YOLO-Based Multi-Object Tracking with Reduced Order Observer for Target Trajectory Estimation

Author

MUHAMMAD NOUMAN

MS-21 (EE)

Regn Number

00000397991

Supervisor

DR. FAHAD MUMTAZ MALIK

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD, PAKISTAN

June 2024

THESIS ACCEPTANCE CERTIFICATE

It is certified that final copy of MS/MPhil thesis written by Mr. Muhammad Nouman (Registration No. 00000397991) Entry-2021, of (College of E&ME) has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors, and mistake and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC member of the scholar have also been incorporated in the said thesis.

Signature: ______________________

Name of Supervisor Dr. Fahad Mumtaz Malik

Date: 05 Aug 2024

Signature (HoD): ______________________

Dr. Qasim Umar Khan

Date: 5/8/24

Rashid

Signature (Dean) ______________________

Brig Dr. Nasir Rashid

Date: 05 AUG 2024

DEDICATION

Dedicated to my exceptional family and teachers whose unwavering support, guidance and encouragement has been instrumental in my journey.

ACKNOWLEDGEMENTS

I am thankful to my Creator Allah Almighty for guiding me throughout this work at every step. Indeed, I could have done nothing without Your priceless help and guidance. I am empowered to Read and Write only by Him, who has bestowed upon me the knowledge I carry forward.

I would also like to express special thanks to my supervisor Dr. Fahad Mumtaz Malik for his help throughout my thesis and knowledge imparted through the courses he taught me. I would also like to pay special thanks to Dr. Shahzad Amin Sheikh and A/P Kamran Aziz Bhatti for their tremendous support and cooperation.

I sincerely express my solemn gratitude with earnest sense of reverence to my parents for their encouragement, heartfelt prayers and kind wishes for successful completion of my studies along with this research work. Specially, I want to thank my spouse and children from all my heart for always supporting and motivating me at every point of my research journey.

Finally, I would like to express my appreciation to all the individuals who have rendered valuable assistance to my study.

Muhammad Nouman

ABSTRACT

This thesis presents an innovative and efficient approach to track multiple objects in a video sequence by integrating state-of-art You Only Look Once (YOLO) object detection framework with efficient Reduced Order Observer (ROO) based target trajectory estimator. Traditionally, Kalman Filter is utilized to estimate detected objects in video frames, however, this approach is computationally demanding due to repeated estimation of large number of state variables in each iteration. To address this challenge, ROO based trajectory estimator is proposed that focuses on estimating a subset of state vector, thereby enhancing processing speed for real-time applications.

A video sequence with frame rate of 30 frames per second is fed into YOLO model which outputs bounding box defined by a state vector for each detected object in a frame. State vector elements include position (x and y axis), velocity (x and y axis), aspect ratio and height of bounding box. State estimator is required to estimate states of these bounding boxes / detected objects in next frame of video sequence. ROO separates observable states from unmeasurable states and only estimates unmeasurable states. These estimates combined with Intersection Over Union (IoU) matching are used to assign bounding boxes / detected objects to tracklets ensuring efficient tracking even in the presence of occlusion and dynamic real-time environments.

This approach has been validated by implementing ROO framework in MATLAB R2023b and subsequently in Microsoft Visual Studio for online tracking of objects in video frame. This research contributes to the field of multi-object tracking by providing a computationally efficient trajectory estimator with potential applications in autonomous driving, surveillance and robotics.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ROO	Reduced Order Observer
YOLO	You Only Look Once
MOT	Multiple Object Tracking
SORT	Simple Online & Real-Time Tracking
CNN	Convolutional Neural Network
FrRCNN	Faster Region CNN
FPN	Feature Pyramid Network
JDE	Joint Learning of Detection and Embeddings
Re-ID	Re Identification
REN	Reciprocal Network
SAAN	Scale Aware Attention Network
BoT	Bag-of-Tricks
GSI	Gaussian Smoothed Interpolation
AFLink	Appearance-Free Link
SLM	Similarity Learning Module
SDE	Separate Detection and Embeddings
PSA	Patch Self Attention
SMC	Similarity Matching Cascade
IOU	Intersection Over Union
CSP-Net	Cross Stage Partial Network
NMS	Non Maximum Suppression
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
IDF1	Identity F1
IDP	Identity Precision
IDR	Identity Recall
IDSW	Identity Switch
MT	Mostly Tracked
ML	Mostly Lost
Frag	Fragments
FPS	Frames Per Second

CHAPTER 1

INTRODUCTION

1.1 Background

In computer vision, object tracking serves as fundamental component in applications like autonomous driving, surveillance systems, robotics etc. Object tracking is often carried out using track by detection technique in which objects are detected in a frame firstly and then associated with objects in next frame for tracklet assignment. However, to carryout assignment of object in detections of next frame, object positions need to be predicted. These predictions are largely carried out using Kalman Filter algorithm. Real-time performance in these tasks is crucial for the success of applications dependent on timely and accurate object tracking.

Kalman Filter is used for estimating the state of a moving object based on incomplete and noisy measurements. Though it is powerful, the Kalman Filter is computationally expensive to execute, especially for multiple objects or high-resolution video streams. The computational overhead associated with them means that they are not generally suited to real-time systems where resources can be scarce.

With the arrival of deep learning models, field of computer vision has powerful new tools to make object detection and tracking more robust. You Only Look Once (YOLO) is one of these tools that have created quite an interest in the field due to the accuracy at which it can detect objects in real-time. It is a full-image-sliding-window object detection network using YOLO with an added bonus - this model is capable of predicting bounding boxes and class probabilities for these boxes at the same time and in a single run, making it much faster than the previous approaches.

The approach put forward in this thesis combines a YOLO-based model for object detection with a Reduced Order Observer (ROO) for trajectory estimation. The ROO intends to lower the computational complexity by estimating only few of the state variables, hence making it computationally less expensive and real-time friendly.

While recent advancements in MOT have shown promise, a critical gap exists in addressing computational complexity and real-time performance. Current methods, while effective, often struggle to meet the demands of real-time applications due to high computational requirements. Integration of YOLO and Reduced Order Observer (ROO) for target estimation in the context of MOT has the potential to significantly reduce computational complexity while enhancing real-time performance. Exploration of this integration represents an unexplored avenue in the literature, presenting an opportunity to contribute to the development of efficient and responsive tracking system suitable for applications in surveillance, autonomous vehicles and beyond.

1.2 Problem Statement

Current challenge in computer vision and object tracking lies in balancing the computational complexity, real-time performance vis-à-vis tracking accuracy of MOT systems. While YOLO demonstrates efficient object detection, integrating it with Reduced Order Observer (ROO) for target estimation in tracking remains unexplored. Existing systems face issues of high computational demands, hindering real-time applications. This research aims to address the challenge by developing an integrated solution that significantly reduces computational complexity, ensuring enhanced real-time performance. The key focus is on seamlessly merging YOLO and ROO to create a comprehensive system, contributing to the evolution of efficient and responsive object tracker in dynamic scenarios.

1.3 Research Objectives

The specific objectives of this study include:

1. **Integration of YOLO and ROO:** To integrate the YOLO object detection framework with a Reduced Order Observer (ROO) for efficient multi-object tracking.
2. **Enhance Processing Speed:** To enhance the processing speed of trajectory estimation, making it feasible for real-time applications.
3. **Validation through Implementation:** To validate the proposed approach through implementation and experimentation in both MATLAB and Microsoft Visual Studio environments.

1.4 Thesis Outline

Chapter 1 provides an overview of the background, problem statement, objectives and organization of the thesis. Chapter 2 reviews relevant literature on object detection, YOLO, trajectory estimation techniques, and Reduced Order Observers. Chapter 3 entails methodology used in this research, including integration of YOLO and ROO, and use of IoU matching for tracklet assignment. Chapter 4 describes implementation of ROO framework in MATLAB and Microsoft Visual Studio and discusses system architecture and performance optimization. Chapter 5 presents experimental setup, dataset description, evaluation metrics, results and analysis. Chapter 6 is summary of findings and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Works

In computer vision (CV), multi-object tracking (MOT) involves frame-wise detection and tracking multiple objects in video sequences, aiming to identify and locate objects across frames despite challenges like occlusion, motion blur, and appearance changes. Various algorithms integrate object detection and data association techniques to address these challenges. Studies highlight the importance of effective data association methods for accurate tracking, such as Global Nearest Neighbor and Multiple Hypothesis Tracking algorithms [1]. Research emphasizes significance of incorporating multiple features for robust data association, improving track quality and accuracy, especially in complex scenarios like occlusion and fast motion [2]. Additionally, advancements in MOT include use of global appearance and motion models to enhance tracking efficiency and accuracy, achieving appreciable results on benchmark datasets.

Simple Online & Real-time Tracking (SORT) architecture used in paper [3] leverages the Faster Region CNN (FrRCNN) detection framework, specifically utilizing two network architectures: FrRCNN. FrRCNN consists of two stages where features are extracted and regions are proposed for object classification. This end-to-end framework allows for efficient detection by sharing parameters between stages and enhances detection performance with different network architectures.

Simple Online And Realtime Tracking With A Deep Association Metric introduces a pragmatic approach to MOT by introducing appearance information and an appearance metric which enhances performance under occluded environments [4]. Algorithm is trained on a deep association metric and achieves 45% reduction in switching of identities under occluded environments. Moreover, this viable performance is achieved at high frame rates. In order to achieve this performance and learning of appearance metric, a well discriminating embedding vector is trained offline and then used in tracking pipeline. For this purpose, this paper has trained a CNN on person re-identification dataset that contains over 1,100,000 images of 1,261 pedestrians. This training makes it suitable for learning / tracking pedestrians in complex environments.

Tracktor is a tracking system that attains exceptional tracking accuracy without requiring specialized training / optimization. Utilizing object detection and bounding box regression, Tracktor demonstrates proficiency in MOT in demanding situations like occlusions and motion prediction [5]. Re-identification, motion prediction and occlusion handling are few major problems encountered in MOT. Tracktor is successful in tracking objects without considering any of these tasks as its primary goal. Moreover, there is no involvement / requirement to optimize algorithm on tracking dataset. It explores depth of bounding box regression and estimates / tracks object locations in consecutive frames.

Towards Real-time Multi Object Tracking proposes “Joint Learning of Detection and Embedding” model [6]. This integration enhances efficiency and establishes a quicker baseline for real-time tracking applications. The architecture involves a Feature Pyramid Network (FPN) as the backbone network. The FPN makes predictions at multiple scales, which benefits detecting pedestrians of varying sizes in each frame of video sequence. Input video frame generates feature maps of 3 varying scales after it gets passed from backbone network. Rates of downsampling for these maps include 1/32, 1/16, and 1/8. Additionally, system incorporates prediction heads on multiple FPN scales. Classification of anchor, box regression and embedding vector learning are 3 important tasks treated by each prediction head. Losses linked with each task are tackled independently by adding randomness / uncertainty to each task. This approach optimizes learning process and improves overall efficiency of system. Furthermore, architecture includes an association algorithm that works with JDE. This method aims to reduce computational costs compared to previous MOT systems, thereby providing a foundation for future developments in real-time MOT algorithm design.

FairMOT balances biasness between detection and reID tasks in Joint Detection & Embedding Models [7]. Earlier work treats re-ID as secondary task and object detection as primary task. Resultantly, network is biased towards detection task (primary) and re-ID is ignored. To solve the problem, FairMOT presents anchor-free architecture CenterNet which eliminates biasness of network towards any task, may it either by re-ID or detection.

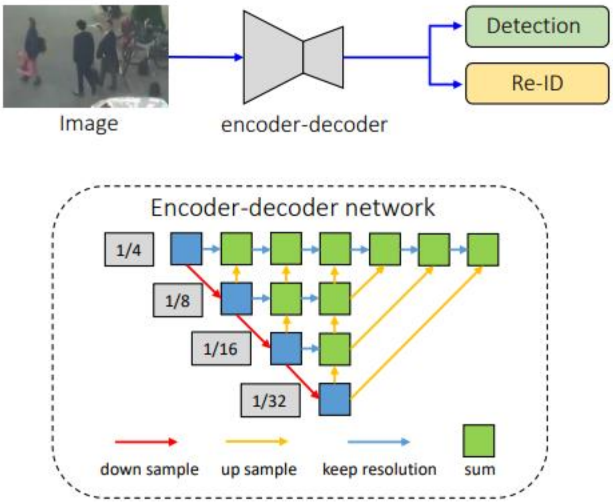

Figure 2.1 - FairMOT Architecture

Image Based Multi Objects' Trajectories Estimation through ROO assumes position of object, its appearance and rate of change of position as states and links MAP for object detection with ROO for trajectory estimation. Error in predicted and actual positions improves future estimation. Calculations are reduced by applying a Single object tracking model multiple times [8].

CSTrack addresses one of the most important challenge faced in one-shot tracking architype It states that re-ID and detection are treated as isolated tasks in this architype, contrary to two-stage trackers [9]. It proposes REN to tackle this problem. REN has:-

- Self-relation.
- Cross-relation design.

Resultantly, it learns representation based on re-ID / detection tasks properly. Further, a scale-aware attention network (SAAN) is also proposed in this paper. SAAN has fol feature:-

- Prevents misalignment of semantics.
- Improves association.

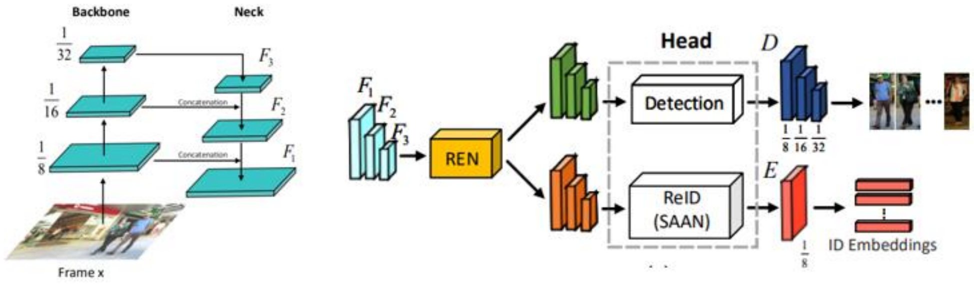

Figure 2.2 - CSTrack Architecture

Bag-of-Tricks (BoT) SORT [10] introduced:-

- Enhanced motion / appearance information.
- Camera-motion compensation.
- Kalman filter state vector with enhanced accuracy.

BoT SORT claimed improvement in performance of MOT by adding above mentioned corrections. Association between detected objects and tracks formulated can be made more robust by using IoU and fusing cosine-distances. BoT-SORT and BoT-SORT-ReID cannot estimate camera motion if background has high density objects. In such cases, tracker response can be erroneous. Moreover, if large images are processed, global motion compensation calculation can be extremely expensive.

Earlier methods only associate detection boxes with high confidence scores and objects detected with low confidence scores were simply thrown away. ByteTrack proved that by iteratively associating detection boxes with both high and low confidence scores, MOT efficiency can be enhanced. Fragments formed in MOT were also reduced and occlusion handling of architecture was also improved [11].

StrongSORT proposes two algorithms:-

- AFLink model for missed associations.
- GSI model for missed detections.

AFLink model conducts global association without requiring appearance information. This method is faster and accurate as compared to traditional methods which associate tracklets with trajectories at cost of high computations.

Gaussian-smoothed interpolation (GSI) is based on Gaussian process regression and addresses problem of missed detections [12].

SMILEtrack proposes two algorithms:-

Similarity Learning Module (SLM) which integrates Siamese network with object detector. It also works on evaluating similarity between appearance of objects and is an enhancement to feature descriptors of SDE models. SLM incorporates Patch Self-Attention (PSA) which is based on vision Transformer.

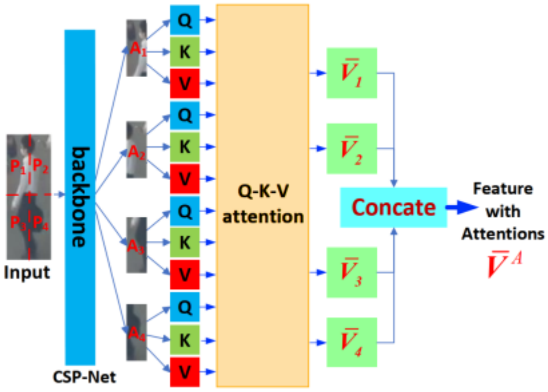

Figure 2.3 - SmileTrack Patch Self-Attention

Similarity Matching Cascade (SMC) is also proposed in this paper with a novel GATE function which improved association performance of tracker.module [13]. Earlier methods used weighted sum between IOU and appearance information for association of objects. In these methods IOU score can dominate the results. SMC GATE function addresses this problem by not allowing IOU score to dominate the weighted sum and rejecting associations with low appearance matching [13].

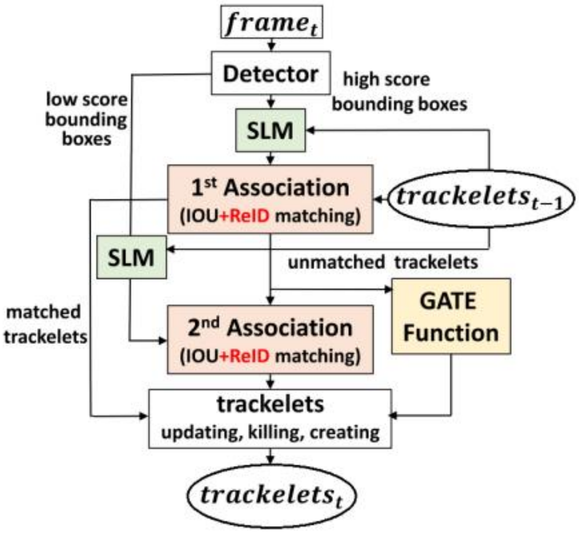
$frame_t$
Detector
high score bounding boxes
low score bounding boxes
SLM
1st Association (IOU+ReID matching)
$trackelets_{t-1}$
SLM
unmatched trackelets
matched trackelets
2nd Association (IOU+ReID matching)
GATE Function
trackelets updating, killing, creating
$trackelets_t$

Figure 2.4 - SmileTrack Association with GATE Function

SMILEtrack is an SDE method and performs MOT tasks faster than JDE methods.

Summarily, race between accuracy and inference speed is still on-going and in near future implementation of these architectures on small chips shall further enhance requirement to reduce computational complexity of these algorithms. Researchers have explored different avenues to reduce computational complexity while maintaining accuracy of these algorithms. Hence, a significant gap exists to further reduce computational complexity of these algorithms thereby easing their implementation on hardware chips in future.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, discussion is carried on methodology used to develop and evaluate the proposed YOLO-based multi-object tracking system with a Reduced Order Observer (ROO) for target trajectory estimation. The methodology is divided into several key components: system architecture, YOLOX-based object detection, ROO for trajectory estimation and integration of these components. This chapter also discusses implementation details, including software tools and frameworks used.

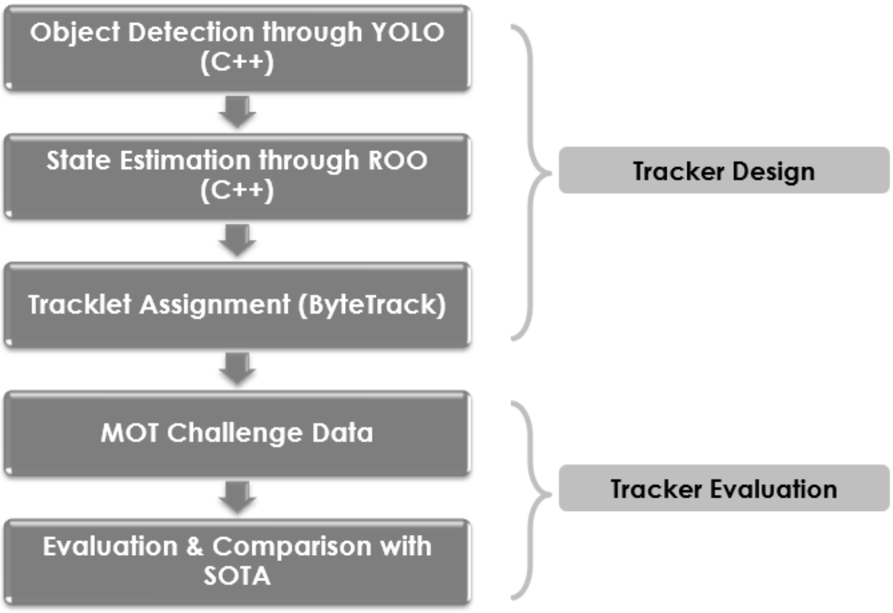
Object Detection through YOLO (C++)
State Estimation through ROO (C++)
Tracklet Assignment (ByteTrack)
MOT Challenge Data
Evaluation & Comparison with SOTA
Tracker Design
Tracker Evaluation

Figure 3.1 - Outline Methodology

3.2 System Architecture

The proposed system includes following components:

- Object Detection using YOLOX.
- Trajectory Estimation using Reduced Order Observer (ROO).
- Tracklet Management and Assignment.

Each component plays role in ensuring efficient and accurate MOT. Overall architecture processes video sequences in real-time and provides reliable tracking information for each detected object.

3.3 YOLO-Based Object Detection

YOLOX is an object detection framework that builds upon principles of original YOLO series and introduces several improvements [14], [15].

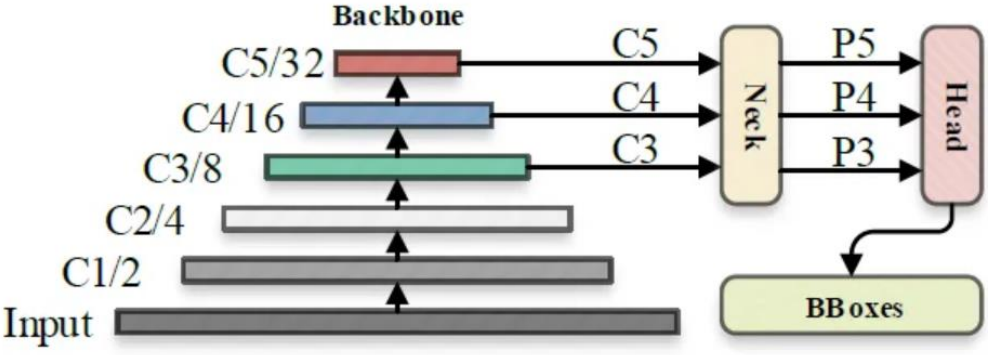

Figure 3.2 - YOLO v3 to v5

YOLOX leverages anchor-free detection and dynamic label assignment strategies as a result of which detection accuracy and inference speed are enhanced [16].

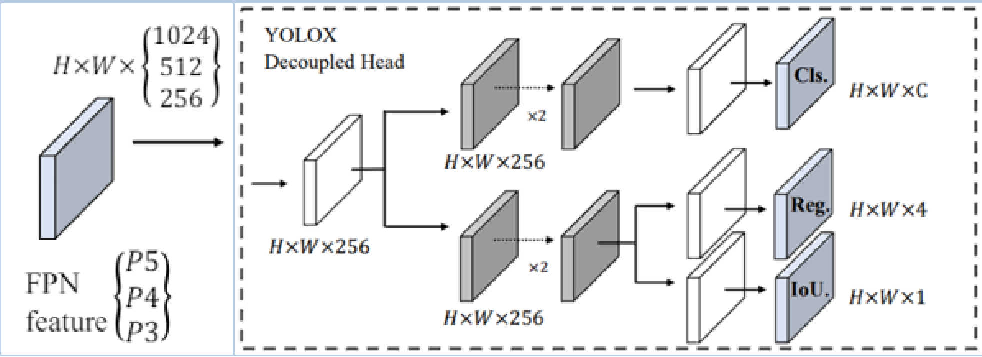

Figure 3.3 - YOLOX Architecture

3.3.1 Model Architecture

The YOLOX model architecture used in this research incorporates several key features:

- **Backbone**: Darknet-based architecture with CSPNet (Cross Stage Partial Network) integration, which balances between speed and accuracy.
- **Neck**: PANet (Path Aggregation Network) for enhanced feature fusion and improved detection performance.
- **Head**: Decoupled head design, separating classification and regression tasks to improve model convergence and accuracy.

3.3.2 Detection Process

The detection process involves the following steps:

- **Image Preprocessing**: Input video frames are re-sized and normalized to match input requirements of YOLOX model.
- **Forward Pass**: The preprocessed frames are fed to YOLOX model, which outputs coordinates / details of bounding boxes and class predictions of detected objects.
- **Post-Processing**: Non-Maximum Suppression (NMS) filters redundant bounding boxes.

3.4 Trajectory Estimation using Reduced Order Observer (ROO)

3.4.1 Concept of ROO

ROO estimates a subset of state variables, focusing on unmeasured states [17]. Resultantly, improving computational complexity and enhancing processing speed and suitability towards real-time applications is achieved.

3.4.2 System Dynamics and Modelling

System dynamics for trajectory estimation are defined as follows:

3.4.2.1 State Vector

The state vector includes following aspects of bounding box:

- Position in x
- Position in y
- Velocity v_x
- Velocity v_y
- Aspect ratio a
- Rate of change of aspect ratio
- Height h
- Rate of change of height

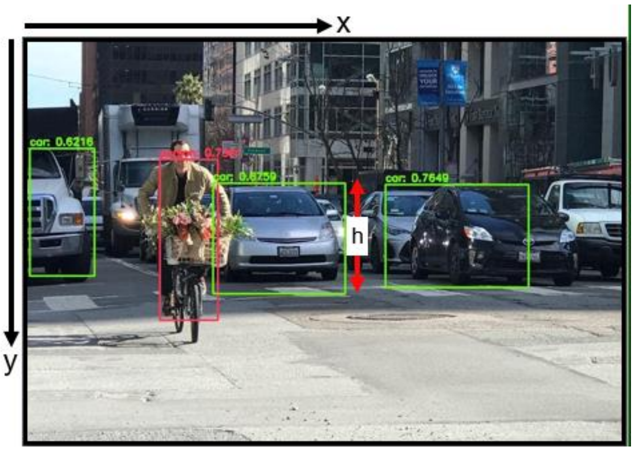

Figure 3.4 - State Description

3.4.2.2 System Dynamics

System Matrix

A = [1 dt 0 0 0 0 0 0;

0 1 0 0 0 0 0 0;

0 0 1 dt 0 0 0 0;

0 0 0 1 0 0 0 0;

0 0 0 0 1 dt 0 0;

0 0 0 0 0 1 0 0;

0 0 0 0 0 0 1 dt;

0 0 0 0 0 0 0 1];

Input Matrix

B = [0 dt 0 dt 0 dt 0 dt]’;

Output Matrix

C = [1 0 0 0 0 0 0 0;

0 0 1 0 0 0 0 0;

0 0 0 0 1 0 0 0;

0 0 0 0 0 0 1 0];

Feedforward Matrix

$$D = [\ 0\ ];$$

3.4.2.3 Separation into Measurable and Unmeasured System

State Vector

In the state vector mentioned above, measurable and unmeasurable states can be separated as under:

Measured

- x – Position in x-axis
- y – Position in y-axis
- a – Aspect Ratio (width / height)
- h – Object height

Unmeasureable

- v_x – Velocity in x-axis
- v_y – Velocity in y-axis
- v_a – Rate of change of aspect ratio
- v_h – Rate of change of object height

Final form of State Vector: $x = [x\ y\ a\ h\ v_x\ v_y\ v_a\ v_h]'$

System Matrix

A_r = [1 0 0 0 dt 0 0 0;

0 1 0 0 0 dt 0 0;

0 0 1 0 0 0 dt 0;

0 0 0 1 0 0 0 dt;

0 0 0 0 1 0 0 0;

0 0 0 0 0 1 0 0;

0 0 0 0 0 0 1 0;

0 0 0 0 0 0 0 1];

A_{r11} = [1 0 0 0;
0 1 0 0;
0 0 1 0;
0 0 0 1];

A_{r12} = [dt 0 0 0;
0 dt 0 0;
0 0 dt 0;
0 0 0 dt];

A_{r21} = [0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0];

A_{r22} = [1 0 0 0;
0 1 0 0;
0 0 1 0;
0 0 0 1];

Input Matrix

B_r = [0 0 0 0 dt dt dt dt]’;

B_{r1} = [0 0 0 0]’;

B_{r2} = [dt dt dt dt]’;

Output Matrix

C_r = [1 0 0 0 0 0 0 0;
0 1 0 0 0 0 0 0;
0 0 1 0 0 0 0 0;
0 0 0 1 0 0 0 0];

C_{r1} = [1 0 0 0;
0 1 0 0;
0 0 1 0;
0 0 0 1];

C_{r2} = [0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0];

Feedforward Matrix

D = [0];

3.4.2.4 Partitioned and Unmeasured Portion of System

Dynamics of partitioned system are as under:

$$\begin{bmatrix} \dot{x}m \\ \dot{x}u \end{bmatrix} = \begin{bmatrix} Ar11 & Ar12 \\ Ar21 & Ar22 \end{bmatrix} \begin{bmatrix} xm \\ xu \end{bmatrix} + \begin{bmatrix} B1 \\ B2 \end{bmatrix} u$$

$$y = \begin{bmatrix} Cr1 & Cr2 \end{bmatrix} \begin{bmatrix} xm \\ xu \end{bmatrix}$$

Unmeasured Portion of System is as under:

$$\dot{x}u = Ar22xu + (Ar21xm + B2u)$$

We design Reduced Order Observer for unmeasured portion of the system.

3.4.3 Reduced Order Observer

Separated system can be visualized as under:

- Total states (n) = 8
- Measured states (m) = 4
- Unmeasured states (n-m) =4

Pole Placement

Poles need to be selected in left half plane to achieve stable output. Following poles were selected for this system. However, there is alot of flexibility in choosing pole location.

```
poles = [-2 -2.5 -3 -3.5];
```

Gain Calculation / Correction Term

Selection of Gain (L) is carried out to reduce observer error.

```
L = (place(A22,(C1*A12),poles));

L= [46.1538       0         0         0;
          0 53.8462         0         0;
          0       0   61.5385         0;
          0       0         0   69.2308];
```

Reduced Order Observer Error is defined as under:

$$\dot{\tilde{x}}u = (A22\ - LA12)\tilde{x}u$$

3.4.4 Final Form of ROO

Final form of ROO is as under:

$$\dot{z} = Dz + Fy + Gu$$

$$\widehat{xu} = z + Ly$$

$\widehat{xu}\ represents\ estimated\ unmeasured\ states.$

$$D = A22\ - LA12$$

$$F = DL + A21 - LA11$$

$$G = B2\ - LB1$$

Calculating **measured states using corresponding estimated unmeasured state** e.g. Calculating position from estimated velocity is done as under for each measured state separately:

$$xn = xn - 1 + vn.\Delta t$$

3.4.5 MATLAB Implementation of Reduced Order Observer

In order to validate performance of reduced order observer, implementation of this algorithm has been carried out separately.

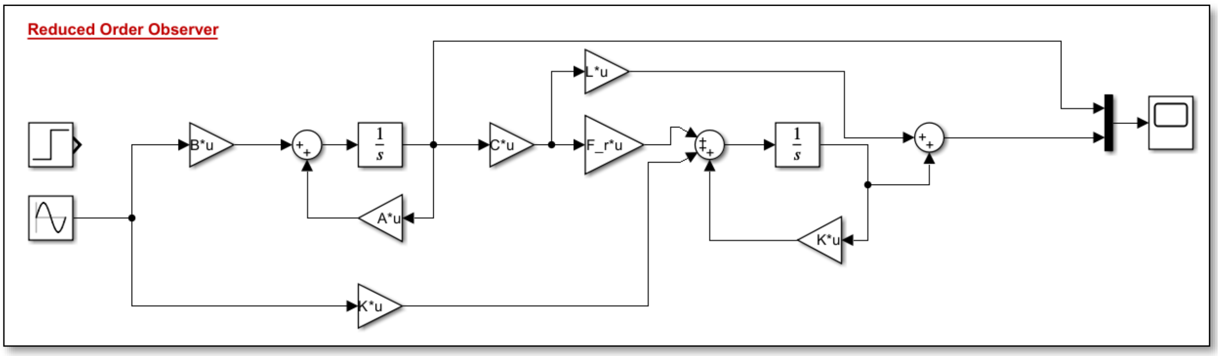

Figure 3.5 - MATLAB-Simulink Implementation of ROO

Reduced Order output on 2D System (one unmeasured state) is as under:

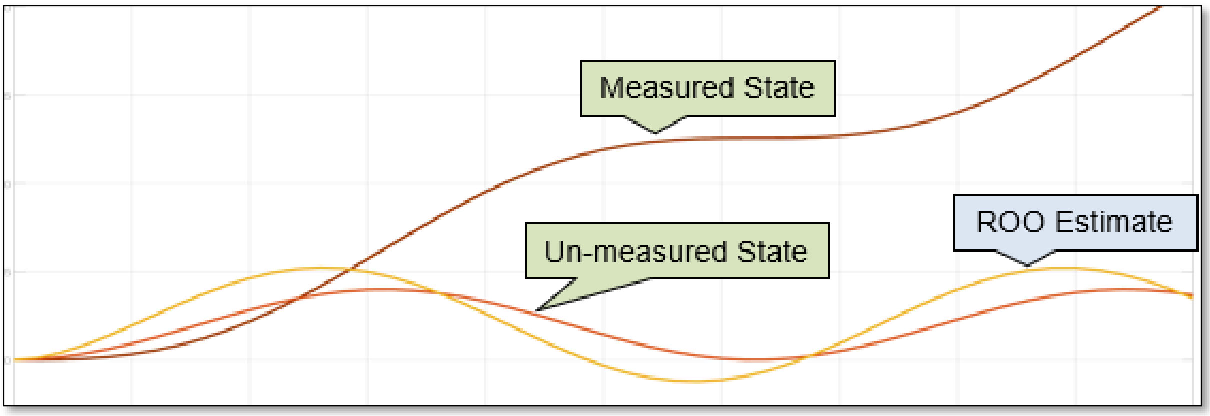

Figure 3.6 - MATLAB-Simulink Output on sample 2D System

Reduced Order output on our 8D System (4 unmeasured states), discussed above, is as under:

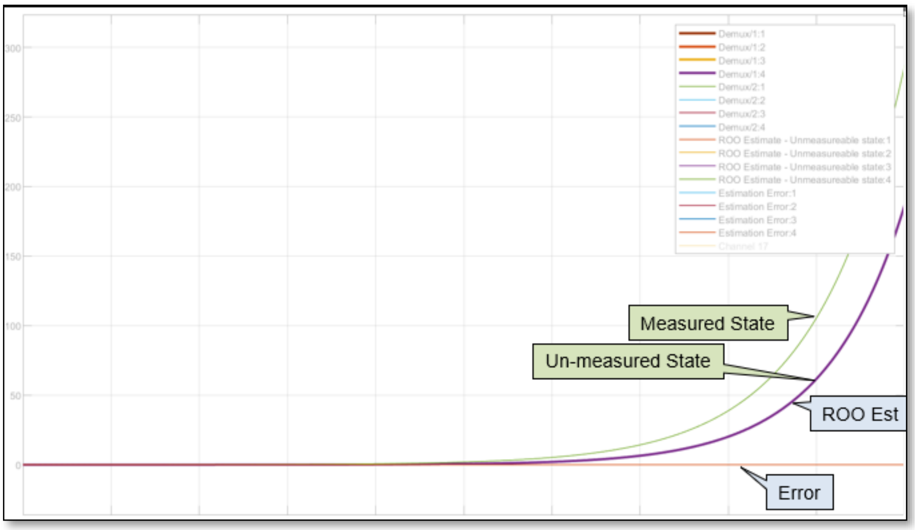

Figure 3.7 - MATLAB-Simulink Output on our 8D System

3.5 Association and Tracklet Assignment

3.5.1 Intersection Over Union (IoU) Matching

To ensure consistent tracking of objects across frames, Intersection Over Union (IoU) matching is used to assign detected objects to tracklets. IoU metric is a measure of amount of region overlapped between predicted bounding boxes and detected bounding boxes.

Algorithm for ByteTrack [11] based tracklet assignment is as under:

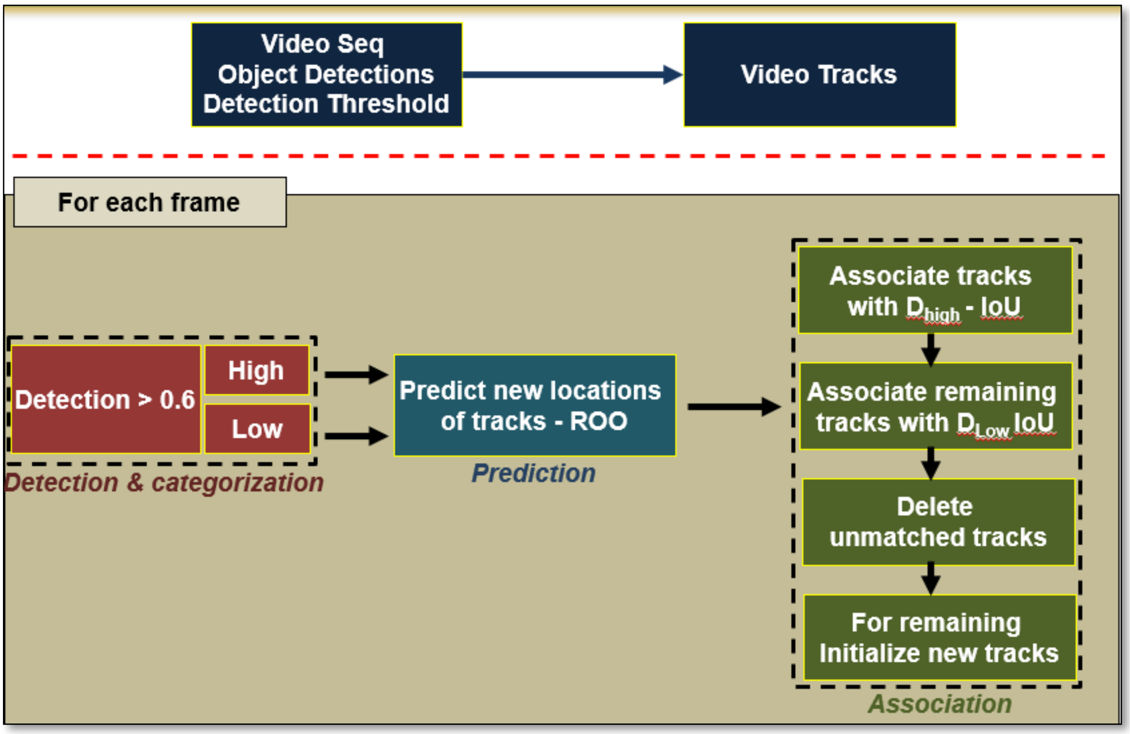
Video Seq
Object Detections
Detection Threshold
Video Tracks
For each frame
Detection > 0.6
High
Low
Detection & categorization
Predict new locations of tracks - ROO
Prediction
Associate tracks with D_{high} - IoU
Associate remaining tracks with D_{Low} IoU
Delete unmatched tracks
For remaining Initialize new tracks
Association

Figure 3.8 - ByteTrack based Tracklet Assignment

3.5.2 Tracklet Management

The tracklet management process involves:

- **Initialization**: Initialize new tracklets for objects detected in first frame.
- **Update**: Update existing tracklets based on IoU matching in subsequent frames.
- **Termination**: Terminate tracklets if an object is not detected for a specified number of consecutive frames, indicating disappearance of object.

3.6 Implementation Details

Software Tools and Frameworks

The implementation of proposed system was carried out using software tools and frameworks, described as under:

- **MATLAB R2023b**: Used for initial development and validation of the ROO-based trajectory estimation algorithm.

- **Microsoft Visual Studio 2019**: Used for implementing the complete system, including YOLOX-based object detection and ROO-based trajectory estimation and tracklet assignment.
- **Google Colab**: Used for evaluating results discussed in next chapter.

Algorithm Development

The algorithm development process involved:

- **Model Training**: Pre-trained YOLOX model was utilized for this project.
- **Observer Design**: Designing and tuning the ROO for efficient trajectory estimation.
- **Integration**: Integrating the YOLOX-based detection and ROO-based trajectory estimation into a unified system alongwith ByteTrack based association algorithm.

3.7 Summary

This chapter details used in developing proposed YOLOX-based MOT system with a ROO for target trajectory estimation. System architecture, YOLOX-based object detection, ROO-based trajectory estimation, and tracklet management were discussed. Implementation details, including software tools and frameworks, are also provided. This methodology sets foundation for analysis and evaluation of proposed approach, demonstrating its potential for real-time MOT applications.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

This chapter explains experimental setup, evaluation metrics, results, and discussion of the findings for the proposed YOLOX-based multi-object tracking system with a ROO. Goal is to demonstrate system's effectiveness and efficiency in real-time multi-object tracking applications.

4.2 Evaluation Metrics

The Clear MOT Metric paper outlines evaluation metrics for evaluating results [18].

MOTA (Multi Object Tracking Accuracy)

MOTA is a measure of overall accuracy of both the tracker and detector

$$MOTA = 1 - \frac{\sum_t (FNt + FPt + IDSWt)}{\sum_t Gt}$$

FN = False Negative

FP = False Positive

IDSW = Identity Switch

G = Ground Truth

If MOTA is 1, then system has good accuracy.

If MOTA is around zero or less, system’s accuracy is poor.

MOTA doesn’t include localization error.

MOTP (Multi Object Tracking Precision)

MOTP measures localization accuracy i.e. average dissimilarity between all TPs and corresponding GT targets. MOTP = 0 implies that there is no distance error.

$$MOTP = \frac{\sum_{t,i} dt, i}{\sum_{t} ct}$$

dt,i = Overlap between bounding box of target i with GT object (IoU).

ct = Correct Matches in frame t

IDF1

IDF1 is ratio of correct identities with average detection in each frame.

$$IDF1 = 2.\frac{IDP.IDR}{IDP + IDR}$$

IDP

IDP is reflection of precision of object identities.

$$IDP = \frac{IDTP}{IDTP + IDFP}$$

IDR

IDR is recall of object identities

$$IDP = \frac{IDTP}{IDTP + IDFN}$$

IDSW (Identity Switch)

IDSW is number of transitions of assigned identity to tracked object from one identity to another.

MT (Mostly Tracked)

Trajectories with more than 80% overlap as compared to trajectory of object in ground-truth. Number of such trajectories is called MT.

ML (Mostly Lost)

Trajectories with less than 20% overlap as compared to trajectory of object in ground-truth. Number of such trajectories is called ML.

Frag (Fragments)

Frag is summary of number of fragments in one trajectory.

4.3 Results

Results have been evaluated on MOT Challenge dataset [19]. Details are under mentioned.

4.3.1 Instance-I

Input Video Sequence

Table I - Input Video Sequence (Instance-1)

Ser	Item	Details
1	Sequence Name	MOT16-03
2	FPS	30
3	Resolution	1920x1080
4	Length	50 sec
5	Track	148
6	Boxes	104556
7	Density	69.7

Parameter Metric Comparison

Taking results from Kalman Filter based tracker as ground truth and ROO based tracker as evaluated results we achieve following comparison.

Table II - Parameter Metric Comparison (Instance-1)

Ser	Item	Details
1	MOTA	0.9854
2	MOTP	0.118
3	IDF1	0.9642
4	IDSw	5
5	ML	0
6	MT	119
7	Frag	21

Inference Time Comparison

Table III - Instance Time Comparison (Instance-1)

Ser	Method	Time
1	Inference Time using Kalman Filter	280.8 ms
2	Inference Time using ROO	203.9 ms

4.3.2 Instance-II

Input Video Sequence

Table IV - Input Video Sequence (Instace-2)

Ser	Item	Details
1	Sequence Name	MOT17-01
2	FPS	30
3	Resolution	1920x1080
4	Length	15 sec
5	Track	24
6	Boxes	6450
7	Density	14.3

Parameter Metric Comparison

Taking results from Kalman Filter based tracker as ground truth and ROO based tracker as evaluated results we achieve following comparison.

Table V - Parameter Metric Comparison (Instance-2)

Ser	Item	Details
1	MOTA	0.96
2	MOTP	0.123
3	IDF1	0.83
4	IDSw	18
5	ML	0
6	MT	33
7	Frag	20

Inference Time Comparison

Table VI - Instance Time Comparison (Instance-2)

Ser	Method	Time
1	Inference Time using Kalman Filter	86.1 ms
2	Inference Time using ROO	82.4 ms

4.3.3 Instance-III

Input Video Sequence

Table VII - Input Video Sequence (Instance-3)

Ser	Item	Details
1	Sequence Name	MOT17-05
2	FPS	14
3	Resolution	640x480
4	Length	1 min
5	Track	181
6	Boxes	6917
7	Density	8.3

Parameter Metric Comparison

Taking results from Kalman Filter based tracker as ground truth and ROO based tracker as evaluated results we achieve following comparison.

Table VIII - Parameter Metric Comparison (Instance-3)

Ser	Item	Details
1	MOTA	0.82
2	MOTP	0.2
3	IDF1	0.72
4	IDSw	173
5	ML	3
6	MT	139
7	Frag	413

Inference Time Comparison

Table IX - Inference Time Comparison (Instance-3)

Ser	Method	Time
1	Inference Time using Kalman Filter	198.3 ms
2	Inference Time using ROO	137.7 ms

4.4 Discussion

Experimental results demonstrate effectiveness of YOLOX-ROO approach for multi-object tracking. The following key observations can be made:

Comparable results of Clear Metric

The ROO-based tracking system achieved comparable results with Kalman filter based tracking system.

Efficient Trajectory Estimation

Reduced Order Observer (ROO) provided efficient trajectory estimation by focusing on unmeasured states, reducing computational load compared to traditional Kalman Filter. Hence, this approach leads to faster processing time without sacrificing accuracy.

Robust Tracking Performance

Proposed approach demonstrated robust tracking performance with high MOTA and MOTP. The number of ID switches and fragmentations was relatively low, indicating consistent and reliable tracking even in during occluded environment.

Real-Time Capability

The system processed video frames at 30 FPS in lesser time duration, achieving real-time performance suitability. Applications requiring fast and accurate MOT, such as autonomous driving and surveillance may benefit from this approach.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusions

This study demonstrates successful implementation of ROO-based tracking system for target trajectory estimation. This approach surpassed traditional Kalman filter based estimation mechanism in terms of inference time. ROO based system is proposed as viable solution for utilization in MOT pipeline where real-time performance is valued higher and minor variations in accuracy can be ignored. Significant reduction in MOT inference time per frame is a contribution in the field due to its rapid requirements and implementation of these systems on embedded systems with limited computational resources.

5.2 Future Work

A value addition to the research may be done by reducing time utilized in detection of objects and appearance learning / matching used for re-identification of objects.

REFERENCES

[1] A. Weng, “Tracking multiple objects using visual-based sensors,” in *International Conference on Artificial Intelligence and Industrial Design (AIID 2022)*, Z. Xiong and R. He, Eds., SPIE, Apr. 2023, p. 54. doi: 10.1117/12.2673209.

[2] S. Bilakeri and K. A K, “Multi-object tracking by multi-feature fusion to associate all detected boxes,” *Cogent Eng.*, vol. 9, no. 1, Dec. 2022, doi: 10.1080/23311916.2022.2151553.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2016, pp. 3464–3468. doi: 10.1109/ICIP.2016.7533003.

[4] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.07402

[5] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” Mar. 2019, doi: 10.1109/ICCV.2019.00103.

[6] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, “Towards Real-Time Multi-Object Tracking,” Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.12605

[7] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking,” Apr. 2020, doi: 10.1007/s11263-021-01513-4.

[8] Y. Raza, A. Zia, M. B. Malik, F. M. Malik, A. Qayyum, and M. I. Malik, “Image Based Multi Objects’ Trajectories Estimation through Reduced Order Observer,” in *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/MACS56771.2022.10023115.

[9] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, “Rethinking the competition between detection and ReID in Multi-Object Tracking,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.12138

[10] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, “BoT-SORT: Robust Associations Multi-Pedestrian Tracking,” Jun. 2022, [Online]. Available: http://arxiv.org/abs/2206.14651

[11] Y. Zhang *et al.*, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” Oct. 2021, [Online]. Available: http://arxiv.org/abs/2110.06864

[12] Y. Du *et al.*, “StrongSORT: Make DeepSORT Great Again,” Feb. 2022, [Online]. Available: http://arxiv.org/abs/2202.13514

[13] Y.-H. Wang, J.-W. Hsieh, P.-Y. Chen, M.-C. Chang, H. H. So, and X. Li, “SMILEtrack: SiMIlarity LEarning for Occlusion-Aware Multiple Object Tracking,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.08824

[14] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[16] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.08430

[17] G. H. H. Raymond.T Stefani, Bahram Shahian, the late Clement J. Savant, *Design of Feedback Control Systems*, 4th ed.

[18] K. Bernardin and R. Stiefelhagen, “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics,” *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008, doi: 10.1155/2008/246309.

[19] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, “MOT16: A Benchmark for Multi-Object Tracking,” Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.00831