Islamic University of Technology

Real-Time Multiple Object Tracking with Hierarchical Attention

Authored by:

Mk Bashar, 180041238

Samia Islam, 180041237

Kashifa Kawaakib Hussain, 180041227

Supervised by:

Dr. Md. Hasanul Kabir

Professor

Md. Bakhtiar Hasan
Assistant Professor

Department of Computer Science and Engineering

Islamic University of Technology(IUT)

A Subsidiary organ of the Organization of Islamic Cooperation (OIC)

Academic Year: 2021 - 2022

May 20, 2023

Declaration of Authorship

This thesis is the result of the research and experiments conducted by **Mk Bashar**, **Samia Islam**, and **Kashifa Kawaakib Hussain** under the supervision of Professor **Dr. Md. Hasanul Kabir** and Assistant Professor **Md. Bakhtiar Hasan** at the Department of Computer Science and Engineering (CSE) at the Islamic University of Technology (IUT) in Gazipur, Dhaka, Bangladesh. We confirm that this thesis has not been previously submitted for any degree or diploma and that all information derived from the published and unpublished work of others has been properly cited and listed in the references.

Authors:

Mk Bashar, 180041238

Samia Islam, 180041237

Kashifa Kawaakib Hussain, 180041227

***Supervisors**:*

Dr. Md. Hasanul Kabir
Professor,
Department of Computer Science and Engineering,
Islamic University of Technology

Md. Bakhtiar Hasan
Assistant Professor,
Department of Computer Science and Engineering,
Islamic University of Technology

Dedicated to our parents

Table of Contents

List of Figures

Page

List of Tables

Page

List of Abbreviations

CNN	Convolutional Neural Network
DL	Deep Learning
GCN	Graph Convolutional Network
JDE	Joint Detection and Embedding
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
MOT	Multiple Object Tracking
ReID	Re-Identification
RNN	Recurrent Neural Network
SOT	Single Object Tracking
SVM	Support Vector Machine

Acknowledgement

We would like to express our sincere appreciation and gratitude to Professor Dr. Md. Hasanul Kabir and Assistant Professor Md. Bakhtiar Hasan, Department of Computer Science and Engineering, IUT for their invaluable support and guidance throughout the research process. Their motivation, suggestions, and insights were crucial in helping us to successfully initialize this research project. We are deeply grateful for their valuable input and the time they dedicated to helping us, from the introduction of thesis topics to the implementation of the project. Without their support and proper guidance, this research would not have been possible. We are truly thankful for their invaluable mentorship and assistance.

Abstract

Multiple object tracking (MOT) is a crucial task in computer vision, with applications in fields such as surveillance, robotics, and autonomous systems. Accurate MOT is essential for maintaining situational awareness in complex environments and detecting objects accurately and tracking objects in real-time. In this paper, we present a novel approach for MOT that combines joint detection and embedding (JDE) which offers simultaneous detection and identification of multiple objects with a Swin Transformer for multi-scale feature extraction. The Swin Transformer, a variant of the popular Transformer architecture, is used to extract rich, multi-scale features from the input data in linear time complexity, enabling our method to handle objects of varying sizes and shapes. We added every stage of Swin blocks with prediction heads to get the multi-scale features. Also, we increased the number of Swin blocks at the first stage to accurately detect objects from large receptive fields. We evaluated our approach on a test set defined by our self-defined MIX dataset and achieved an accuracy of **84.9%**. While this is a promising result, there is more room for improvement like improving the reidentification part or modifying the mlp layers of Swin blocks.

Chapter 1

Introduction

Over the past decade, deep learning algorithms have been successfully applied to a wide range of real-world problems. In particular, deep learning has played a significant role in the field of computer vision. Object tracking is a key task in computer vision, and it follows closely after object detection. To perform object tracking, the first step is to locate the object in a frame. Each object is then given a unique ID. As the same object appears in consecutive frames, it creates a trajectory. An object can be anything, such as a pedestrian, a vehicle, a sports player, or a bird in the sky. When tracking more than one object in a frame, it is known as multiple object tracking (MOT). In MOT, we can track all objects of a single class or all objects of specified classes. In this work, we are focusing on person or pedestrian tracking. When tracking a single object, it is called single object tracking (SOT). MOT is more challenging than SOT, and as a result, researchers have proposed a variety of deep learning-based architectures to tackle MOT problems.

1.1 Motivation and Scope

The research on MOT has the potential to be applied in various application domains, such as autonomous driving, pedestrian tracking, visual surveillance, security monitoring, and player tracking etc. These areas all require the ability to accurately and efficiently track multiple objects in real time, making MOT a relevant and important topic.

Also, one of the significant reasons to be motivated to research MOT due to its trendiness in the field of computer vision. With the increasing demand for advanced video analytics, there is a need for effective solutions to track multiple objects in complex and dynamic environments. As a result, MOT has become a popular and actively researched topic in the field.

Finally, there are numerous existing methods for solving MOT-related problems.

These methods provide a rich set of approaches and techniques that can be drawn upon in research. By studying and building upon these methods, contributions can be made to the advancement of MOT technology and improving its performance and capabilities. Overall, the potential applications, trendiness, and existing methods in MOT make it a compelling and important research area.

1.2 Problem Statement

The main purpose of multi-object tracking (MOT) is to detect and identify objects and track their trajectories ensuring that the identity of the objects does not change.
Multiple object tracking can be divided into two main stages:

1. **Object Detection:** In this phase, a deep learning model is used to identify and locate each object present in each frame of the video. Each person is represented by a bounding box.

2. **Person Tracking:** In this phase, the information gathered during object detection is used to track the movement of the objects. Each object is assigned a unique ID, and this ID remains consistent as long as the object is present in the video. If the ID changes, it means the system is treating the same object as a different one.

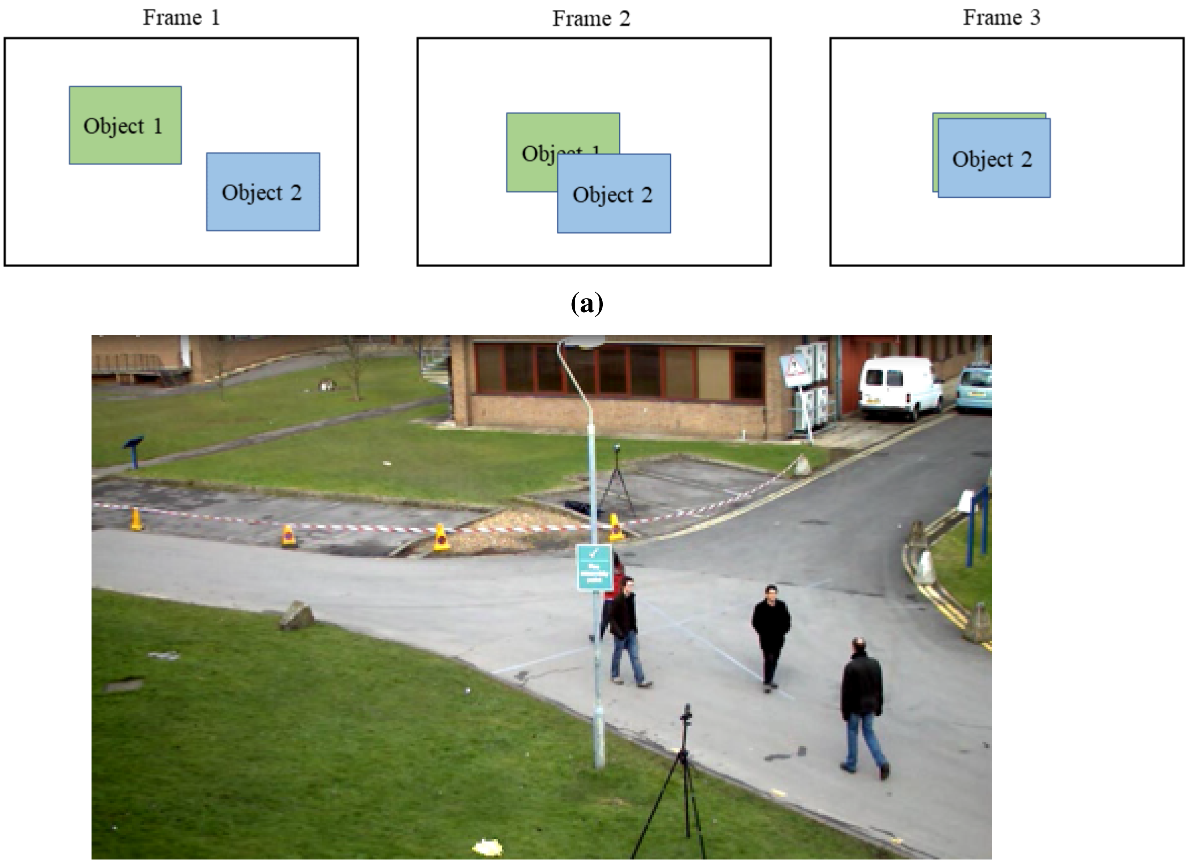
1.3 Research Challenges

One of the main challenges in the field of object tracking is occlusion handling, which refers to the problem of accurately tracking an object when it is partially or fully obscured from view. This can occur when another object or obstacle moves in front of the target object, making it difficult for the tracking algorithm to distinguish the target from its surroundings.

Another challenge is ID switching, which occurs when the tracking algorithm mistakes one object for another and switches the assigned ID from one object to another. This can lead to errors in the tracking results and may require manual intervention to correct them.

Real-time tracking is another challenge, as the tracking algorithm must be able to process and update the object's location in real-time, often with limited computational resources. This requires the use of efficient algorithms and optimization techniques to ensure that the tracking system can operate at the desired frame rate.

In addition to the challenges mentioned above, there are several other factors that can impact the accuracy and efficiency of object tracking architectures. One such fac-



(a)

(b)

Figure 1.1: (a) Illustration of the occlusion of two objects (green and blue). In frame 1, two objects are separate from each other. In frame 2, they are partially occluded. In frame 3, they are totally occluded. (b) A real-life example of occlusion [1]

tor is the presence of background clutter, which can make it difficult for the tracking algorithm to distinguish the target object from its surroundings. This can be especially problematic in environments with complex or dynamic backgrounds, such as in surveillance or traffic monitoring applications. Another challenge is the variable lighting conditions that may be encountered in real-world scenarios. Changes in lighting can affect the appearance of the target object and make it more difficult for the tracking algorithm to accurately identify and follow it. Motion blur can also be a challenge, as it can cause the appearance of the target object to become distorted and make it more difficult for the tracking algorithm to accurately detect and track it. This can be especially problematic when the camera or the target object is moving at high speeds. Finally, the size and appearance of the target object can also impact the accuracy and efficiency of object tracking algorithms. Smaller objects or those with similar appearances to other objects in the scene can be more difficult to track accurately and may require the use of specialized techniques or algorithms to overcome these challenges.

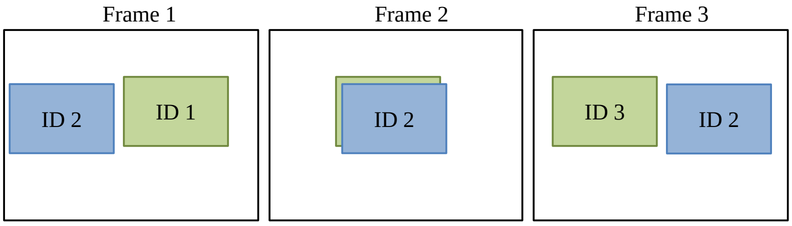


Figure 1.2: Illustration of ID switching. In frame 1, the green object had ID 1. In frame 2, the blue object with ID 2 occluded the green object. In frame 3, the ID of the blue object switched to ID 3.

1.4 Contributions

Some of the papers have done detection and association separately [3, 6–8], whereas some of them have done jointly [9–11]. The advantage of joint detection and association is lower inference time. Wang et al. have presented a multiple object tracking (MOT) system that combines target detection and appearance embedding into a single model [9]. The system is formulated as a multi-task learning problem and is able to output detections and corresponding embeddings simultaneously. The resulting system is able to run in near real-time, with a speed of 18.8 to 24.1 FPS depending on the input resolution, while achieving tracking accuracy comparable to state-of-the-art trackers. They have used feature pyramid network (FPN) [12] for feature extraction in multiple scales and in each stage of the scales, a prediction head is added [9]. However, this system has some limitations, such as comparatively low accuracy and the use of a convolutional network as the detector, which has a low inductive bias. Zhu et al. have addressed the issue of lower inductive bias by replacing the FPN with vision transformer (ViT), as transformer uses attention instead of convolution [13]. They have implemented a lightweight architecture for object tracking by using the encoder of a transformer to generate a feature map, and then employing three tracing heads to predict bounding box classification, regression, and embedding. This approach is different from many other approaches that utilize convolutional layers or popular CNN architectures to extract features from a frame, as these can add extra load to the main architecture. The ViTT architecture proposed by Zhu et al. uses relatively lightweight transformer encoders and simple feed-forward networks as tracking heads, resulting in a lightweight overall architecture. But this system also has some limitations like its inability to take into account scales and its computational cost, which is quadratic in complexity. To address all these problems, in this research paper, we present a novel approach for multiple object tracking (MOT) that combines joint detection and embedding with a Swin Transformer [14] for multi-scale feature extraction. The use of the Swin Transformer allows our method to extract rich, multi-scale features from the input data, enabling it to handle objects of varying sizes and shapes.

Through our evaluation of our approach on a test set defined by our defined data protocol, we achieved an accuracy of 26.3%. We have trained other relevant models with our small data protocol and constructed a comparative analysis. We can see that our method outperforms some of the existing models which highlights a huge research potential. In summary, our contributions are as following:

1. Proposed a novel architecture by introducing Swin transformer in Joint Detection and Embedding (JDE) to get multi-scale hierarchical features

2. Added skip connections from each stage to its consecutive prediction head after upsampling

3. Changed the backbone of original swin transformer by putting more number of blocks at the first stage instead of third to increase the span of receptive field

4. Defined a new MIX dataset by combining MOT 15, MOT 16 and MOT 17 for fair evaluation

5. Trained our method as well as other relevant architectures on the MIX dataset

6. Made an analytical presentation of the results found

1.5 Organization

In this work, we have structured our research as follows: Chapter 2 provides an overview of various frequently used multi-object tracking (MOT) approaches that address challenges in this field. Chapter 4.1 explains the MIX dataset which is defined by us. In Chapter 3, we present our proposed architecture. Chapter 4 includes an analysis of our results, including the experimental setup, implementation details, and evaluation metrics. Finally, Chapter 5.2 outlines our plans for future research in this field and identifies areas for further investigation.

Chapter 2

Background Study

Target association and item detection are often the first two phases in multiple object tracking. While some methods concentrate on data association, others concentrate on object detection. For these two processes, there are many different methodologies, and it is not always obvious if a methodology is for the detection or association phase. Numerous methods also mix and overlap various MOT elements. As a result, identifying the techniques that are independent of one another can be challenging. Nevertheless, in order to help in choosing which technique to utilize, we have made an effort to identify the most often applied approaches.

2.1 Transformer

In recent years, transformer models have been widely used in the field of computer vision and multiple object tracking (MOT) [15]. Transformer models consist of an encoder and a decoder [16], with the encoder capturing self-attention and the decoder capturing cross-attention. This attention mechanism allows for long-term context memorization. Because they can handle sequential data, transformer models are frequently employed in MOT to anticipate the placement of objects in the following frame based on information from the previous frame.

Several papers have explored the use of transformer models for MOT. Peize et al. developed TransTrack, which produces two sets of bounding boxes from object and track queries and uses simple IoU matching to determine the final set of boxes, representing the tracking boxes for each object [17]. Tim et al. proposed a similar approach called TrackFormer [2]. In another approach, patches of images were first detected and probabilistic concepts were used to obtain expected tracks, with frames being cropped according to the bounding boxes to obtain patches [18]. These patches were then used to predict tracks for the current frames.

En et al. combined an attention model with a transformer encoder to create the

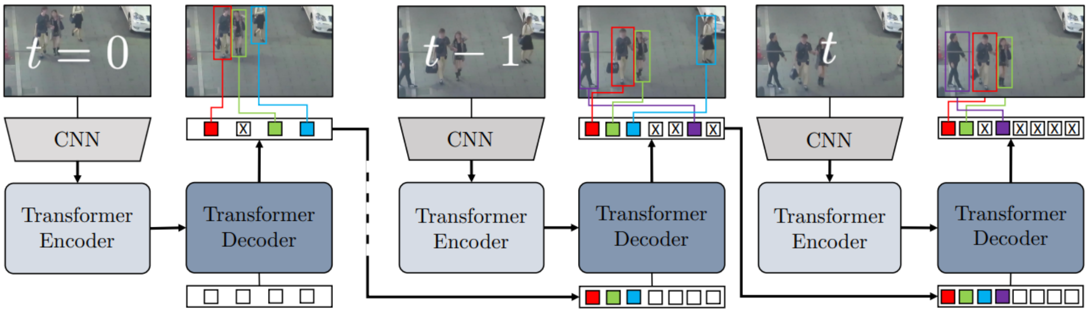


Figure 2.1: Utilizing the encoder-decoder architecture of the transformer, TrackFormer [2] converts multi-object tracking as a set prediction problem performing joint detection and tracking-by-attention.

Guided Transformer Encoder (GTE), which only processes significant pixels of each frame in a global context [19]. Yihong et al. proposed a multi-scaled pixel-by-pixel dense query system that generates dense heatmaps for targets to improve accuracy [20]. Some papers have focused on improving the computation cost for real-time transformer-based MOT, such as using an exemplar attention module to reduce input dimension or inserting a lightweight attention layer into a pyramid network [21]. Zhou et al. introduced the concept of global tracking, using a window of 32 frames and applying the transformer's cross-attention mechanism more efficiently [6]. Zeng et al. extended the DETR object detection transformer with a Query Interaction Module to filter the output of the decoder before adding a detection to the tracklet [22]. Zhu et al. used the encoder of a transformer to generate a feature map and employed three tracing heads to predict bounding box classification, regression, and embedding [13]. This ViTT architecture used relatively lightweight transformer encoders and simple feed-forward networks as tracking heads, resulting in a lightweight overall architecture.

Reference	Year	Detection/Appearance Feature Extraction	Data Association	Dataset	MOTA (%)
[17]	2020	Decoder of DETR	Decoder of Transformer	MOT17, MOT20	74.5,64.5
[2]	2021	CNN	Decoder of Transformer	MOT17	62.5
[18]	2020	CNN	Transformer	MOT16, MOT17	73.3, 73.6
[19]	2022	Faster R-CNN	Hungarian Algorithm	MOT16, MOT17, MOT20	75.8. 74.7, 70.5
[23]	2022	CNN+Encoder of Transformer	Decoder + Feed Forward Network	MOT15, MOT16, MOT17	40.3, 65.7, 65.0
[20]	2021	DETR	Deformable Dual Decoder	MOT17, MOT20	71.9, 62.3
[21]	2021	Exemplar Attention based encoder	Exemplar Attention based encoder	TrackingNet	70.55 (Precision)
[4]	2022	Transformer Pyramid Network	Multihead and pooling attention	UAV123	85.83 (Precision)
[6]	2022	CenterNet	Tracking transformer Decoder and Query Interaction	TAO, MOT17 MOT 17,DanceTrack	45.8 (HOTA), 75.3 57.2(HOTA),54.2
[22]	2021	DETR	Module + Temporal aggregation network	BDD100k	(HOTA),32.0 (nMOTA)
[13]	2021	Encoder	Bounding Box Regression Network	MOT16	65.7

Table 2.1: Summary of Transformer based Approaches

2.2 Graph Model

In contrast to linear convolutional networks, graph convolutional networks (GCNs) employ neural networks in a graph-based manner [24]. Multiple object tracking (MOT) problems are increasingly being solved using graph models, in which the connections between nodes and edges represent the identified objects from successive frames. The Hungarian method [25] is frequently used in this field to do data association.

Several papers have explored the use of graph models for MOT. Guillem et al. used a message passing network combined with a graph to globally detect and track objects by extracting deep features throughout the graph [26]. Gaoang et al. followed a similar approach but removed appearance information and used an advanced embedding strategy to design tracklets [27]. Jiahe et al. used two graphs, one for appearance and one for motion, to identify similarities among frames [28]. Peng et al. used two graph modules, one for generating proposals and one for scoring them, and trained a GCN to rank the proposals according to their scores [3]. Jiawei et al. focused on solving both the association and assignment problems, using a quadratic programming layer to learn more robust features [29]. Kha et al. addressed the multi-camera MOT problem by establishing a dynamic graph to accumulate new feature information [30].

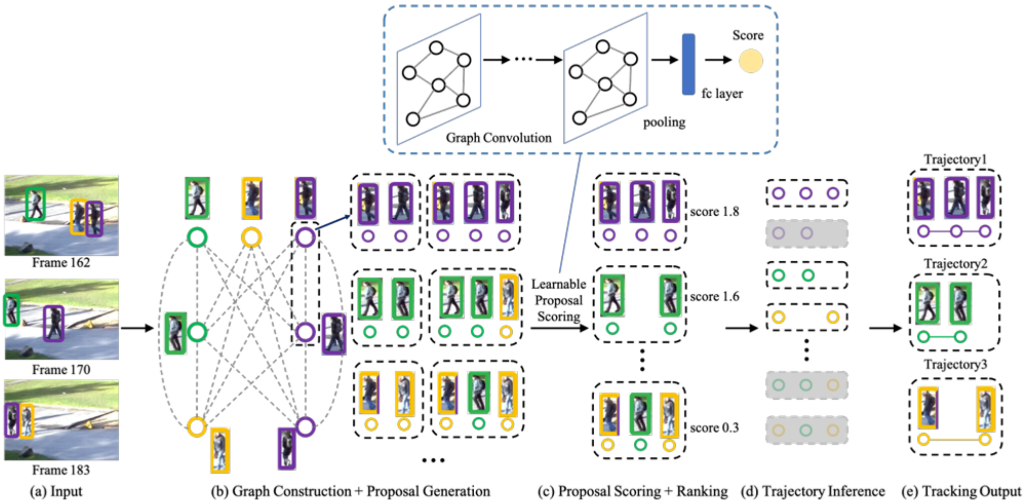


Figure 2.2: (a) Frames with detected objects. (b) Graph constructed with the detected objects or tracklets as each node and proposal generation. (c) Ranking the proposals with GCN. (d) Trajectory Inference. (e) Final output [3]

2.3 Motion Model

In the discipline of multiple object tracking (MOT), motion is a crucial component of objects that may be used for both detection and association purposes. The change in

Reference	Year	Detection	Association	Dataset	MOTA(%)
[26]	2020	ResNet50	Message Passing	MOT15, MOT16, MOT17	51.5, 58.6, 58.8
[28]	2020	ResNet-34	Hungarian algorithm	MOT16, MOT17,	47.7, 50.2
[31]	2021	SeResNet-50	Human-Interaction Model	MOT15,MOT16 DukeMTMCT	80.4, 50.0, 86.7
[27]	2021	CenterNat, CompACT	Box and Tracklet Motion Embedding	MOT17, KITTI, UA-Detrac	56.0, 87.6, 22.5
[28]	2020	ResNet-34	Hungarian algorithm	MOT16, MOT17	47.7, 50.2
[3]	2021	ResNet50-IBN	Proposal Generation and Scoring	MOT17, MOT20	59.0, 56.3
[29]	2021	CenterNet	Graph Matching	MOT16, MOT17	65.0, 66.2
[7]	2022	CenterPoint, MEGVII	Message Passing	nuScenes	55.4

Table 2.2: Summary of Graph Model based Approaches

an object's location between two frames may be used to compute motion, and this data can be used to guide a variety of tracking-related choices.

There have been several papers that have utilized motion in MOT. For example, Hasith et al. and Oluwafunmilola et al. used motion to compute dissimilarity cost in their respective works [32] and [33]. Bisheng et al. used a motion model based on Long Short-Term Memory (LSTM) to predict the location of occluded objects [34]. Wenyuan et al. incorporated a motion model with a Deep Affinity Network (DAN) [35] to optimize data association by eliminating locations where it is not possible for an object to be situated [36].

Qian et al. also calculated motion by measuring the distance between consecutive satellite frames using Accumulative Multi-Frame Differencing (AMFD) and low-rank matrix completion (LRMC) [37], and formed a motion model baseline (MMB) to detect and reduce false alarms. Hang et al. used motion features to identify foreground objects in the field of vehicle driving [38], detecting relevant objects by comparing motion features with a Generalized Linear Model (GLV). Gaoang et al. proposed a local-global motion (LGM) tracker that finds consistencies in motion and associates tracklets accordingly [27]. In addition, Ramana et al. used a motion model to predict the motion of an object rather than for data association, with a system comprising three modules: Integrated Motion Localization (IML), Dynamic Reconnection Context (DRC), and 3D Integral Image (3DII) [39].

In 2022, Shoudong et al. used a motion model for both motion prediction and association with their proposed Motion-Aware Tracker (MAT) [40]. Zhibo et al. introduced a compensation tracker (CT) with a motion compensation module to recover lost objects [41]. Xiaotong et al. used a motion model to predict the bounding boxes of objects [18] and create image patches, similar to the approach taken by Hang et al. [38].

Overall, motion has proven to be a useful feature in MOT, and it has been utilized in various ways in different works to improve detection and association performance.

Reference	Year	Motion Mechanism	Dataset	MOTA(%)
[32]	2019	Dissimilarity Distance between Detected and Predicted Object	MOT17, KITTI	46.9, 85.04
[33]	2021	Dissimilarity Distance between Detected and Predicted object	MOT15, MOT16, MOT17, MOT20	55.8, 73.8, 74.0, 60.2
[34]	2021	LSTM-based Model on Consecutive Frames	MOT16, MOT17	76.3, 76.4
[36]	2021	Kalman Filtering	MOT17	44.3
[37]	2021	Accumulative Multi-Frame Differencing and Low-Rank Matrix Completion	VISO	73.6
[38]	2021	Distance of Motion Feature and Mean Vector of Gaussian Local Velocity Model	NJDOT	100 (Anomaly Detection Accuracy)
[27]	2021	Box and Tracklet Motion Embedding	MOT17, KITTI, UA-Detrac	56.0, 87.6, 22.5
[39]	2021	Particle Filtering and Enhanced Correlation Coefficient Maximization	CroHD	63.6
[40]	2022	Combination of Camera Motion and Pedestrian Motion (IML),Dynamic Motion-based Reconnection(DRC)	MOT16, MOT17	70.5, 69.5
[41]	2022	Motion Compensation with Basic Tracker	MOT16, MOT17, MOT20	69.8, 68.8. 66.0
[18]	2022	Kalman Filtering	MOT16, MOT17	73.3, 73.6

Table 2.3: Summary of Motion model based Approaches

2.4 Siamese Network

Because Siamese networks can recognize similarities between inputs and distinguish between them, they have gained popularity in multiple object tracking (MOT) systems in recent years. Two parallel sub-networks with the common weight and parameter spaces make up this kind of network, which is then connected and trained on a loss function to gauge how semantically similar the two sub-networks are.

One common application of Siamese networks in MOT systems is the use of a region proposal network (RPN) structure as a predictor, as proposed by Xinwen et al. [42]. Another approach is the incorporation of a transformer layer into a Siamese tracking network, as seen in the work of Philippe et al. [21].

Daitao et al. proposed a pyramid network that includes a lightweight transformer attention layer. Their Siamese Transformer Pyramid Network augmented the target features with lateral cross attention between pyramid features, resulting in robust target-specific appearance representation [4]. Bing et al. aimed to improve the region-based multi-object tracking network by adding motion modeling [43]. They integrated the Siamese network tracking framework into Faster-RCNN to achieve efficient tracking through lightweight tracking and shared network parameters.

Other researchers have used Siamese networks to enhance the localization of foreground objects, as in the work of JiaXu et al. [5], or to improve the overall stability of the system, as seen in the work of Xinwen et al. [42]. In addition, Siamese networks have been used to post-process trajectories and eliminate corrupted tracklets, as in the Cleaving Network proposed by Cong et al. [31].

Overall, the use of Siamese networks in MOT systems has shown promising results

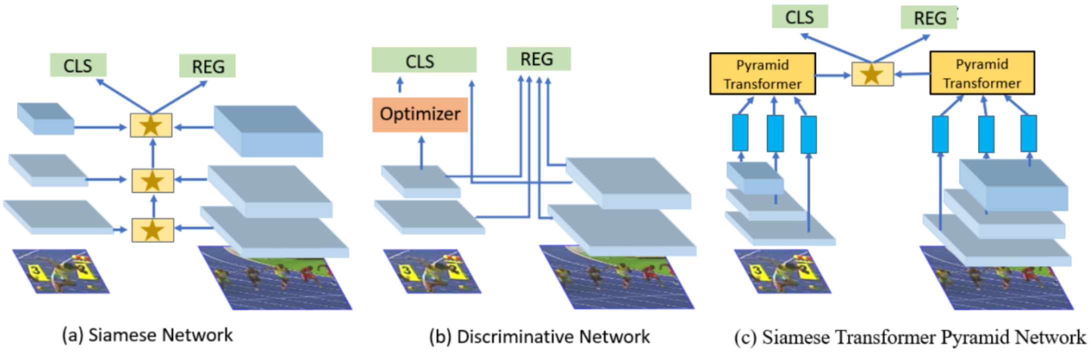


Figure 2.3: (a) A typical Siamese Network that has symmetric pyramid architecture, (b) A typical Discriminative network, (c) Siamese Transfer Pyramid Network that is proposed in [4]

in terms of improved accuracy and efficiency.

Reference	Year	Method	Dataset	MOTA (%)
[4]	2020	CNN for Apprearance extraction, LSTM and RNN for Motion modelling	Duke-MTMCT, MOT16	73.5, 55.0
[43]	2021	Implicit and Explicit motion modelling	MOT17, TAO-person, HiEve	65.9, 44.3 (TAP@0.5), 53.2
[42]	2021	Siamese Network with Region Proposal Network	MOT16, MOT17, MOT20	65.8, 67.2, 62.3
[21]	2021	Single instance level attention	TrackingNet	70.55 (Precision)
[5]	2022	Dynamic search region refine and attention based tracking	MOT17, MOT20	67.2, 70.4
[4]	2022	Transformer based appearance similarity	UAV123	85.83 (Precision)

Table 2.4: Summary of Siamese Network based Approaches

2.5 Detection and Association

In the field of multiple object tracking (MOT), various approaches have been proposed to address the challenge of associating targets, or keeping track of the trajectory of the objects of interest [44]. Some approaches, such as the one proposed by Margret et al., use both bottom-up and top-down methods to determine the trajectory of objects [45]. Bounding boxes are determined in top-down procedures whereas point trajectories are determined in bottom-up approaches. A complete track of things may be produced by combining these two techniques.

Other approaches, such as the one proposed by Hasith et al., focus on simply detecting objects and using the Hungarian algorithm to associate information [32]. In 2019, Paul et al. proposed Track-RCNN, a 3D convolutional network that can perform detection, tracking, and segmentation [11]. In 2020, Yifu et al. proposed an approach called FairMOT, which uses two separate branches for object detection and re-identification, both using center-based feature extraction [8].

In 2021, several approaches have been proposed that use long short-term memory (LSTM) for data association. Bisheng et al. proposed Detection Refinement for

Tracking (DRT), which uses semi-supervised learning to produce heatmaps for more accurate object localization and LSTM [46] for data association [34]. Chanho et al. also used bilinear LSTM for data association [47]. Qiang et al. proposed CorrTracker, a correlational network that propagates information across associations, using self-supervised learning for object detection [48]. Jiangmiao et al. proposed Quasi Dense Tracking (QDTrack), which combines object detection using Faster-RCNN with residual networks and similarity learning [49]. Yaoye et al. introduced the D2LA network, which is based on FairMOT [8] and uses a strip attention module to handle occlusion [50]. Norman et al. proposed a method that estimates the geometry of each detected object and maps it to its corresponding pose in order to identify the object after occlusion [51].

In 2022, various approaches to MOT have been proposed for diverse applications, such as indoor multiple object tracking and tracking crop seedlings. Cheng-Jen et al. proposed the depth-enhanced tracker (DET) to improve the tracking-by-detection strategy for indoor environments, along with an indoor MOT dataset [52]. Chenjiao et al. proposed a method for tracking crop seedlings using YOLOv4 as an object detector and optical flow to track the bounding boxes [53]. Oluwafunmilola et al. proposed a method for object tracking in soccer videos using an LSTM-based data association method [54].

Reference	Year	Detection	Association	Dataset	MOTA (%)
[45]	2018	Faster R-CNN	Correlation Co-Clustering	MOT15, MOT16, MOT17	35.6, 47.1, 51.2
[32]	2019	DPM, F-RCNN, SDP, RRC	Hungarian Algorithm	MOT17, KITTI	46.9, 85.04
[11]	2019	Mask R-CNN	Distance Measurement	KITTI, MOTS, MOTS Challenge	65.1, KITTI MOTS, (MOTSA)
[50]	2021	CenterNet	Hungarian Algorithm	MOT15, MOT16, MOT17 MOT20	60.6, 74.9, 73.7, 61.8
[34]	2021	ResNet50	LSTM-based Motion Model	MOT16, MOT17	76.3, 76.4
[47]	2021	CenterNet	Bilinear LSTM	MOT16, MOT17	48.3, 51.5
[48]	2021	CenterNet	Correlation Learning	MOT15, MOT16, MOT17, MOT20	62.3, 76.6, 76.5, 65.2
[49]	2021	Faster R-CNN	Quasi-dense Similarity Matching	MOT16, MOT17, BDD100K, Waymo	69.8, 68.7, 64.3, 51.18
[39]	2021	HeadHunter	HeadHunter-T	CroHD	63.6
[55]	2021	CenterNet	CVA (Cost Volume based Association	MOT16, MOT17, nuScenes, MOTS	70.1, 69.1, 5.9 (AMOTA), 65.5 (MOTSA)
[52]	2022	Mask-RCNN	Hungarian Algorithm	MOT17, MOT20 NTU-MOTD	43.21, 57.70, 92.12
[53]	2022	YOLOv4	Hungarian Algorithm	TAMU2015V, UGA2015V, UGA2018	79.0%, 65.5%, 73.4%
[54]	2022	DLA-34	Hungarian Algorithm	MOT15, MOT16, MOT17, MOT20	55.8, 73.8, 74.0, 60.2
[56]	2022	DPM and YOLOv5 with detection modifier(DM)	Global and Partial Feature Matching	MOT16	46.5
[57]	2022	YOLO X with later NMS	Kalman Filtering, Bicubic Interpolation and ReID Model	MOT17, MOT20	78.3, 75.7
[58]	2022	T-ReDet module	ReID-NMS Model	MOT16, MOT17, MOT20	63.9, 62.5, 57.4

Table 2.5: Summary of Detection and ASsociation based Approaches

2.6 Attention Module

In multiple object tracking (MOT), attention mechanisms are often used to re-identify occluded objects. Attention involves considering only the objects of interest and nullifying the background, in order to better remember the features of the objects even after occlusion.

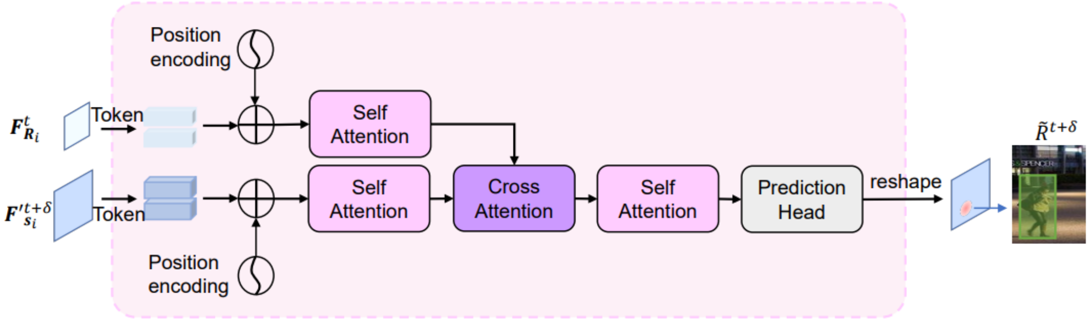


Figure 2.4: The structure of Attention based head of cross-attention [5]

Several approaches to MOT have incorporated attention modules in order to handle occlusion. Yaoye et al. proposed a strip attention module to re-identify occluded pedestrians in their D2LA network [50]. This module is a pooling layer that uses max and mean pooling to extract more useful features from the pedestrians, so that the model can remember them even when they are occluded. Song et al. used two attention modules, one for target and one for distraction, to link object localization and data association and applied a memory aggregation to create strong attention [59].

Tianyi et al. proposed a spatial-attention mechanism using a Spatial Transformation Network (STN) in an appearance model to force the model to focus only on the foreground [60]. Lei et al. proposed the Prototypical Cross-Attention Module (PCAM) to extract relevant features from past frames and the Prototypical Cross-Attention Network (PCAN) to transmit the contrasting feature of foreground and background throughout the frames [61].

Huiyuan et al. proposed a self-attention mechanism for vehicle detection [62], and JiaXu et al. used both cross and self-attention in a lightweight architecture for MOT [5]. The self-attention module is used to extract robust features and reduce background occlusion, while the cross-attention module is used for instance association.

Reference	Year	Attention Mechanism	Dataset	MOTA(%)
[50]	2021	Strip Pooling	MOT15, MOT16, MOT17, MOT20	60.6, 74.9, 73.7, 61.8
[59]	2021	Temporal Aware Target Attention and Distractor Attention	MOT16, MOT17, MOT20	59.1, 59.7, 56.6
[60]	2021	Spatial Transformation Network (STN)	MOT16, MOT17	50.5, 50.0
[61]	2021	Spatio-Temporal Cross-Attention	BDD100K (Validation), KITTI-MOTS(Validation)	27.4 (MOTSA), 66.4 (mMOTSA)
[62]	2021	Self-Attention in Detection	Custom Dataset: Sparse Scene, Dense Scene	70.9, 56.4
[30]	2021	Graph Structural and Temporal Self-Attention	PETS09, EPFL, CAMPUS, MCT CityFlow(Validation)	93.5, 66.3, 96.7, 95.7, 90.9
[5]	2022	Self- and Cross-Attention as Tracking Head	MOT17, MOT20	75.6, 70.4

Table 2.6: Summary of Attention based Approaches

2.7 Tracklet Association

Tracklet association is the process of identifying and connecting consecutive frames of objects of interest, or tracklets, in order to establish a trajectory. This is a challenging task in multiple object tracking (MOT). Different approaches have been proposed to address this issue.

Jinlong et al. proposed the Tracklet-Plane Matching (TPM) method, which creates short tracklets from detected objects and aligns them in a tracklet plane, assigning each tracklet with a hyperplane based on their start and end time [63]. This process can handle non-neighboring and overlapping tracklets, and the authors also proposed two schemes to improve performance.

Duy et al. used a 3D geometric algorithm to create tracklets and optimized the association globally by incorporating spatial and temporal information from multiple cameras [64]. Cong et al. proposed the Position Projection Network (PPN) to transfer the trajectories from local to global context [31]. Daniel et al. used a tracking-by-regression approach, re-identifying occluded objects based on motion and using already-found tracks for regression, and also extended this approach by incorporating temporal direction to improve performance [65].

The multi-view trajectory contrastive learning (MTCL) technique, which treats each trajectory as a center vector and builds a trajectory-center memory bank (TMB) that is dynamically updated and computes cost [66], was proposed by En et al. They also created the similarity-guided feature fusion (SGFF) strategy to eliminate ambiguous features and the learnable view sampling (LVS) approach, which regards each detection as a key point and aids in seeing the trajectory in a global context. The tracklet booster (TBooster) approach was created by Wang et al. to reduce association mistakes while using the [67] command. TBooster consists of two modules: a Connector module that binds tracklets belonging to the same object and executes tracklet embedding, and a Splitter module that divides tracklets when ID switching takes place.

Reference	Year	Method	Dataset	MOTA(%)
[63]	2020	Tracklet-plane matching process to resolve confusing short tracklets	MOT16, MOT17	50.9, 52.4
[68]	2021	CenterTrack [69] and DG-Net [70] as tracking graph and GAEC+KLj [71] heuristic solver for lifted multicut solver	WILDTRACK, PETS-09, Campus	97.1, 74.2, 77.5
[31]	2020	CNN for Apprearance extraction, LSTM and RNN for Motion modelling	Duke-MTMCT, MOT16	73.5, 55.0
[65]	2021	Regression based two stage tracking	MOT16, MOT17, MOT20	66.8, 65.1, 61.2
[67]	2021	Tracklet splitter splits potential false IDs and connector connects pure tracks to trajectory	MOT17, MOT20	61.5, 54.6
[66]	2022	Learnable view sampling for similarity-guided feature fusion and Trajectory-center memory bank for re-identification	MOT15, MOT16, MOT17, MOT20	62.1, 74.3, 73.5, 63.2

Table 2.7: Summary of Tracklet Association based Approaches

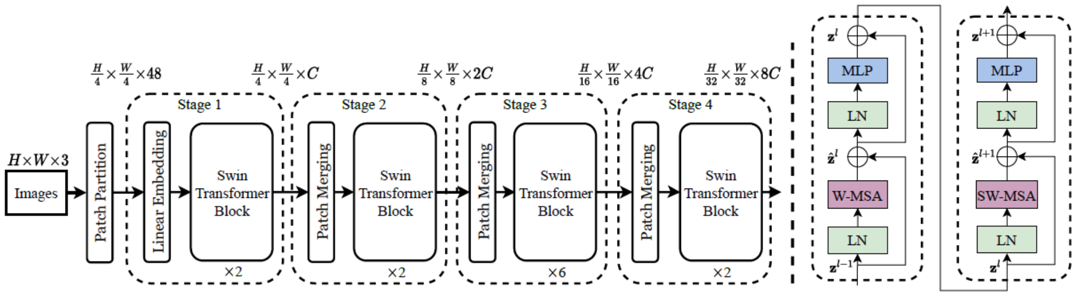
Chapter 3

Proposed Methodology

The objective of our proposed method is to develop a real-time online tracker. To this end, we proposed a Swin Transformer-based Joint Detection and Embedding Architecture. In our pipeline, swin transformer is utilized for hierarchical attention-based feature map generation. This multi-scale feature map is further used to correctly track the objects in a video frame in our network's prediction head. Though our pipeline is highly motivated by the Joint Detection and Embedding model [9] and Swin Transformer [14], it is not a naive combination of JDE and Swin. Instead, we have tried to present the Swin Transformer as a better alternative to the FPN of JDE.

3.1 Swin Transformer

Swin transformer is a perfect modification of Feature Pyramid Network to JDE. Because in JDE, Feature Pyramid Network is used to extract hierarchical feature map which is necessary to predict the tracklets. Here hierarchical feature map is necessary to tackle the different sizes of objects. We have proposed Swin Transformer in the place of FPN. Because swin transformer has the same hierarchical structure as FPN. From each stage of swin transformer, by using an mlp head, we can get a feature map which is passed to a yolo layer to predict bounding box, classification result and unique id. Here, swin transformer enables the attention mechanism, which ensures more robust feature map than FPN. By integrating Swin Transformer in the place of FPN, intuitively, our model should get better generalization and inductive biases. The modification we have introduced in the original swin transformer is we have add mlp block after each stage of the swin transformer as we want to get the feature map from each stage. In the swin transformer, the first, second and fourth blocks are stacked twice, but the third block is stacked six times. But in our cases, we stacked the first block is stacked for ten times, and the rest of the blocks are stacked twice. This is because we wanted to increase the receptive field size of the features at the firat time when we are passing the getting features of the first block. So, for the first block, the receptive field size is 70x70, and it increases for 14x14 for the consecutive blocks. But for the original swin, the receptive field size of the feature from the first block was 14x14. In this way, the proposed swin transformer is effective in feature extraction with hierarchical attention.



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Figure 3.1: (a) The architecture of a Swin Transformer; (b) two successive Swin Transformer Blocks

3.2 Joint Detection and Embedding

As of now, we mostly use the joint detection and embedding prediction head as the original paper implemented [9]. However, we have increased some convolution layers before passing the feature map to the prediction head. Because convolution layers works well for object detection. As object detection is a very crucial in our task, we added more convolution layer to get higher accuracies. In the original JDE there are total 3x2 = 6 layers are used in each hierarchical output, where we have used 5x2 = 10 layers. In the prediction head, we get three predictions and three losses. We calculated triplet loss from these three losses, as like in the JDE paper. Finally, our total architecture is presented in figure 3.2

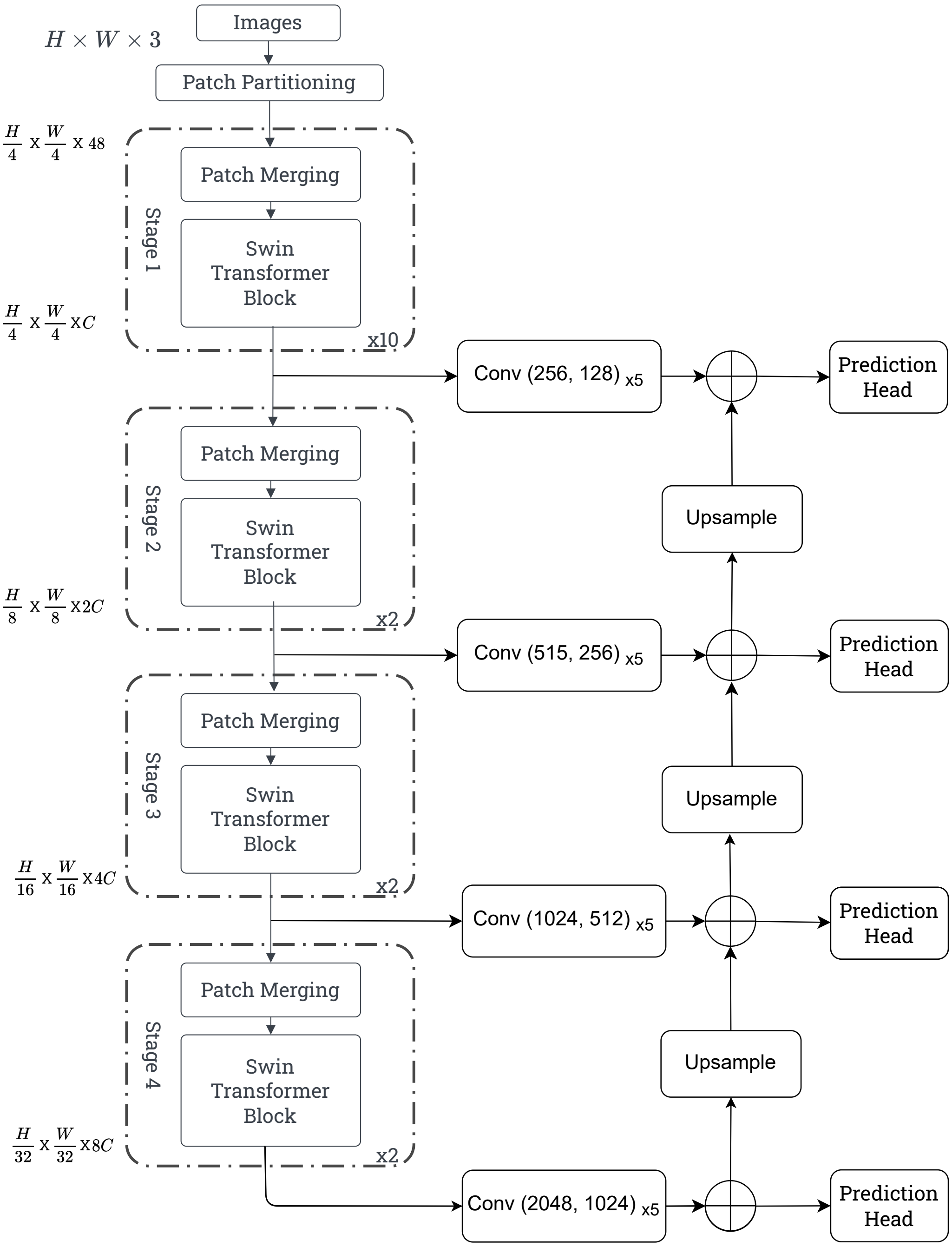


Figure 3.2: Proposed Architecture (Swin-JDE)

Chapter 4

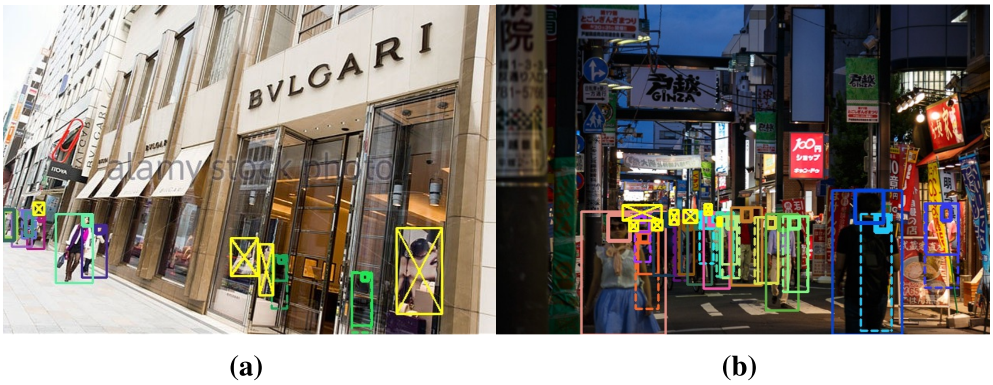
Results and Discussion

4.1 Dataset Description

There are several datasets in MOT that are regarded as benchmark datasets. Among these datasets, we used MOT15, MOT16 and MOT17 for training and testing our proposed methods and architectures. Also, we have used CrowdHuman dataset to train the backbone.

4.1.1 CrowdHuman

We have train our core backbone network, that is modified swin transformer on CrowdHuman dataset to produce a pretrained weight. This dataset contains 15000 images for training, and 5000 images for testing. The datset is only for human detection. We have used this dataset because our task is human specific, also the size of the dataset is feasible for us.



(a) **(b)**

Figure 4.1: Sample images of CrowdHuman dataset with ground truth: (a) Day scene, (b) Night scene

4.1.2 MIX Dataset

We have defined MIX dataset with the combination of MOT15, MOT16 and MOT17 datasets. We chose 11 unique sequences from these datasets and took 7 of them as training and 4 of them as test dataset. During splitting, we made sure that there is even distribution of the scenes. Table 4.1 and Table 4.2 represents detailed explanations.

Sequence Name	**Dataset**	**Length**	**FPS**
KITTI-13	MOT15	340	10
MOT17-09-SDP	MOT15, MOT16, MOT17	525	30
MOT17-05-SDP	MOT15, MOT16, MOT17	837	14
TUD-Campus	MOT15	359	25
TUD-Stadmitte	MOT15	179 25	
MOT17-11-SDP	MOT16, MOT17	900	30
MOT17-04-SDP	MOT16, MOT17	1050	30

Table 4.1: Training set of MIX dataset

Sequence Name	**Dataset**	**Length**	**FPS**
KITTI-17	MOT15	145	10
MOT17-10-SDP	MOT15, MOT16, MOT17	654	30
ETH-Sunnyday	MOT15	354	14
PETS09-S2L1	MOT15	795	7

Table 4.2: Test set of MIX dataset

4.2 Experimental Setup

We trained our model and performed all experiments on 2 RTX 3090 GPUs. For a fair comparison with other MOT trackers, we train and evaluate all experiments in a conda environment with the same hardware.

4.3 Implementation Details

We employ the swin transformer as the backbone network. The network is trained with Adam optimizer for 50 epochs in the CrowdHuman dataset. The learning rate is initialized as 3e-4 and decreased by 0.1 in the 41st and 47th epochs. Several data augmentation techniques, such as random rotation, random scale, and color jittering, are applied to reduce overfitting. Finally, the augmented images are adjusted to a fixed resolution.

After training the backbone of Swin-JDE in the CrowdHuman dataset, we loaded the weight of swin in the backbone and fine-tuned our whole architecture on our MIX dataset. We used both SGD and Adam optimizer and cosine annealing and reduce on plateau scheduler on different experiments. In our final experiment, we fine-tuned our model for 50 epochs with a learning rate of 3e-4.

4.4 Evaluation Metric

We select three metrics among the several MOT metrics to test our model. They are: Multiple Object Tracking Accuracy(MOTA), ID Switch(IDs), and Multiple Object Tracking Precision(MOTP).

Multiple Object Tracking Accuracy (MOTA) measures how accurately a model can detect objects and predict trajectories. It is the prime metric to evaluate an object-tracker's performance.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

m_t: The number of misses at time t
fp_t: The number of false positives
mme_t: The number of identity switches
g_t: The number of objects present at time t

Multiple Object Tracking Precision(MOTP) measures how accurately the model was able to find the objects location in the video. It is often used alongside MOTA as it can account for localization accuracy.

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$$

d_t^i: The distance between the actual object and its respective hypothesis at time t, within a single frame for each object o_i from the set a tracker assigns a hypothesis h_i.
c_t: Number of matches between object and hypothesis made at time t.

ID Switch(IDs) gives us an idea about how much our model is good at reidentification.

$$IDs = \frac{number\ of\ ID\ Switches}{recall}$$

4.5 Quantitative Results

In the joint detection and embedding technique, there is a trade-off between accuracy and inference time. We have observed a similar phenomenon in our experiment also. When we set input image size to 868×480, then we found the highest FPS among all models we compared. But the best MOTA is produced by input image size of 1080×608.

4.5.1 Ablation 1: Prediction Head Count

The goal of our first experiment was to measure the importance of prediction heads. Each prediction head gets a fixed-size feature map and 12 anchor boxes to predict bounding boxes based on that feature map. Intuitively, if we have more prediction heads, then we will get more variable-size feature maps, and each head will predict based on a small subset of anchor box choice, instead of all 12 together. For one prediction head, as the 12 anchor box choices are applied to only one fixed-size feature map, the MOTA is not that much high. But it allows the model to be relatively smaller and results in the highest FPS. If we increase the prediction head to two, the MOTA improved a lot, and as expected FPS drops. Lastly, we found a good MOTA with decent FPS with 4 prediction heads.

Prediction Head	**MOTA↑**	**IDs↓**	**MOTP↑**	**FPS↑**
1	45.80%	956	0.167	**45.6**
2	78.80%	664	**0.22**	35.67
4	**83.60%**	532	0.163	30.5

Table 4.3: Prediction Head Count

4.5.2 Ablation 2: Swin Block Order (Without Pretraining)

We ran the second set of experiments keeping in mind the logic of the receptive field. In each stage of the swin layer, the swin block repeats several times. This repeating siwn block increases the receptive field of a feature map. When we integrate 4 feature maps, we take the feature map from the first stage of swin. In that case, if the swin block only repeats 2 times, it will produce a receptive field of size $7 \times 2 = 14$. Instead, if we choose to repeat the first stage ten times, then it will produce a receptive field of size $7 \times 10 = 70$. This is a very good feature map compared to the former one. We ran these experiments without taking pre-trained weight as the architecture changed, we can not load weights from official ImageNet weights. Also, our new architecture needs to be trained on the CrowdHuman dataset for better convergence in the final set of experiments. The results is presented on the table 4.4.

Swin Block Order	**MOTA↑**	**IDs↓**	**MOTP↑**	**FPS↑**
2, 2, 18, 2	75.60%	524	0.183	**36.74**
18, 2, 2, 2	**82.70%**	**504**	0.167	30.64
10, 2, 2, 2	81.650%	579	**0.21**	32.40

Table 4.4: Swin Block Order (Without Pretraining)

4.5.3 Ablation 3: Convolution Layers Count

The convolution layers, which are placed after the swin blocks in our proposed architecture carry a significant value as it determines the object detection accuracy. The convolution blocks are responsible for producing high-level features. So as we repeated them larger times, more high-quality high-level features are extracted, which results in better detection accuracy. As the direct impact is reflected in MOTA, that is presented in table 4.5

Conv Layers	**MOTA↑**	**IDs↓**	**MOTP↑**	**FPS↑**
3x	76.50%	536	0.145	**34.77**
4x	77.20%	576	**0.23**	32.49
5x	**81.80%**	588	0.145	30.49

Table 4.5: Convolution Layers Count

4.5.4 Comparison of the Proposed Method with State of the Art

After some successful ablation studies, we get our proposed architecture in a very good shape, which consists of 4 prediction heads, with 10-2-2-2 swin block order and 5x conv layers. Our model with a larger image size clearly beat the SOTA in MOTA metric and our model with a smaller image size could achieve the highest FPS. We can see some observations such as, with larger image size ID switch also increases. Because in larger images, the detection of objects stiffly increases, which ultimately increases the ID switches. So, MOTA and ID-switch kind of played a trade-off role in this case.

We can clearly see that ViTT performs very badly because the feature extraction part of ViTT consists of single-scale transformer block. Instead of using Vision transformer, we use Swin transformer and overcome this limitation. Tracformer has a very low FPS, which forfeits the competition in terms of real-time tracker. Though JDE has a very good FPS, our Swin-JDE with smaller image size beats it in this field too.

Lastly, FairMOT is one of the good competitors of our proposed architecture. But our Swin-JDE with bigger image size clearly has higher MOTA and MOTP. Overall, our proposed architecture made some improvement in all metrics by a good margin.

Model	**MOTA↑**	**IDs↓**	**MOTP↑**	**FPS↑**
ViTT [13]	56.33%	2563	0.14	28.45
Trackformer [2]	63.80%	414	0.167	7.4
JDE [9]	66.70%	1021	0.21	36.74
FairMOT [8]	82.60%	504	0.183	32.64
Swin-JDE (Ours - 868 x 480)	79.8%	**328**	**0.22**	**38.56**
Swin-JDE (Ours - 1088 x 608)	**84.9%**	664	**0.22**	27.34

Table 4.6: Comparative Analysis of State of The Art Models on MIX

Chapter 5

Conclusion

5.1 Summary

In this work, we tried to combine the JDE with hierarchical attention to solve multiple object tracking related tasks. Specifically, we replace Swin Transformer with FPN in JDE, though it is not a naive combination. Rather we provide reasoning like, the Swin blocks of our proposed architecture extract low-level features which help to detect objects in a video frame and the convolution layers (after the Swin blocks) extract high-level features which help in the identification of objects in consecutive frames. This modification resulted in a significant improvement in the MOTA metric, which is widely used to evaluate the performance of multiple object-tracking algorithms, Particularly, our modified approach achieved a MOTA of 84.9%, indicating that it was able to successfully track a higher number of objects compared to the original JDE approach. In addition to this, we also developed our own data protocol for evaluating the performance of multiple object-tracking algorithms using the MOT15, MOT16, and MOT17 datasets. These datasets are commonly used to benchmark the performance of different approaches, and our new protocol allows us for more thorough and accurate evaluations. Overall, the results of our work demonstrate the potential of our modified JDE approach for improving multiple object tracking in a variety of contexts. The use of the Swin Transformer for hierarchical multi-scaled attention has proven to be a valuable addition to the JDE approach, and there is still room for further improvement through additional modifications and optimization. Our goal is to make further advances in the field of multiple object tracking and contribute to the development of more effective and efficient algorithms for this important task.

5.2 Future Works

The findings and contributions of this study have opened up several directions for future research. By building upon the work presented so far, it is expected that these future studies will lead to further advances and improvements in the field. In this section, we will outline some of the areas where further exploration and development could be particularly valuable.

1. **Contribution to data association technique**: One possible direction for future research is to further explore and contribute to the development of data association techniques. This could involve studying the effectiveness of different algorithms and approaches for data association in various contexts and environments, and identifying potential improvements or modifications that could be made to existing techniques.

2. **Modification of MLP head of Swin blocks**: Another potential area of focus could be on modifying the MLP head of Swin blocks to improve performance. This could involve exploring different architectures or optimization techniques, or studying the impact of different hyperparameter settings on the performance of the MLP head.

3. **Redefine dataset protocol**: A third direction for future work could be to redefine the dataset protocol used in the current study. This could involve expanding the size and diversity of the dataset, or developing new protocols for collecting and annotating data that more closely reflect real-world scenarios.

4. **More hyperparameter fine tuning**: Finally, further hyperparameter fine-tuning could be performed to further improve the performance of the proposed approach. This could involve studying the impact of different hyperparameter settings on the performance of the model, and identifying optimal settings that achieve the best results.

5. **More modifications in the backbone architecture**: More modifications to the backbone architecture of the Swin Transformer can be possible to enhance the performance. This may include adding or removing layers, changing the number of neurons in each layer, or altering the type of activation function used. These modifications are intended to further improve the performance of the model and achieve better results on our target tasks.

REFERENCES

[1] J. Ferryman and A. Shahrokni, “Pets2009: Dataset and challenge,” in *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*. IEEE, 2009, pp. 1–6.

[2] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” *arXiv preprint arXiv:2101.02702*, 2021.

[3] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding, “Learning a proposal classifier for multiple object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2443–2452.

[4] D. Xing, N. Evangeliou, A. Tsoukalas, and A. Tzes, “Siamese transformer pyramid networks for real-time uav tracking,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2139–2148.

[5] J. Wan, H. Zhang, J. Zhang, Y. Ding, Y. Yang, Y. Li, and X. Li, “Dsrrtracker: Dynamic search region refinement for attention-based siamese multi-object tracking,” *arXiv preprint arXiv:2203.10729*, 2022.

[6] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, “Global tracking transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8771–8780.

[7] J.-N. Zaech, A. Liniger, D. Dai, M. Danelljan, and L. Van Gool, “Learnable online graph representations for 3d multi-object tracking,” *IEEE Robotics and Automation Letters*, 2022.

[8] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “Fairmot: On the fairness of detection and re-identification in multiple object tracking,” *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.

[9] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, “Towards real-time multi-object tracking,” in *European Conference on Computer Vision*. Springer, 2020, pp. 107–122.

[10] B. Shuai, A. G. Berneshawi, D. Modolo, and J. Tighe, "Multi-object tracking with siamese track-rcnn," *arXiv preprint arXiv:2004.07786*, 2020.

[11] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[13] X. Zhu, Y. Jia, S. Jian, L. Gu, and Z. Pu, "Vitt: vision transformer tracker," *Sensors*, vol. 21, no. 16, p. 5608, 2021.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[15] J. Bi, Z. Zhu, and Q. Meng, "Transformer in computer vision," in *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*. IEEE, 2021, pp. 178–188.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[18] X. Chen, S. M. Iranmanesh, and K.-C. Lien, "Patchtrack: Multiple object tracking using frame patches," *arXiv preprint arXiv:2201.00080*, 2022.

[19] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *IEEE Transactions on Multimedia*, 2022.

[20] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense queries for multiple-object tracking," *arXiv preprint arXiv:2103.15145*, 2021.

[21] P. Blatter, M. Kanakis, M. Danelljan, and L. Van Gool, "Efficient visual tracking with exemplar transformers," *arXiv preprint arXiv:2112.09686*, 2021.

[22] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei, “Motr: End-to-end multiple-object tracking with transformer,” *arXiv preprint arXiv:2105.03247*, 2021.

[23] Y. Liu, T. Bai, Y. Tian, Y. Wang, J. Wang, X. Wang, and F.-Y. Wang, “Segdq: Segmentation assisted multi-object tracking with dynamic query-based transformers,” *Neurocomputing*, 2022.

[24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[25] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[26] G. Brasó and L. Leal-Taixé, “Learning a neural solver for multiple object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6247–6257.

[27] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song, and J.-N. Hwang, “Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9876–9886.

[28] J. Li, X. Gao, and T. Jiang, “Graph networks for multiple object tracking,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 719–728.

[29] J. He, Z. Huang, N. Wang, and Z. Zhang, “Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5299–5309.

[30] K. G. Quach, P. Nguyen, H. Le, T.-D. Truong, C. N. Duong, M.-T. Tran, and K. Luu, “Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 784–13 793.

[31] C. Ma, F. Yang, Y. Li, H. Jia, X. Xie, and W. Gao, “Deep trajectory post-processing and position projection for single & multiple camera multiple object tracking,” *International Journal of Computer Vision*, vol. 129, no. 12, pp. 3255–3278, 2021.

[32] H. Karunasekera, H. Wang, and H. Zhang, “Multiple object tracking with attention to appearance, structure, motion and size,” *IEEE Access*, vol. 7, pp. 104 423–104 434, 2019.

[33] O. Kesa, O. Styles, and V. Sanchez, "Joint learning architecture for multiple object tracking and trajectory forecasting," *arXiv preprint arXiv:2108.10543*, 2021.

[34] B. Wang, C. Fruhwirth-Reisinger, H. Possegger, H. Bischof, G. Cao, and E. M. Learning, "Drt: Detection refinement for multiple object tracking," in *32nd British Machine Vision Conference: BMVC 2021.* The British Machine Vision Association, 2021.

[35] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 104–119, 2019.

[36] W. Qin, H. Du, X. Zhang, Z. Ma, X. Ren, and T. Luo, "Joint prediction and association for deep feature multiple object tracking," in *Journal of Physics: Conference Series*, vol. 2026. IOP Publishing, 2021, p. 012021.

[37] Q. Yin, Q. Hu, H. Liu, F. Zhang, Y. Wang, Z. Lin, W. An, and Y. Guo, "Detecting and tracking small and dense moving objects in satellite videos: A benchmark," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[38] H. Shi, H. Ghahremannezhad, and C. Liu, "Anomalous driving detection for traffic surveillance video analysis," in *2021 IEEE International Conference on Imaging Systems and Techniques (IST).* IEEE, 2021, pp. 1–6.

[39] R. Sundararaman, C. De Almeida Braga, E. Marchand, and J. Pettre, "Tracking pedestrian heads in dense crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3865–3875.

[40] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, and X. Pan, "Mat: Motion-aware multi-object tracking," *Neurocomputing*, 2022.

[41] Z. Zou, J. Huang, and P. Luo, "Compensation tracker: Reprocessing lost object for multi-object tracking," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 307–317.

[42] X. Gao, Z. Shen, and Y. Yang, "Multi-object tracking with siamese-rpn and adaptive matching strategy," *Signal, Image and Video Processing*, pp. 1–9, 2022.

[43] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 372–12 382.

[44] 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, 2016, pp. 3464–3468.

[45] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, “Motion segmentation & multiple object tracking by correlation co-clustering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 140–153, 2018.

[46] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] C. Kim, L. Fuxin, M. Alotaibi, and J. M. Rehg, “Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9553–9562.

[48] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, “Multiple object tracking with correlation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3876–3886.

[49] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, “Quasi-dense similarity learning for multiple object tracking,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[50] Y. Song, P. Zhang, W. Huang, Y. Zha, T. You, and Y. Zhang, “Multiple object tracking based on multi-task learning with strip attention,” *IET Image Processing*, vol. 15, no. 14, pp. 3661–3673, 2021.

[51] N. Muller, Y.-S. Wong, N. J. Mitra, A. Dai, and M. Nießner, “Seeing behind objects for 3d multi-object tracking in rgb-d sequences,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6071–6080.

[52] C.-J. Liu and T.-N. Lin, “Det: Depth-enhanced tracker to mitigate severe occlusion and homogeneous appearance problems for indoor multiple-object tracking,” *IEEE Access*, 2022.

[53] C. Tan, C. Li, D. He, and H. Song, “Towards real-time tracking and counting of seedlings with a one-stage detector and optical flow,” *Computers and Electronics in Agriculture*, vol. 193, p. 106683, 2022.

[54] O. Kesa, O. Styles, and V. Sanchez, “Multiple object tracking and forecasting: Jointly predicting current and future object locations,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 560–569.

[55] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, “Track to detect and segment: An online multi-object tracker,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 352–12 361.

[56] Z. Sun, J. Chen, M. Mukherjee, C. Liang, W. Ruan, and Z. Pan, “Online multiple object tracking based on fusing global and partial features,” *Neurocomputing*, vol. 470, pp. 190–203, 2022.

[57] H. Liang, T. Wu, Q. Zhang, and H. Zhou, “Non-maximum suppression performs later in multi-object tracking,” *Applied Sciences*, vol. 12, no. 7, p. 3334, 2022.

[58] J. He, X. Zhong, J. Yuan, M. Tan, S. Zhao, and L. Zhong, “Joint re-detection and re-identification for multi-object tracking,” in *International Conference on Multimedia Modeling*. Springer, 2022, pp. 364–376.

[59] S. Guo, J. Wang, X. Wang, and D. Tao, “Online multiple object tracking with cross-task synergy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8136–8145.

[60] T. Liang, L. Lan, X. Zhang, and Z. Luo, “A generic mot boosting framework by combining cues from sot, tracklet and re-identification,” *Knowledge and Information Systems*, vol. 63, no. 8, pp. 2109–2127, 2021.

[61] L. Ke, X. Li, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, “Prototypical cross-attention networks for multiple object tracking and segmentation,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[62] H. Fu, J. Guan, F. Jing, C. Wang, and H. Ma, “A real-time multi-vehicle tracking framework in intelligent vehicular networks,” *China Communications*, vol. 18, no. 6, pp. 89–99, 2021.

[63] J. Peng, T. Wang, W. Lin, J. Wang, J. See, S. Wen, and E. Ding, “Tpm: Multiple object tracking with tracklet-plane matching,” *Pattern Recognition*, vol. 107, p. 107480, 2020.

[64] D. M. Nguyen, R. Henschel, B. Rosenhahn, D. Sonntag, and P. Swoboda, “Lmgp: Lifted multicut meets geometry projections for multi-camera multi-object tracking,” *arXiv preprint arXiv:2111.11892*, 2021.

[65] D. Stadler and J. Beyerer, “Improving multiple pedestrian tracking by track management and occlusion handling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 958–10 967.

[66] E. Yu, Z. Li, and S. Han, “Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking,” *arXiv preprint arXiv:2203.14208*, 2022.

[67] G. Wang, Y. Wang, R. Gu, W. Hu, and J.-N. Hwang, “Split and connect: A universal tracklet booster for multi-object tracking,” *IEEE Transactions on Multimedia*, 2022.

[68] D. M. Nguyen, R. Henschel, B. Rosenhahn, D. Sonntag, and P. Swoboda, “Lmgp: Lifted multicut meets geometry projections for multi-camera multi-object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8866–8875.

[69] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[70] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, “Joint discriminative and generative learning for person re-identification,” in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2138–2147.

[71] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres, “Efficient decomposition of image and mesh graphs by lifted multicuts,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1751–1759.