الهدف مرة أخرى.النظام المقترح في هذه الرسالة هو تتبع والتنبؤ باتجاه حركة الهدف عبر الشبكه وهو يجمع بين تقنيات التعلم العميق والمعلومات المكانية من طوبولوجيا شبكة الكاميرا. يتكون النظام المقترح من عدة مراحل: أولاً ، تقسيم الفيديو إلى إطارات متتالية واستخراج الميزات المهمه للهدف باستخدام خوارزمية YOLOv3 لاكتشاف الهدف. ثانيًا ، تتبع الهدف باستخدام مرشح كالمان والفرز العميق ثم إنشاء نقاط مسار للهدف المتحرك. ثالثًا ، توقع اتجاه الهدف المتحرك باستخدام الخوارزمية المقترحة مع نموذج LSTM لربط الميزات المهمة للهدف عبر الاطارات المتسلسلة. أخيرًا ، استخدام نموذج إعادة تحديد هوية الشخص لإعادة تحديد نفس الهدف عبر الكاميرات المجاورة اعتمادًا على ميزات الهدف ، ونتيجة التنبؤ للاتجاه والمعلومات المكانية والزمانية من طوبولوجيا شبكة الكاميرا لاكتشاف المكان الذي قد يظهر فيه الهدف مرة أخرى.

تم اعتماد مجموعتين من البيانات في هذه الرسالة هما "مجموعة بيانات الفيديو متعددة الكاميرات (MTA) التي تقدمها مؤسسة رؤية الكمبيوتر والضريح العباسي في مجموعة بيانات مدينة كربلاء. تحتوي مجموعة بيانات MTA على ست كاميرات (داخلية وخارجية) موزعة في مجالات عرض متداخلة وغير متداخلة ؛ تحتوي مجموعة البيانات الثانية على ست كاميرات مع توزيع غير متداخل. يحتوي نوعان من مجموعات البيانات على عدد من مقاطع الفيديو مع سيناريوهات مختلفة للأشخاص المتحركين في ظل ظروف الإضاءة المختلفة. أخيرًا ، كانت نتائج تقييم نماذج التنبؤ وإعادة تحديد الهوية لـ MTA 0.89 و 0.90 على التوالي ومجموعات بيانات الضريح العباسي 0.91 و 0.86 على التوالي. النظام المقترح يحقق دقة تصل إلى 90٪.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل

كلية تكنولوجيا المعلومات

قسم البرمجيات

# نظام تنبؤ مطور لأتجاه الكائن المتحرك وتتبعه عبر طوبولوجيا غير متداخله لشبكة كاميرات المراقبه باستخدام التعلم العميق

أطروحة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات/ جامعة بابل كجزء من متطلبات
الحصول على درجة الدكتوراه فلسفة في تكنولوجيا المعلومات/ برمجيات

من قبل

**وائل مهدي بريج جلغان**

بإشراف

**أ.د. اسراء هادي علي حسين**

1444 هـ

2022