GROUP TRAJECTORY ANALYSIS IN SPORT VIDEOS

Thesis

Submitted to

The College of Arts and Sciences of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Computer Science

By

Shreenivasan Duraivelan

Dayton, Ohio

May 2021

GROUP TRAJECTORY ANALYSIS IN SPORT VIDEOS

Name: Shreenivasan Duraivelan

APPROVED BY:

Tam V. Nguyen, Ph.D.
Faculty Advisor and Committee Chair

James P. Buckley, Ph.D.
Committee Member

Luan Nguyen, Ph.D.
Committee Member

ABSTRACT

GROUP TRAJECTORY ANALYSIS IN SPORT VIDEOS

Name: Duraivelan, Shreenivasan
University of Dayton

Advisor: Tam V. Nguyen

Trajectory Prediction is the problem of predicting the short-term and long-term spatial coordinates of various agents such as cars, buses, pedestrians. In Trajectory Prediction we use the moment of an agent's past trajectories to predict action or course the agent is going to make in the future. It is difficult to predict trajectory of a player since various factors after the decision which a player makes, leading to his next step. We propose a model which uses video input from a sport, then extracting the data from the video to extract the players movement pattern from which the trajectory of the player is predicted through trajectory prediction Network. We implement object detection and efficient tracking to extract the players position information from the dataset. Our model has a Mean Average Displacement (MAD) of 8.35

Dedicated to my parents

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my supervisor, Dr. Tam V. Nguyen, who has the substance of a genius: he convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Dr. Nguyen supported me with all of the required resources to complete the thesis, as well as for providing me with research opportunities and for his insightful advice, persistence. Without his persistent help, the goal of this thesis would not have been realized.

Additionally, I would like to thank my parents for all their love and support throughout my studies and encouraging me to purse higher education. Furthermore, I wish to thank all of my friends whose assistance was a milestone in the completion of this thesis. Last but not the least, I would also like to thank all professors in Department of Computer Science, University of Dayton for their kind support during my master's study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND NOTATIONS

MAD	Mean Average Displacement
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
GAN	Generative Adversarial Network
STGCNN	Spatio-Temporal Graph Convolutional Neural Network
MOF	Multiple Object Forecasting
MSE	Mean Squared Error
DTW	Dynamic Time Warping
MCTF	Multi-Camera Forecasting Database
CIOU	Complete IoU
FAD	Final Average Displacement
SORT	Simple Real Time Tracker

CHAPTER I

INTRODUCTION

The term “trajectory” Is used to describe the path an object follows as it moves through space. Trajectory of an object is affected by various factors such as force applied on it, surface and obstacles. Trajectory prediction is the concept of Predicting the spatial coordinates of various object using those factors [1]. The future trajectory of an agent is predicted using the agent’s past trajectories.

The trajectory of Human can be influenced by multiple factors such as scene topologies, Human beliefs, and the most complex one, human-human interactions [2]. there are many variables which are strongly relevant for the trajectories of single Agent: The nature of the surrounding obstacles and their spatial distribution, the nature of the ground, the long-term goal of the agent, his age and his mental state [3].

Due to the current advancements in automated cars like the new Tesla, trajectory projection plays a major role in the entire design and working of their systems. Predicting other traffic participants trajectories is a crucial task for an autonomous vehicle, in order to avoid collisions on its planned trajectory, since it depends on each driver's intention and driving habits. However, certain considerations about vehicle dynamics can provide partial or fuzzy knowledge on the future.

Most of the pedestrian trajectory modules, learns the interaction between surrounding and a pedestrian. These models are good in predicting trajectories of pedestrians, but they do not perform well on predicting the trajectories of players since there are various other factors which affect the action the players take.

In the recent years, the use of LSTM in trajectory prediction are in raise. LSTM – Long Short-Term Memory is an artificial recurrent neural network (RNN) architecture used in the field of deep learning [4]. Unlike standard feedforward neural networks, LSTM has feedback

connections, which enables it to learn the sequence order of the parameters[5]. This helps the LSTM based trajectory prediction model to learn the sequence pattern of the pedestrian movements. Generic prediction of pedestrian trajectory produces good result, But for our implementation of trajectory prediction of players in sports video, LSTM based model would not produce proper results. Figure 1 shows the comparison of different LSTM models in pedestrian trajectory prediction.

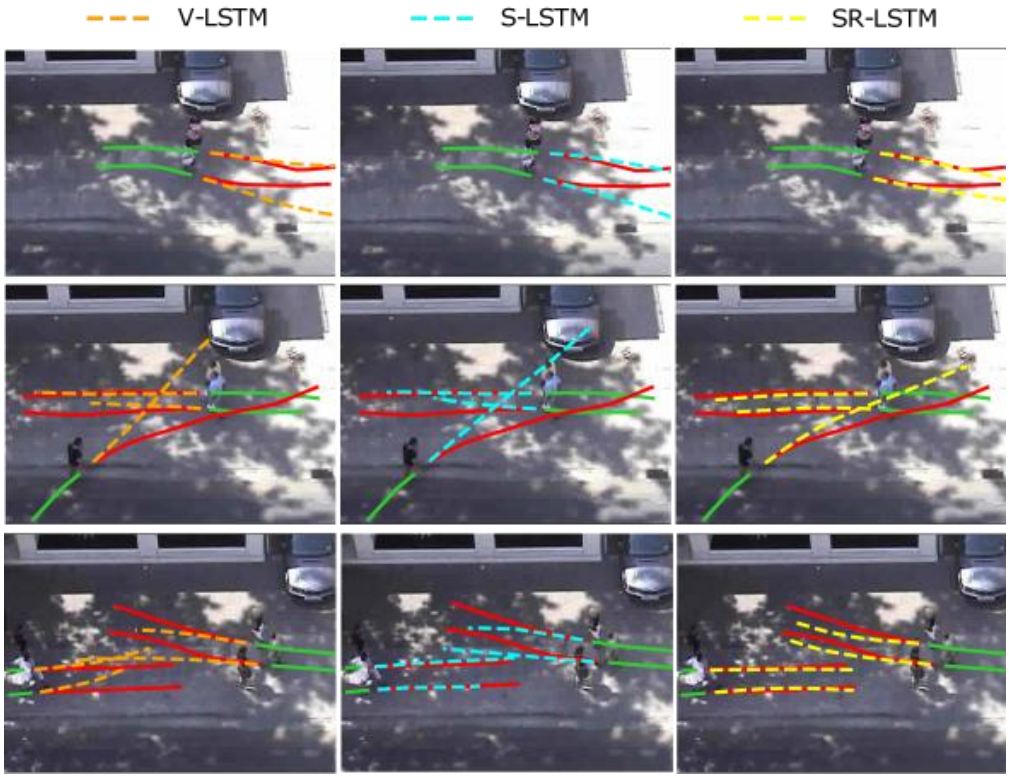

Figure 1. Comparison of different LSTM model pedestrian trajectory prediction [2].

Hence, we propose a Trajectory Prediction Model which utilizes Transformer Neural Networks. Transformers were developed to solve the problem of sequence transduction [6], or neural machine translation, including tasks that transforms an input sequence into an output sequence. Transformer Neural Networks are widely used in Natural Language Processing due to its efficiency in predicting the sequence correlation between the input text sequence and the translated text sequence.

Transformer Neural Networks uses Attention mechanism to improve its efficiency. Attention mechanisms are input processing techniques for neural networks that allows the network to focus on specific aspects of a complex input, one at a time until the entire dataset is categorized [6]. This enables the Transformer Neural Networks to get better results over the on non-attention-based models such as Long Short-Term Memory model in sequence-to-sequence predictions.

In our model we extract the trajectories of a player in a video using object detection and tracking mechanism, which generates the player coordinate dataset. We implement a Transformer neural network model to learn the correlation between the current trajectory and the feature trajectory.

CHAPTER 2

RELATED WORK

In this chapter, we review the earlier work to predict trajectory of agent and object using various methods. We also discuss about the problem of tracking object in multiple camera view. At last, we discuss about attention based neural network, namely Transformer Networks which we implement in our model.

2.1 Models

Liang et al. [7] proposed a model to predict future person activities and locations in videos. This model is an end-to-end, multi-task learning system utilizing rich visual features about human behavioral information and interaction with their surroundings. This model utilizes person behavior module and person interaction module to encode rich visual semantics into a feature tensor. Employing two modules to encode rich visual information about each person's behavior and interaction with the surroundings. Person interaction module looks at the interaction between a person and their surroundings. Trajectory generator summarizes the encoded visual features and predicts the future trajectory by the LSTM decoder with focal attention. Activity prediction utilizes rich visual semantics to predict the future activity label for the person. In addition, they divide the scene into a discretized grid of multiple scales, the Manhattan Grid, to compute classification and regression for robust activity location prediction. In another work, Liang et al. [8] proposed a model of Garden of Forking Paths towards multi-future trajectory prediction. This studies the problem of predicting the distribution over multiple possible future paths of people as they move through various visual scenes. They gather new dataset, created in a realistic 3D simulator, which is based on real world trajectory data, and then Annotating the image based on their requirement. This provides the first benchmark for quantitative evaluation of the models to predict multi-future trajectories. The other contribution is creating a new model which produces multiple feature trajectories, using multi-scale location encodings and convolutional RNNs over graphs. The focus

in on predicting the locations of a single agent for multiple steps into the future, given a sequence of past video frames and agent locations. There is multiple inherent uncertainty in this task. To combat the uncertainty, they design a model that can effectively predict multiple plausible future trajectories, by computing the multimodal distribution p (Lh+1: T |L1:h, V1:h).

Meanwhile, Amirian et al. [9] introduced a model to learn multi-modal distributions of pedestrian trajectories with GANs. They evaluate the usage of GAN to Predict the trajectory of pedestrian based on their interaction with other agents. Social Ways GAN generates independent random trajectory samples that mimic the distribution of trajectories among the training data. Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset. GANs are a way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either domain or generated. The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fails over half the time, which implies that the generator model is generating good examples. The Discriminator is trained to detect fake samples from real samples and the Generator is used for generating new samples. Using this properly of GAN in trajectory prediction, The Encoder block in Generator contains one layer of 128 LSTM units (LSTM-E), Discriminator: uses two LSTM blocks (LSTM-OE and LSTM-PE). Due to the implementation of the GAN, the distribution of generated trajectories is closer to the ground truth. Chandra et al. [10] proposed predicting trajectory and behavior of road-agents using spectral clustering in Graph-LSTMs. An approach for traffic forecasting in urban traffic scenarios using a combination of spectral graph analysis and deep learning. Graph Clustering is the process of grouping the nodes of the graph

into clusters, considering the edge structure of the graph in such a way that there are several edges within each cluster and very few between clusters. This model predicts the low-level information (future trajectories) as well as the high-level information (road-agent behavior) from the extracted trajectory of each road-agent. Uses a two-stream graph convolutional LSTM network to perform traffic forecasting using these weighted traffic-graphs. Mohamed et al. [11] presented a social spatio-temporal graph convolutional neural network for human trajectory prediction. Evaluates the use of Social Spatio-Temporal Graph Convolutional Neural Network (Social-STGCNN), which substitutes the need of aggregation methods by modeling the interactions as a graph. The usage of STGCNN makes the model more data efficient which also leads to faster interface time. A temporal graph is, informally speaking, a graph that changes with time. When time is discrete and only the relationships between the participating entities may change and not the entities themselves, a temporal graph may be viewed as a sequence of static graphs over the same set of nodes V. The ST-GCNN conducts spatiotemporal convolution operations on the graph representation of pedestrian trajectories to extract features. TXP-CNN takes these features as inputs and predicts the future trajectories of all pedestrians. The trajectory data set are in the form of graph, this model has better results when compared to GAN based trajectory prediction.

In a different approach, Kothari et al. [12] proposes a model for human trajectory forecasting in crowds in a deep learning perspective. Recently, deep learning methods have outperformed their handcrafted counterparts, as they learned about human-human interactions in a more generic data-driven fashion. This model analysis the present deep learning-based methods for modelling social interactions. This evaluates the, The presence and the impact of the surrounding and the motion of the person and object near the agent. The modelling of the observation of one sequence which leads to the change in another sequence is an essential prerequisite for a good human trajectory forecasting model. They use a LSTM model to process the observation and predict the trajectory of the agent. A good human trajectory forecasting model should provide physically

acceptable outputs, for instance, the model prediction should not undergo collisions. Quantifying the physical feasibility of a model prediction is crucial for safety-critical applications. Liang et al. [13] proposes a model for learning robust representations from 3D simulation for pedestrian trajectory prediction in unseen cameras. Addresses the problem of predicting future trajectories of people in unseen scenarios and camera views. Proposes a method to utilize multi-view 3D simulation data for training. This approach finds the hardest camera view to mix up with augmented data from the original camera view in training, this generates new data sets which are deferent from the once that are generated from the original camera view, it enables to predict the trajectory of pedestrians in an efficient manner since they have more unseen data set to predict some uncertainties which are not extractable from the original view. The model is trained on Multi-verse model on simulation data set that can effectively predict the future trajectory in real-world test videos that are unseen during training.

Styles et al. [14] multiple objects forecasting model predicting future object locations in diverse environments. This paper introduces the problem of multiple object forecasting (MOF), in which the goal is to predict future bounding boxes of tracked objects. they formulate the problem based on the perspective of an agent and call for the prediction of full object bounding boxes, rather than trajectories alone. This paper adapts existing trajectory forecasting methods for MOF and confirm cross-dataset generalizability on the MOT-17 dataset without fine-tuning. Model an encoder-decoder architecture for MOF. STED combines visual and temporal features to model both object-motion and ego-motion. Guen et al. [15] studies the shape and time distortion loss for training deep time series forecasting models. Addresses the problem of time series forecasting for non-stationary signals and multiple future steps prediction. Introduces Distortion Loss including Shape and Time, an objective function for training deep neural networks. Distortion Loss including Shape and Time aims at accurately predicting sudden changes, incorporates shape and temporal change detection. Introduces a differentiable loss function suitable for training deep neural networks and a back

propagation network for optimization. This also introduce a variant of Distortion Loss including Shape and Time, which provides a smooth generalization of temporally constrained Dynamic Time Warping. Experiments carried out on various non-stationary datasets reveal the very good behavior of this model is compared to models trained with the standard Mean Squared Error (MSE) loss function, and to DTW and variants.

Gomez-Gonzalez et al. [16] proposes a model for real time trajectory prediction using deep conditional generative models. Data driven methods for time series forecasting that quantify uncertainty open new possibilities for robot tasks with hard real time constraints, allowing the robot system to make decisions that tradeoff between reaction time and accuracy in the predictions. Despite the recent advances in deep learning, it is still challenging to make long term accurate predictions with the low latency required by real time robotic systems. This paper proposes a deep conditional generative model for trajectory prediction that is learned from a data set of collected trajectories. This method uses encoder and decoder deep networks that map complete or partial trajectories to a Gaussian distributed latent space and back, allowing for fast inference of the future values of a trajectory given previous observations. The encoder and decoder networks are trained using stochastic gradient variational Bayes. Styles et al. [17] proposes a multi-camera trajectory forecasting model for pedestrian trajectory prediction in a network of cameras. This paper introduces the task of multi-camera trajectory forecasting, Which Uses multiple cameras to extract the data, and predict the trajectory of the pedestrian on an array of cameras. This addresses the challenging scenario of forecasting across multiple non-overlapping camera views. They create their own dataset which is Warwick-NTU Multi-camera Forecasting Database, it consists of data set for pedestrian across multiple non overlapping camera. They develop a semi-automated annotation method for labelling the dataset which has over 600 hours of video data. An effective MCTF model should proactively anticipate where and when a person will re-appear in the camera network. This model considers the task of predicting the next camera a pedestrian will re-appear

after leaving the view of another camera and present several baseline approaches. Diodato et al. [18] proposes a model for accurate trajectory prediction for autonomous Vehicles. Predicting vehicle trajectories, angle and speed is important for safe and comfortable driving. This paper demonstrates the best predicted angle, speed. The contributions of this paper are, (i) implementing general neural network system architecture which embeds and fuses together multiple inputs by encoding, and decodes multiple outputs using neural networks, (ii) using pre-trained neural networks for augmenting the given input data with segmentation maps and semantic information, and (iii) leveraging the form and distribution of the expected output in the model.

Recently, Giuliari et al. [19] proposed transformer networks model for trajectory forecasting. The LSTM is based on sequentially processing sequences and storing hidden states to represent knowledge about the people, e.g., its speed, direction and motion pattern. Transformer networks use attention instead of sequential processing. They propose a multi-agent framework where each person is modelled by an instance of our transformer network. Each Transformer Network predicts the future motion of the person as a result of its previous motion. TF is a modular architecture, both the encoder and the decoder are composed of 6 layers, each containing three building blocks: an attention module, a feed-forward fully connected module and two residual connections after each of the previous blocks. Attention module is responsible for learning the non-linearity in the sequence. The encoding stage creates an observation sequence, which makes the model memory. The decoder predicts auto-regressively the future track positions. At each new prediction step, a new decoder query is compared against the encoder keys and values and against the previous decoder prediction to yield the next-step prediction.

CHAPTER 3

PROPOSED FRAMEWORK

3.1 Framework

We propose a model to extract images from a video and then use the dataset to predict the trajectory of a player in sports video. Out model consist of four segments, (i) Dataset (ii) object-detection (iii) object-tracking (iv) trajectory prediction.

From the dataset we train an object detection model to extract the features in the dataset, then we track the object through the video using the object tracking model. We implemented two object detection models and tracking models, which are yolov4 object detection, Faster R-CNN object detection, KCF tracking and Deep-SORT tracking algorithms.

We get the player coordinates from the Tracking algorithm and then use it in the Trajectory prediction stage for training the model. We use Transformer Neural Networks for trajectory prediction, since it is Attention based model. Figure 2 illustrates the model flowchart.

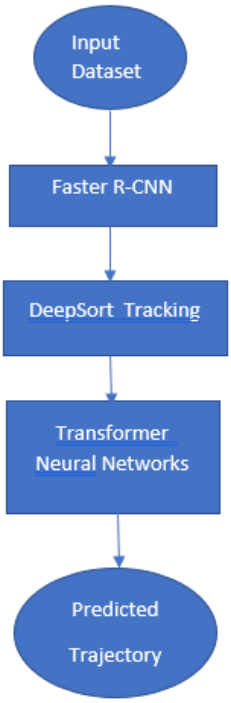

Figure 2. Model Flowchart.

3.2 Dataset

3.2.1 Video Extraction

We create a custom dataset for our model. We recorded a video of a basketball game of various length. We used the widows inbuild Xbox screen recorder to record the video. Combined the video using a video editing tool to create the training video dataset.

3.2.2 Data Annotation

Data annotation is the process of labeling or classifying a data using text, annotation tools, or both, to show the data features you want your model to recognize on its own. Proper Data annotation is crucial for precise detection of an object in the dataset. We used an image annotation tool (labeling) for this process. We annotated the data set with 1 class which is the player class. Improper annotation of the dataset might lead to miss identifying the people outside the court as players. Hence, we carefully annotated the data and then re-check the entire annotated data set for any errors. The annotation information is saved in the form of text file which consist of the top left x, top left y, height and width coordinate information of the object in the data set. Figure 3 shows the example of our annotated data.

Figure 3. Example for annotated data.

3.3 Object Detection

Object detection is a method related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in images or videos. We Implemented Two types of object detection models which are yolov4 object detection and Faster R-CNN object detection.

3.3.1 YOLOV4

A YOLO object detector uses a single neural network to predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network [20], it can be optimized end-to-end directly on detection performance. YOLO makes more localization errors but is less likely to predict false positives on background. it identifies objects more rapidly and more precisely than other recognition systems.

YOLO is based on Convolutional Neural Networks. It divides the image data into various regions, visualize the confirmed edges and probabilities of every region. Simultaneously, it also anticipates various confirmed edge boxes and probability of the respective classes [21].

YoloV4 is an improved version of Yolo algorithm, the implementation of a new architecture in the Backbone and the modifications in the Neck have improved the Mean Average Precision.

The YOLOV4 Architecture consists of three components, which are (i) Backbone (ii) Neck (iii) Dense Prediction. YOLOv4 is a one-stage detector. The backbone has the feature extraction architecture. There is multiple backbone architecture for yolov4 such as VGG, Darknet53, ResNet. We use a CSPDarknet53 architects as our backbone for our model. Darknet53 is more accurate but it is slower when compared to other architectures. Since we do not predict trajectory in real time, the speed of the prediction is not important for our model.

The purpose of the neck block is to add extra layers between the backbone and the dense prediction layer. The Head (Prediction) block is where we locate all the bounding boxes and classify the object present in that box. YOLOV4 also implements Bag Of-Freebies and Bag of Specials.

Bag of Freebies [20] methods increase the cost of training or change the training strategy without increasing the inference time. Figure 4 shows the flowchart of BoF. Bag of specials [20] methods increase inference cost by a small amount but can significantly improve the accuracy of object detection. YOLOv4 implements CIOU loss function.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{3.1}$$

$$\alpha = \frac{v}{(1-\mathrm{IoU})+v} \tag{3.2}$$

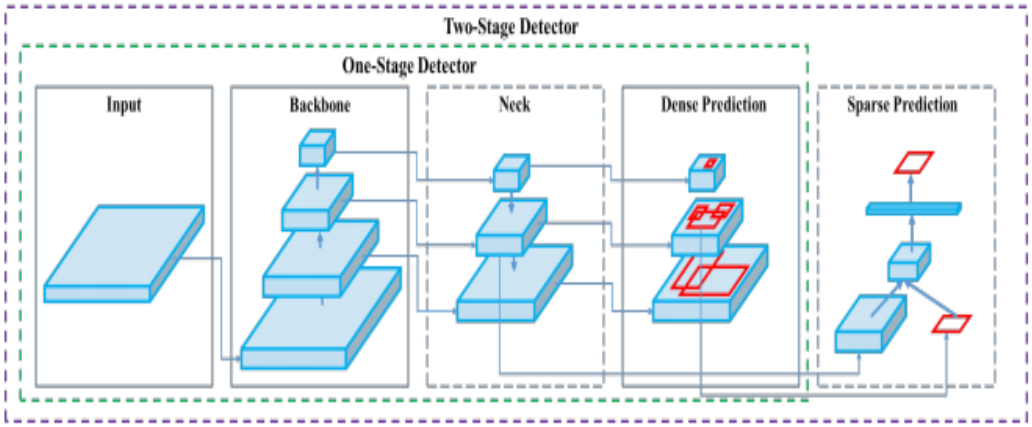

Figure 4. Object detector with a two-stage mechanism [20].

Figure 5 illustrates the detection results of YOLO algorithm using Bag of Freebies mechanism. Most of the players are well detected.

Figure 5. Output from YOLOv4 Object Detection Stage.

3.3.2 Faster R-CNN

Faster R-CNN is a region-based convolution neural network. In the convolution layers we train filters to extract the appropriate features the image. Convolution networks are generally composed of Convolution layers, pooling layers and a last component which is the fully connected or another extended thing that will be used for an appropriate task like detection. Fast R-CNN overcomes several issues in R-CNN. As its name suggests, one advantage of the Fast R-CNN over R-CNN is its speed. As shown in Figure 6, Faster R-CNN uses ROI polling layer instead of calculation for each proposal independently. This increases the speed faster R-CNN. We use the annotated data as input for training. The output prediction has the bounding box of the object in the image along with a score which refers to the confidence level of that object belonging to a desired class. Figure 7 shows the exemplary output of Faster RCNN detection.

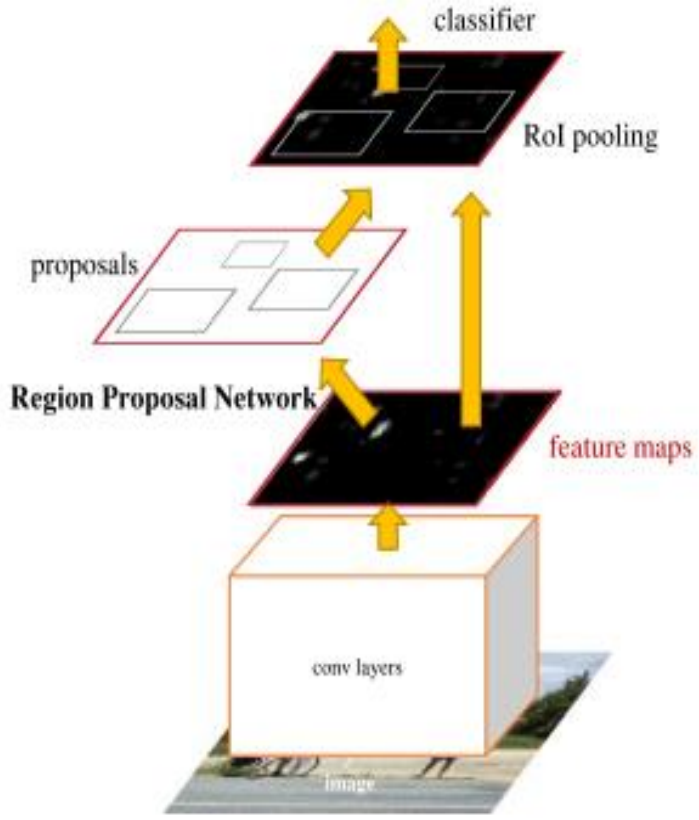

Figure 6. Faster R-CNN framework[22].

Figure 7. Example output from Faster R-CNN object detection.

3.4 Object Tracking

Visual Object Tracking is one of crucial task in Computer Vision, which is used to locate a certain object in the frames of a video, given its location in the initial frame where the object appeared for the first time. We use The Object Tracking stage to track the object which are detected from the detection stage. For each frame, the data is passed to the object detection stage to get the object coordinates and if the object detected is a new object, then it is assigned a new identification and then tracked. We use matching algorithms to find where the object detected is already being tracked.

3.4.1 Kernelized Correlation Filter

The idea of Kernelized Correlation Filter (KCF) is to estimate an optimal image filter such that the filtration with the input image produces a desired response. The desired response is typically of a Gaussian shape centered at the target location; therefore, the score decreases with the distance [23].

3.4.2 DeepSORT

The DeepSort algorithm is the extension of SORT (simple Real Time Tracker). The SORT algorithm consists of four steps Detection, Estimation, Target association, Track Identity creation and destruction [24]. The detection layer has one of the object detections models. In our case we use YOLOV4 and Faster R-CNN for detection.

In the estimation layer we propagate the detection from the current frame to the next frame using a linear constant velocity model. Constant velocity means that the object in motion is moving in a straight line at a constant speed.

$$x = x_0 + v_t x = x_0 + v_t \tag{3.3}$$

where x_0 represents the position of the object at $t = 0$, and the slope of the line indicates the object's speed. In the target association layer, the target's bounding box is computed by predicting its new position in the current frame. In summary, DeepSort algorithm is run as follows:

- The objects are detected in the image.
- Existing tracks positions are updated using a Kalman filter.
- Cluster the tracks by age (how long the tracks as not been associated with a detection) and run the Hungarian algorithm on each of the cluster in increasing age order.
- All the remaining unmatched and unconfirmed tracks of age 1 are processed using the original SORT algorithm.
- Finally, un-matched detection is set as new tracks.

Figure 8. Sample output from DeepSort tracking.

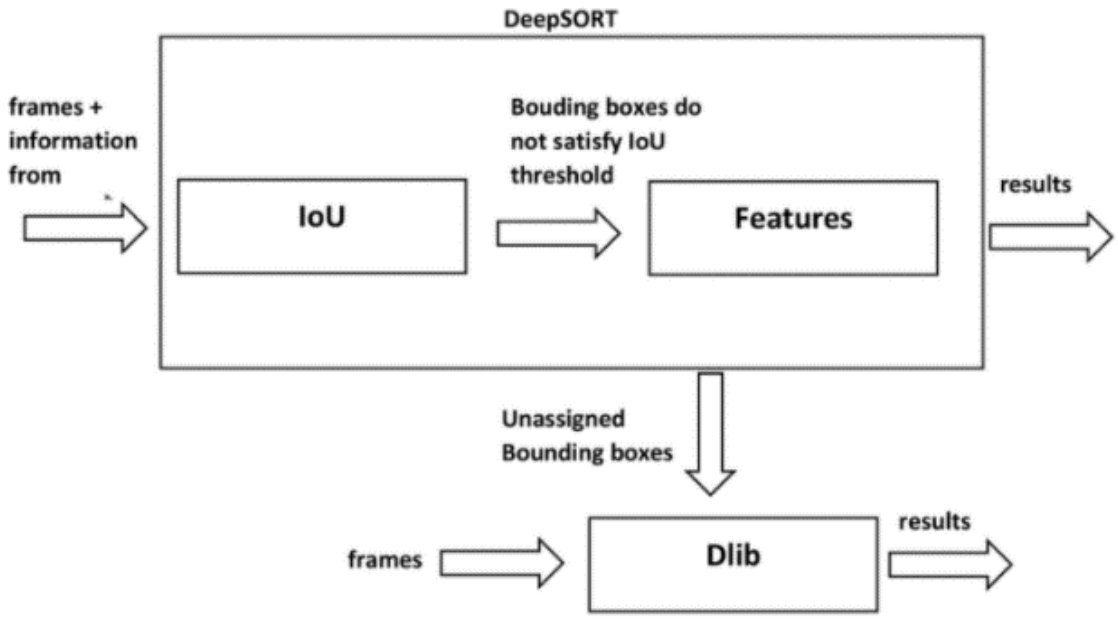

Figure 9. DeepSort tracking framework [24].

Figure 8 shows the exemplary results of DeepSORT tracking on a video. Meanwhile, Figure 9 depicts the tracking mechanism of DeepSORT.

3.5 Trajectory Prediction

Trajectory Prediction is the process of predicting the trajectory of an object or agent based on its previous recorded movements/actions. In our case we use trajectory prediction to predict the movement of a player in a sports video. We get the input for the trajectory prediction block from the object tracking block and by using a transformer neural network we train our model to predict the trajectory.

3.5.1 Transformer Network

Transformer Network was developed to solve the problem of sequence transduction and neural machine translation, which includes track that transforms an input sequence to a output sequence. Transformer Neural Networks uses Attention [6] Mechanism which focus on subset of information passed to them, in this method, instead of encoding the whole sequence to a hidden

state, the element of the sequence is encoded to its corresponding hidden states that is passed to the decoding state.

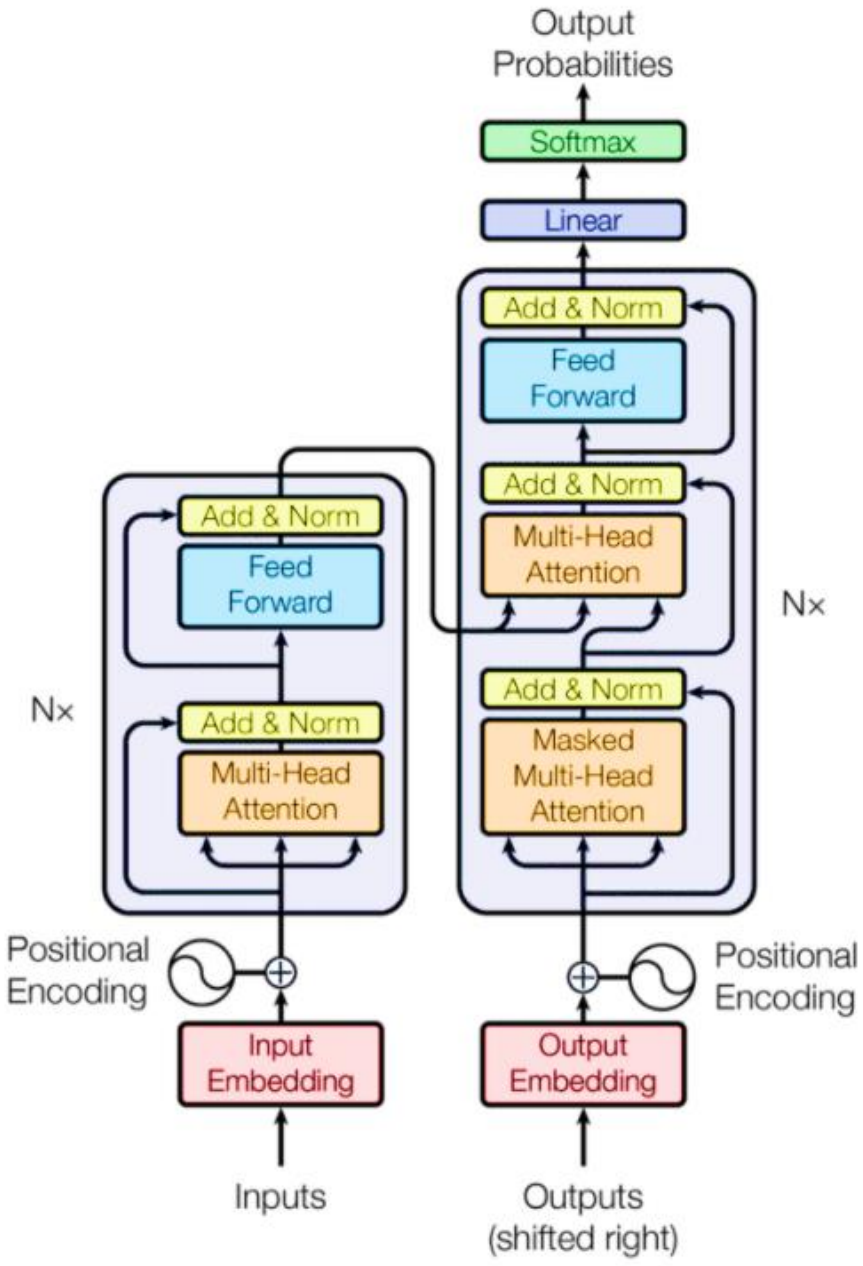

Figure 10. The input and output of the Transformer model [6].

As shown in Figure 10, Transformer Neural Network has two blocks, the encoder block and the decoder block. Both blocks consist of 6 layers each [6,19]. The Transformer Networks

maintain the encoder memory separate from the decoder sequence. The precision of the networks capturing the non-linearity of a given sequence is mainly affected by the attention layer. For each attention module present in this block, a sequence query is compared to all other sequence by using a scaled dot product [6].

$$Attention(Q, K, V) = SoftMax\left(\frac{QK^T}{\sqrt{dk}}\right)V \quad (3.2)$$

, where Q is the query sequence, K is the key and V is the value matrix.

The encoder block is used for creating representation of the input sequence, based on the attention mechanism, this makes the memory block. After encoding the input, the encoder block generates two vectors which are Key's 'K' and the values 'V', these outputs are then passed to the decoding block.

The decoder predicts the future position of the element in the sequence. At each new prediction, the entity decoder query 'Q' is compared to the encoder Key 'K' and the encoder values 'V' [6].

3.6 Implementation

In our model we have three blocks, (i) Object detection, (ii) Object tracking (iii) Object tracking. In the Object detection block we use a Faster R-CNN object detection algorithm based on Inception V2 CNN architecture. The inception V2 has a 3x3 Convolution and it also converts the n x n factorization to 1 x n and n x 1 factorizations. This increases the speed of the objection model. We mainly chose Faster-RCNN over yolov4 even though Faster R-CNN is comparatively slower, for our final modal because of its high accuracy. We use tensorflow1 for implementation the Faster R-CNN. In the Object tracking stage we mainly use the Deep Sort tracking with its backbone as CSPDarknet53. We use this implementation of deep sort because of its high precision in avoiding false positive.

In the Trajectory Prediction stage, we use Transformer Networks. We use the transformer network with parameters of the original transformer [6] with dmodel = 512, 8 attention head and 6 layers. We use drop out value of 0.1 and use L2-loss between predicted and ground truth player position. We train the network with backpropagation using Adam.

CHAPTER 4

EVALUATION

In this chapter we first discuss about the dataset which we used for evaluating our model. After that we do evaluate our object detection model. And at last, we evaluate our trajectory prediction model and discuss about the impact of object detection and tracking on the predicted trajectory.

4.1 Dataset

We used our custom dataset for evaluation our model since we had a unique set of data which we required for our implementation. For the video dataset, we captured video from NBA game. We captured video from 5 matches each of length about 16 minutes, then combined the video using a video editor and used it as our training dataset. For testing data, we captured multiple videos of length 2 minute for individual testing.

4.2 Metrics

We use MAD and FAD metric to evaluate our model. MAD is mean average displacement, it is used for measuring the fit of the prediction with respect to the ground truth, averaging the discrepancy at each time step. FAD is Final Average Displacement; it is used for evaluation the displacement error at the final step.

4.3 Object Detection

We used two object detection models (YOLOV4 and Faster R-CNN), using 1300 images as training data set. We annotated the data with 1 class and up to 10 objects in a single image. For the validation dataset we used 100 different annotated images.

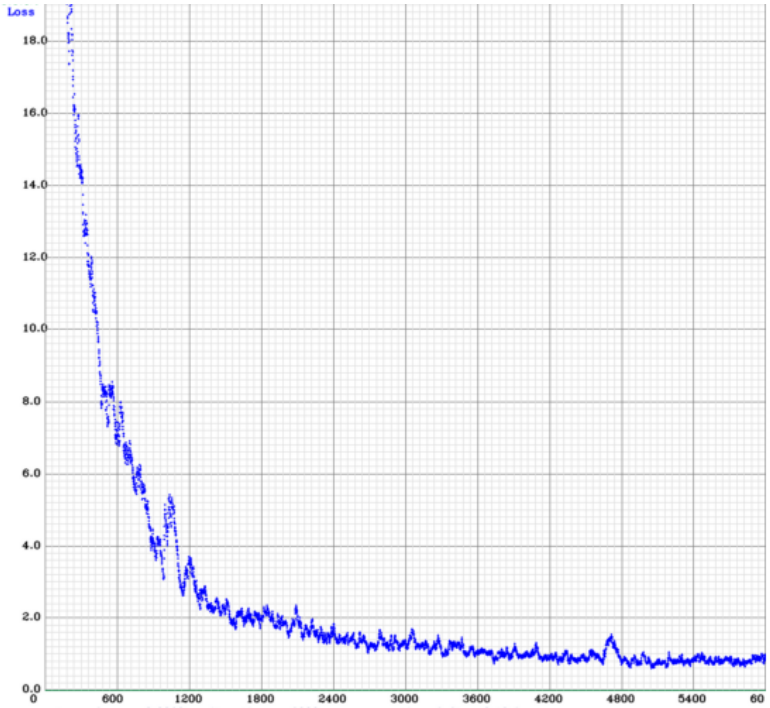

Figure 11. Training loss for YOLOv4 object detection.

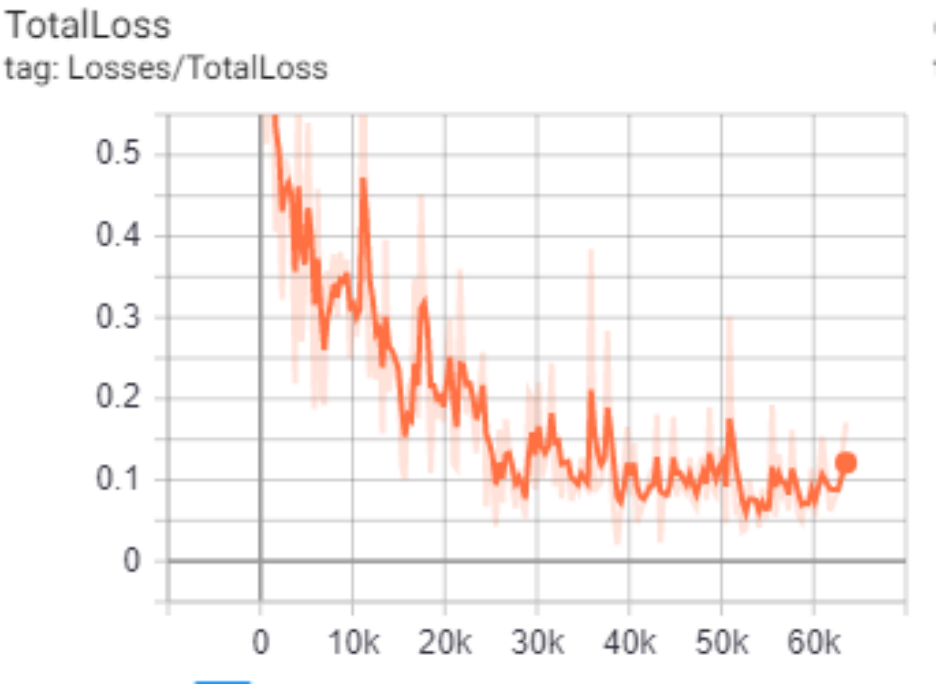

Figure 12. Training loss for Faster R-CNN.

While testing we found that Faster R-CNN had better accuracy in prediction the score of the class when compared to YOLOv4, but it took longer for the Faster R-CNN model to perform prediction in the dataset.

4.4 Trajectory Prediction

In total we used 4 combination of object detection and tracking, which are (i)YOLOv4 detection with DeepSort tracking, (ii)Faster R-CNN with DeepSort Tracking, (iii) YOLOv4 with KCF tracking and (iv) Faster R-CNN with KCF tracking.

We extract the bounding box coordinates, tracker id, frame id from the video using the tracking algorithm. Then we store that information to a text for training and testing dataset for the trajectory prediction stage.

Table 1. Mean Average Displacement for the trajectories predicted.

	DeepSort	KCF
YOLOv4	19.320	21.814
Faster R-CNN	8.358	9.498

Table 2. Final Average Displacement for the trajectories predicted.

	DeepSort	KCF
YOLOv4	38.751	40.325
Faster R-CNN	15.745	16.879

We got good results when using Faster R-CNN in our detection stage, but it takes longer to detect the object using Faster R-CNN. Faster R-CNN uses Region Proposal Network, it feeds the input to the backbone CNN, the input is resized to 600, 1024 on minimum and maximum dimension, respectively. It also implements inception v2 architecture, which is known for its accuracy.

YOLOv4 has comparatively worse result for predicting trajectory. YOLOv4 uses a CSPDarknet backbone. It has a decent accuracy, but it is optimized to run faster, hence it is designed to process more frame by compromising its accuracy. For real-time object detection YOLOv4 produces way better results since it runes over 2x faster than Faster R-CNN.

DeepSort tracker provided better results in our experiment. The DeepSort uses Deep learning method to match features and perform the tracking. It has a better precision in avoiding false positive, while maintain the tracking accuracy. Hence, it gives better trajectory when passed to the trajectory prediction model.

KCF tracking is an inbuild tracking algorithm in OpenCV. KCF is a tracking framework that utilizes properties of circulant matrix to enhance the processing speed.

$$\min_{\alpha} \left|\left|K\alpha - \mathrm{y}\right|\right|^2 + \lambda\alpha^{\mathrm{T}} K\alpha \tag{4.1}$$

$$\alpha = (\mathrm{K} + \lambda \mathrm{I})^{-1} y \tag{4.2}$$

$$\alpha' = \frac{y'}{k'^{xz} + \lambda} \tag{4.3}$$

$$r' = k'^{xz} \odot \alpha' \tag{4.4}$$

$$k^{xx^{"}} = exp\left(-\frac{1}{\sigma^2}\left(\left|\left|x\right|\right|^2 + \left|\left|x^{"}\right|\right|^2 - 2F^{-1}(x'^{*} \odot x^{"'})\right.\right. \tag{4.5}$$

, where matrix K is the kernel matrix with elements of dot products $k_{i,j}=\varphi(x_i)^T\varphi(x_j)k_{i,j}=\varphi(x_i)^T\varphi(x_j)$. The original problem thus becomes non-linear, kernelized ridge regression.

From the result of trajectory prediction (Table 1 and Table 2), we observe that the detection stage and the tracking stage affects the MAD and FAD by a huge margin, since better detection and tracking will give us a better coordinate of the agent, which leads us to plot the trajectory of the player with better accuracy. Having more accurate trajectories improves the training of the transformer network. This leads to having better prediction and provides us with better testing trajectories.

CHAPTER 5

CONCLUSION

In this thesis we proposed a model to analyze the trajectory of group of players in the sports video. We collected dataset video from an NBA game, and then annotating data for our training. We studied the impact of different parameters had on our trajectory prediction model.

We implement different models for tracking and object detection. Secondly, we implemented Transformer network for trajectory prediction. Through the experimentation we found that the object detection and tracking model have huge impact on the final trajectory prediction. Even though we have high MAD and FAD, it is still can be further improved by implementing a better object tracking and detection algorithm. The factor of moving camera affects the accuracy of prediction as well, along with the players personal habits which affects the decision they make in the game.

Since, our model is generalized for all the players, it does not capture those features. We conclude that our model has improved result, but still it cannot produce good enough result which can be used for real time analysis. We believe this will encourage more research work in the future. With the improvement in the tracking and object detection technology along with improved trajectory prediction model will get us close to predicting accurate trajectories.

CHAPTER 6

FUTURE WORK

There are various factors affecting the prediction of trajectory such as proper extracted trajectories from dataset for training, better model to learn the trajectories. We can try more object detectors to improve the result of our first stage, as we seen in our results, proper detection of players coordinates plays a huge part in the final predicted trajectories. We can try more tracking algorithms; from our experiments we know that better tracking also improves the accuracy of trajectory prediction, because improved tracking provides better information of the player movements, giving us better trajectory of training the model. We can also try to implement different type of tracker, we implemented class-based object tracker, instead of that we can try to implement a instance based object tracker.

We can also collect more datasets from NBA matches. More datasets are generally good for training, but the disadvantage is that it takes too long to train a model, this will be solved by future improvement in the hardware we use to run our program. Finally, we are able to implement a model to capture the player individual game habits, which can improve the prediction accuracy by a huge margin. Also, the development in a better object detection and tracking algorithm will help in getting more accurate data for the training and testing of the trajectory prediction module.

REFERENCES

[1] Rohan Chandra, Tianrui Guan, Srujan Panuganti, Trisha Mittal, Uttaran Bhattacharya, Aniket Bera, Dinesh Manocha: Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs. IEEE Robotics Autom. Lett. 5(3): 4882-4890 (2020)

[2] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, Nanning Zheng: SR-LSTM: State Refinement for LSTM Towards Pedestrian Trajectory Prediction. CVPR 2019: 12085-12094

[3] Ruijie Quan, Linchao Zhu, Yu Wu, Yi Yang: Holistic LSTM for Pedestrian Trajectory Prediction. IEEE Trans. Image Process. 30: 3229-3239 (2021)

[4] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Fei-Fei Li, Silvio Savarese: Social LSTM: Human Trajectory Prediction in Crowded Spaces. CVPR 2016: 961-971

[5] Tara N. Sainath, Oriol Vinyals, Andrew W. Senior, Hasim Sak: Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. ICASSP 2015: 4580-4584

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: Attention Is All You Need. CoRR abs/1706.03762 (2017)

[7] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, Li Fei-Fei: Peeking into the Future: Predicting Future Person Activities and Locations in Videos. CVPR 2019: 5725-5734

[8] Junwei Liang, Lu Jiang, Kevin P. Murphy, Ting Yu, Alexander G. Hauptmann: The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction. CVPR 2020: 10505-10515

[9] Javad Amirian, Jean-Bernard Hayet, Julien Pettré: Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs. CVPR Workshops 2019: 2964-2972

[10] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, Dinesh Manocha: TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions. CVPR 2019: 8483-8492

[11] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, Christian G. Claudel: Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. CVPR 2020: 14412-14420

[12] Parth Kothari, Sven Kreiss, Alexandre Alahi: Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. CoRR abs/2007.03639 (2020)

[13] Junwei Liang, Lu Jiang, Alexander G. Hauptmann: SimAug: Learning Robust Representations from 3D Simulation for Pedestrian Trajectory Prediction in Unseen Cameras. CoRR abs/2004.02022 (2020)

[14] O. Styles, T. Guha and V. Sanchez, "Multiple Object Forecasting: Predicting Future Object Locations in Diverse Environments," *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass, CO, USA, 2020, pp. 679-688, doi: 10.1109/WACV45572.2020.9093446.

[15] Vincent Le Guen, Nicolas Thome: Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models. CoRR abs/1909.09020 (2019)

[16] Sebastián Gómez-González, Sergey Prokudin, Bernhard Schölkopf, Jan Peters: Real Time Trajectory Prediction Using Deep Conditional Generative Models. IEEE Robotics Autom. Lett. 5(2): 970-976 (2020)

[17] Olly Styles, Tanaya Guha, Victor Sanchez, Alex C. Kot: Multi-Camera Trajectory Forecasting: Pedestrian Trajectory Prediction in a Network of Cameras. CVPR Workshops 2020: 4379-4382

[18] Michael Diodato, Yu Li, Antonia Lovjer, Minsu Yeom, Albert Song, Yiyang Zeng, Abhay Khosla, Benedikt D. Schifferer, Manik Goyal, Iddo Drori: Accurate Trajectory Prediction for Autonomous Vehicles. CoRR abs/1911.08568 (2019)

[19] Francesco Giuliari, Irtiza Hasan, Marco Cristani, Fabio Galasso: Transformer Networks for Trajectory Forecasting. CoRR abs/2003.08111 (2020)

[20] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR abs/2004.10934 (2020)

[21] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection. CVPR 2016: 779-788

[22] Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39(6): 1137-1149 (2017)

[23] S. Yadav and S. Payandeh, "Understanding Tracking Methodology of Kernelized Correlation Filter," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2018, pp. 1330-1336, doi: 10.1109/IEMCON.2018.8614990.

[24] Dang, Linh & Nguyen, Gia & Cao, Thang. (2020). Object Tracking Using Improved Deep_Sort_YOLOv3 Architecture. ICIC Express Letters. 14. 961-969. 10.24507/icicel.14.10.961.

[25] Adrian M. P. Brasoveanu, Razvan Andonie: Visualizing Transformers for NLP: A Brief Survey. IV 2020: 270-279