Player tracking and identification in broadcast ice hockey video

by

Kanav Vats

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. James J. Little Professor Emeritus, Dept. of Computer Science University of British Columbia
Supervisor:	Dr. David A. Clausi Professor, Systems Design Engineering University of Waterloo
Supervisor:	Dr. John S. Zelek Associate Professor, Systems Design Engineering University of Waterloo
Internal Member:	Dr. Alexander Wong Professor, Systems Design Engineering University of Waterloo
Internal Member:	Dr. Bryan Tripp Associate Professor, Systems Design Engineering University of Waterloo
Internal-External Member:	Dr. Mark Crowley Associate Professor, Electrical & Computer Engineering University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis contains material from four papers for which I was the lead author. As the lead author, I was responsible for the conceptualization, writing code as well as writing and submitting manuscripts.

1. **K. Vats**, M. Fani, D. A. Clausi, J. S. Zelek, "Multi-task learning for jersey number recognition in Ice Hockey", *In Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports, 2021*

2. **K. Vats**, W. McNally, P. Walters, D. A. Clausi, J. S. Zelek, "Ice hockey player identification via transformers and weakly supervised learning", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022*

3. **K. Vats**, M. Fani, D. A. Clausi, J. S. Zelek, "Evaluating deep tracking models for player tracking in broadcast ice hockey video", *In Proceedings of the Linköping Hockey Analytics Conference Research Track (LINHAC), 2022*

4. **K. Vats**, P. Walters, M. Fani, D. A. Clausi, J. S. Zelek, "Player Tracking and Identification in Ice Hockey", *Submitted to Expert Systems with Applications, Elsevier*

Additionally, I also authored/co-authored several research papers broadly in the areas of computer vision based sports analytics and human pose estimation.

5. Z. Cai, H. Neher, **K. Vats**, D. A Clausi, J. S. Zelek, "Temporal hockey action recognition via pose and optical flows", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019*

6. W. McNally, **K. Vats**, T. Pinto, C. Dulhanty, J. McPhee, A. Wong, " GolfDB: A Video Database for Golf Swing Sequencing", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019*

7. **K. Vats**, H. Neher, A. Wong, D. A. Clausi, J. S. Zelek, "KPTransfer: improved performance and faster convergence from keypoint subset-wise domain transfer in human pose estimation", *International Conference on Image Analysis and Recognition (ICIAR), 2019*

8. M. Fani, **K. Vats**, C. Dulhanty, D. A. Clausi, J. S. Zelek, "Pose-projected action recognition hourglass network (parhn) in soccer", *In 16th Conference on Computer and Robot Vision (CRV), 2019*

9. **K. Vats**, H. Neher, D. A. Clausi, J. S. Zelek, "Two-stream action recognition in ice hockey using player pose sequences and optical flows", *In 16th Conference on Computer and Robot Vision (CRV), 2019*

10. **K. Vats**, W. McNally, C. Dulhanty, Z. Q. Lin, D. A. Clausi, J. S. Zelek, "Pucknet: Estimating hockey puck location from broadcast video", *In AAAI-20 Workshop on Artifical Intelligence in Team Sports, AAAI 2020*

11. **K. Vats**, M. Fani, P. Walters, D. A. Clausi, J. S. Zelek, "Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020*

12. W. McNally, P. Walters, **K. Vats**, A. Wong, J. McPhee, "DeepDarts: Modeling keypoints as objects for automatic scorekeeping in darts using a single camera", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021*

13. **K. Vats**, M. Fani, D. A. Clausi, J. S. Zelek, "Puck localization and multi-task event recognition in broadcast hockey videos", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021*

14. W. McNally, **K. Vats**, A. Wong, J. McPhee, "EvoPose2D: Pushing the boundaries of 2d human pose estimation using accelerated neuroevolution with weight transfer", *IEEE Access Journal, 2021*

15. W. McNally, **K. Vats**, A. Wong, J. McPhee, "Rethinking Keypoint Representations: Modeling Keypoints and Poses as Objects for Multi-Person Human Pose Estimation", *Submitted to ECCV 2022. arXiv preprint arXiv:2111.08557*

Abstract

Tracking and identifying players is a fundamental step in computer vision-based ice hockey analytics. The data generated by tracking is used in many other downstream tasks, such as game event detection and game strategy analysis. Player tracking and identification is a challenging problem since the motion of players in hockey is fast-paced and non-linear when compared to pedestrians. There is also significant player-player and player-board occlusion, camera panning and zooming in hockey broadcast video. Identifying players in ice hockey is a difficult task since the players of the same team appear almost identical, with the jersey number the only consistent discriminating factor between players.

In this thesis, an automated system to track and identify players in broadcast NHL hockey videos is introduced. The system is composed of player tracking, team identification and player identification models. In addition, the game roster and player shift data is incorporated to further increase the accuracy of player identification in the overall system. Due to the absence of publicly available datasets, new datasets for player tracking, team identification and player identification in ice-hockey are also introduced.

Remarking that there is a lack of publicly available research for tracking ice hockey players making use of recent advancements in deep learning, we test five state-of-the-art tracking algorithms on an ice-hockey dataset and analyze the performance and failure cases.

We introduce a multi-task loss based network to identify player jersey numbers from static images. The network uses multi-task learning to simultaneously predict and learn from two different representations of a player jersey number. Through various experiments and ablation studies it was demonstrated that the multi-task learning based network performed better than the constituent single-task settings.

We incorporate the temporal dimension into account for jersey number identification by inferring jersey number from sequences of player images - called player tracklets. To do so, we tested two popular deep temporal networks (1) Temporal 1D convolutional neural network (CNN) and (2) Transformer network. The network trained using the multi-task loss served as a backbone for these two networks. In addition, we also introduce a weakly-supervised learning strategy to improve training speed and convergence for the transformer network. Experimental results demonstrate that the proposed networks outperform the state-of-the art.

Finally, we describe in detail how the player tracking and identification models are put together to form the holistic pipeline starting from raw broadcast NHL video to obtain uniquely identified player tracklets. The process of incorporating the game roster and player shifts to improve player identification is explained. An overall accuracy of 88% is

obtained on the test set. An off-the-shelf automatic homography registration model and a puck localization model are also incorporated into the pipeline to obtain the tracks of both player and puck on the ice rink.

Acknowledgements

First, I would like to thank my supervisors Prof. David A. Clausi and Prof. John S. Zelek for their immense support and encouragement without which this thesis would not have been possible. Thank you for being wonderful mentors and being a source of inspiration.

I would like thank Prof. James J. Little for taking out time from his busy schedule to become my external committee member. I would also like to thank my committee members Prof. Alexander Wong, Prof. Bryan Tripp and Prof. Mark Crowley for their time and constructive feedback during my comprehensive exam. I would also like to thank Stathletes Inc, for providing me with necessary resources to conduct this research.

A huge thanks to the members and alumni of Vision and Image Processing research group, especially my collaborators Mehrnaz Fani, William McNally and Pascale Walters for fruitful collaborations and discussions.

I would like to thank my family for providing me with love and support. My parents Minoo Vats and Rajesh Vats for being my backbone and pillar of strength. My aunt, Punam Sharma for never making me feel homesick in Canada.

Lastly, I would like to thank my friends outside the lab for their help and support in my life, be it work or personal.

Dedication

This thesis is dedicated to my parents.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ice hockey is played by an estimated 1.8 million people worldwide [3]. As a team sport, the positioning of the players and puck on the ice are critical to team offensive and defensive strategy [4]. The location of players and puck on the ice is essential for hockey analysts for determining the location of play and analyzing game strategy and events. The data generated by player tracking is used in many other downstream computer vision tasks, such as game event detection [5] and game strategy analysis [6].

At the time of writing this thesis, the NHL has introduced puck and player tracking technology in 2021 season [1]. The technology includes sensors fitted in pucks and player jerseys. Also, 14-18 infrared cameras have been installed in NHL arenas to detect the infrared signals emitted by sensors embedded in the puck and player jerseys. The infrared signals are triangulated to track players and puck in three dimensions. This technology is bound to come at significant costs and will not be viable for junior leagues or other major international leagues.

Although player tracking data can be obtained manually, the process of labelling data by hand on a per-game basis is extremely tedious and time consuming. Automated tracking of player and puck using computer vision and deep learning can be significantly cheaper without any additional equipment or sensors required. Therefore, an automated computer vision-based player tracking and identification system is of high utility.

[1] www.sporttechie.com/nhl-starts-2021-season-with-puck-and-player-tracking-in-all-arenas/

1.1 Problem Statement

The problem of interest in this thesis is to automate the tracking of players in broadcast video while also identifying the players in each video frame.

Let $\mathscr{F} = \{f_1, f_2, ...f_n\}$ denote a hockey broadcast video of n frames such that f_i denotes the i-th frame of the video. Let $\mathscr{O} = \{o_1, o_2, ...o_k\}$ denote the player observations in $\mathscr{F}$. Each observation $o_j = \{x_j, y_j, w_j, h_j\}$ is a bounding box at location (x_j, y_j) of width and height w_j, h_j in some frame f_i obtained by a person detector. Let the number of players in the video be k.

The problem of tracking consists of forming player trajectories $\mathcal{T}_l = \{o_{l_1}, o_{l_2}, ...o_{l_n}\}$ (also called player tracklets) , where $\mathcal{T}_l$ is an ordered sequence of observations $\{o_{l_i}\}$. The set of all player trajectories of k players is denoted by $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2....\mathcal{T}_m\}$.

The problem of player identification consists of assigning k unique identities $\mathcal{I} = \{I_1, I_2, ...I_k\} : k \leq m$ to each of the m player trajectories $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2....\mathcal{T}_m\}$. Note that the number of unique identities k may be less than the number of trajectories m. This means that more than one trajectory may belong to the same player.

1.2 Challenges

The challenges associated with player tracking and identification in ice hockey are:

1. The motion of players in hockey is fast paced as compared to the setting of pedestrian tracking. There is also significant camera panning, especially when the players move from one offensive zone to the other. Player-player occlusion and player-board occlusion is also a significant issue in hockey player tracking.

2. Compared to other sports such as soccer and basketball, uniquely identifying players is much more challenging in ice hockey due to the players wearing bulky equipment and helmets that occlude body characteristics and skin color, especially reducing the discriminability between players on the same team that wear the same color uniforms and helmets.

3. Conducting research on player tracking and identification in ice hockey is challenging since there are no publicly available datasets for player tracking team identification, and player identification.

1.3 Contributions

Following are the contributions of this thesis:

1. A holistic system that combines player tracking, puck tracking, team identification, and player identification to track players and puck in broadcast ice hockey videos is established. Considering that there can be only between 3 and 5 players on the ice for each team at any point in the game (plus one goalie per team), the system only considers the players present on the ice rink for establishing player identities. Taking into account only players present on the ice rink for identification is shown to significantly improve identification accuracy.

2. Although commercial solutions for hockey player tracking exist [7], to the best of our knowledge, no network architectures used, training data or performance metrics are publicly reported. There is currently no published work for hockey player tracking making use of the recent advancements in deep learning while also reporting the current accuracy metrics used in literature. Therefore we compare and contrast several state-of-the-art tracking algorithms and analyze their performance and failure modes in ice hockey.

3. We design a multi-task loss function for jersey number recognition consisting of the combination of (1) "Holistic" representation loss term treating the jersey number as a separate class (2) "Digit-wise" representation loss term treating digits in a number as independent classes. We conduct an ablation study to demonstrate that the holistic and digit-wise losses complement each other with appropriate weight given to them.

4. A transformer based player identification approach is implemented that infers jersey number from player tracklets. The model utilizes novel weakly supervised training using approximate labels for faster convergence. Experimental results demonstrate the effectiveness of the model by obtaining state-of-the-art results compared to other approaches.

5. New ice hockey datasets are introduced for player tracking, team identification, player identification from static images and player identification from tracklets. The dataset created for player identification from static images consisting of $50,000$ images is the biggest dataset of such kind in the literature.

1.4 Thesis structure

This thesis is organised into eight chapters and is structured as follows:

- Chapter 2 presents a literature review of the applications of computer vision in sports analytics focusing on the problems of player tracking and identification.
- Chapter 3 compares and contrasts several state-of-the-art tracking algorithms and analyzes their performance and failure modes in ice hockey.
- Chapter 4 introduces the multi-task loss based network to perform jersey number identification from static images.
- Chapter 5 discusses the methods developed to directly infer jersey numbers from player tracklets taking into account the temporal information present in video.
- Chapter 6 presents the holistic system that combines player tracking, puck tracking, team identification and player identification models while also utilizing team roster and shift data to track and identify players in broadcast NHL video clips.
- Chapter 7 concludes the thesis providing a summary of the contributions and also discusses the thesis limitations and potential for future research.

Chapter 2

Background and related work

2.1 Multi-object tracking

The objective of multi-object tracking (MOT) is to detect objects of interest in video frames and associate the detections with appropriate trajectories while maintaining their identities. MOT has a variety of applications ranging from surveillance to sports player tracking. MOT is a big challenge in computer vision, especially in crowded scenes due to the enormous number of degrees of freedom for possible trajectories present and the presence of occlusions.

2.1.1 Tracking by detection (TBD)

Tracking by detection (TBD) is a widely used approach for multi-object tracking. Tracking by detection consists of three steps: (1) detecting objects (hockey players in our case) frame-by-frame in the video (2) affinity calculation between detected objects and (3) inference - linking player detections to produce tracks.

Object detection

In tracking by detection MOT algorithms, the first task is to detect the object in each frame, agnostic of their actual identity. With the advent of deep learning, there has been significant progress in deep learning based object detection methods[8, 9, 10, 11]. The Faster RCNN [12] is one of the most popular object detection algorithm and is employed

by many MOT algorithms [13, 14]. The Faster RCNN consists of a region-proposal convolutional neural network (RPN) to calculate high quality object proposals. The RPN object bounding box proposals are then refined and classified into appropriate object classes. Other object detection algorithms such as Single Shot MultiBox Detector (SSD) [11] and You Only Look Once (YOLO) [10] do not incorporate RPNs, but directly regress bounding box coordinates and then does classification. RPN based object detection methods generally perform better than direct regression/classification methods [15].

Affinity Calculation

Affinity calculation is the most crucial phase of any detection based tracking algorithm. It consists of two steps (1) obtaining appropriate features from detections and existing tracks (2) calculating affinity between new detections and existing tracks using the features obtained. Traditionally, handcrafted features such as histogram of oriented gradients (HOG) [16] and color-based features [17] were employed for feature extraction. Recently, deep neural networks have been extensively used for feature extraction for affinity calculations. These networks include convolutional neural networks (CNNs) [18, 19, 20] and recurrent neural networks (RNNs) [21, 22, 23]. New techniques such as tracklet inpainting through stochastic motion modelling [24] have also been introduced to calculate affinities between existing tracks and detections.

Inference

The object detection and affinity calculation methods are plugged into an appropriate inference method that requires object detections and affinity scores to generate final trajectories. This step is also called data association. The inference process in detection based tracking is mostly dominated by techniques such as graph based inference [1, 25, 26, 27, 28] and probabilistic filtering [13, 14, 20, 29]. Many works in literature infer final trajectories with the help of graph-based inference wherein the detections are represented as nodes of a graph and the trajectories are represented by sequences/clusters of edges. Appropriate optimization algorithms such as minimum cost flow [1, 25, 26], lifted multicut[27, 28] or minimum cliques [30] are then employed for finding the optimal graph connectivity representing distinct tracks for a trajectory. Recently, Braso *et al.* [1] formulate the MOT problem into as a simplified min cost flow [26] and introduce a novel message passing network for classifying whether the graph edge belongs to an actual target trajectory. Kalman filter [31] and particle filter [32] are two of the most widely used filtering algorithms for

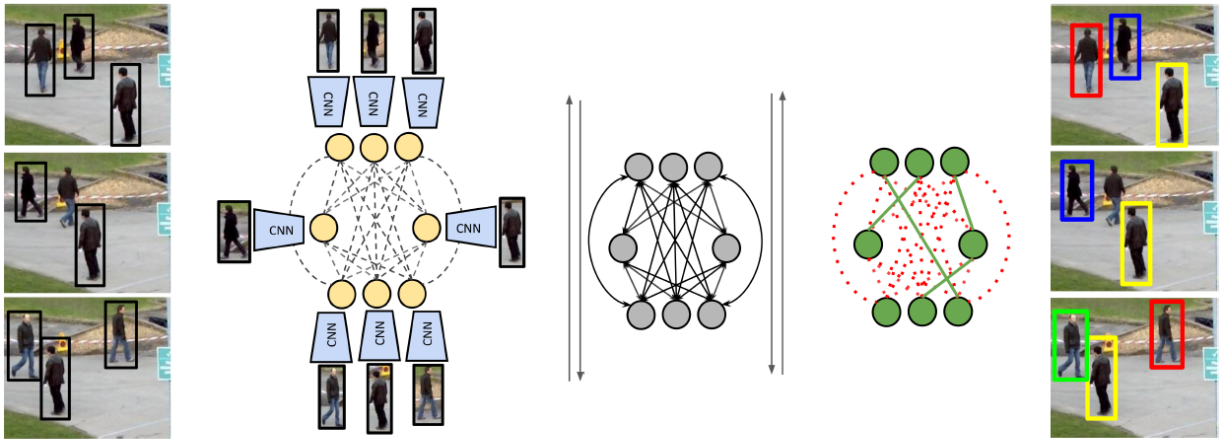


Figure 2.1: Graph based inference proposed by Braso *et al.* [1] method where (a) Video frames and detections are used as input (b) Each detection is considered a node of a graph with all detection nodes connected with graph edges. The input embedding of each node is obtained by a CNN (c) Graph message passing is performed to determine which edges belong to actual trajectories (d) Classification result where green edges correspond of actual trajectory with (e) as the tracking output.

tracking. Bayesian filtering is highly suited for online applications since it requires only the past and present observations for inference.

2.1.2 Joint detection and tracking (JDT)

The latest trend in multi-object tracking research is the paradigm of joint detection and tracking (JDT) [19, 33, 34, 35, 36]. These methods convert an object detector to a tracker by estimating the location of a bounding box in the adjacent frames. Bergmann *et al.* [19] use the bounding box regressor of a Faster RCNN [12] to regress the position of a person in the next frame. The reidentification is performed using a separate siamese network. Wang *et al.* [34] introduce a single network for single-shot object detection and appearance embedding learning trained using a multi-task loss function. Zhang *et al.* [35] adopt an anchor free tracking approach by augmenting the CenterNet [37] object detection model with a re-identification branch. Peng *et al.* [33] introduce chained tracker that combines object detection, feature extraction (affinity calculation) and inference into a single step of

regressing bounding boxes of a target in two adjacent frames. Joint detection and tracking methods enjoy high inference speeds, especially when lightweight and fast detectors are used [34]. This is because the detection and appearance affinity calculation are combined into a single stage step. However, these methods struggle to preserve tracking identities since they are overly dependent on the performance of the detector used and also because they only use temporally local information to perform data association.

2.2 Sports player tracking

Player tracking is an important problem in computer vision-based sports analytics, since player tracking combined with an automatic homography estimation system [38] is used to obtain absolute player locations on the sports rink. Also, various downstream computer vision and machine learning based tasks, such as sports event detection [5, 39, 40] and game strategy analysis [6] can be improved with player tracking data.

2.2.1 Soccer and basketball

For basketball player tracking, Sangüesa *et al.* [41] demonstrated that deep features perform better than classical handcrafted features for basketball player tracking. Lu *et al.* [42] perform player tracking in basketball using a Kalman filter by making the assumption that the relationship between time and player's locations is approximately linear in a short time interval. Zhang *et al.* [43] perform basketball player tracking in a multi camera setting. Theagarajan *et al.* [44] track players in soccer videos using the DeepSORT algorithm [29] for generating tactical analysis and ball possession statistics . Hurault *et al.* [45] introduce a self-supervised detection algorithm to detect small soccer players and track players in non-broadcast settings using a triplet loss trained re-identification mechanism, with embeddings obtained from the detector itself. Theiner *et al.* [46] present a pipeline to extract player position data on the soccer field from video. The player tracking was performed with the help of CenterTrack [36]. However, the major focus of the work was on detection accuracy rather than tracking and identification. Gadde *et al.* [47] use a weakly supervised transductive approach for player detection in soccer broadcast videos by treating player detection as a domain adaptation problem. The dataset used is generated with the help of the DeepSort algorithm [29].

2.2.2 Ice hockey

In ice hockey, prior published research [48, 49] perform player tracking with the help of handcrafted features for player detection and re-identification. Okuma *et al.* [48] track hockey players by introducing a particle filter combined with mixture particle filter (MPF) framework [50], along with an Adaboost [51] player detector. The MPF framework [50] allows the particle filter framework to handle multi-modality by modelling the posterior state distributions of M objects as an M component mixture. A disadvantage of the MPF framework is that the particles merge and split in the process and leads to loss of identities. Moreover, the algorithm did not have any mechanism to prevent identity switches and lost identities of players after occlusions. Cai *et al.* [49] improved upon [48] by using a bipartite matching for associating observations with targets instead of using the mixture particle filter framework. However, the algorithm is not trained or tested on broadcast videos, but performs tracking in the rink coordinate system after a manual homography calculation.

2.3 Player identification

Identifying players and referees is one of the most important problems in computer vision-based sports analytics. Analyzing individual player actions and player performance from broadcast video is not feasible without determining the identities of the tracked players. In the literature, player identification has been performed from static images by using features such as player appearance [2, 42] and jersey number [52, 53, 54, 55]. Prior works also identify players by incorporating temporal context from video [42, 56].

2.3.1 Player identification from static images

In sports such as basketball, body parts such as skin-color, face and legs, not covered by the jersey, play a key role in player re-identification. Senocak *et al.* [2] perform player identification in basketball on a per-frame basis by combining multi-scale CNN features and part-based pose features into a single vector. They conclude that body features such as head, legs, arms etc are discriminative for basketball player identification (Fig 2.2). A shortcoming of the work however, is that the the evaluation set was limited to only five players of Houston Rockets NBA team.

Being one of the most prominent discriminatory feature on the jersey of any sports player, jersey number recognition is a problem of great interest in the computer vision

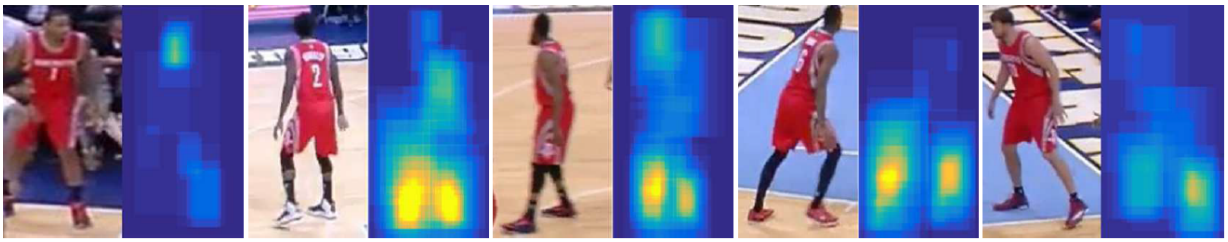


Figure 2.2: Visualization of distinctive parts of players presented in Senocak *et al.* [2]. In sports such as basketball distinctive features for player identification may come from body parts such as foot, legs, head etc.

community. Gerke *et al.* [52] was the first to employ CNNs for soccer jersey number recognition. The CNN outperformed handcrafted HOG features by a huge margin. Gerke *et al.* [57] also merged their image-based jersey number identification system with player location features on the soccer field. Li *et al.* [53] improve upon the method introduced by Gerke *et al.* [52] by introducing a deeper CNN for recognizing jersey number and employing the Spatial Transformer Network for proper alignment of the player so that the jersey number is more readable. However, a limitation of the method is that it requires additional quadrangle annotations for training the Spatial Transformer Network. Liu *et al.* [54] introduce a a network for performing jersey number localization as well as recognition. A Faster RCNN [12] is employed such that the Region Proposal Network (RPN) of the Faster RCNN used three classes to represent the person, digit and background. Further, a pose based supervision is also performed to improve localization of bounding boxes. Nady *et al.* [55] train CRAFT text detection framework [58] to detect jersey numbers and then use a pre-trained text recognition framework [59] to identify jersey numbers from player images.

2.3.2 Player identification from video

Compared to inferring jersey numbers from static images, inferring jersey numbers from player tracklets has been found advantageous [42, 56]. This is because the image sequences provide beneficial temporal information. Lu *et al.* [42] construct a conditional random field (CRF) consisting of feature nodes and identity nodes with appropriate connections and learn the CRF with weakly-supervised learning using a variant of expectation-maximization (EM). Player identification feature vectors are created from handcrafted features such maximally stable extremal regions (MSER) [60], SIFT [61] and color histograms. (Fig

(a) raw image (b) MSER visual words (c) SIFT visual words

Figure 2.3: Handcrafted features such as MSER [60] and SIFT [61] were used for identifying players in basketball by Lu *et al.*[42].

2.3). The algorithm surpassed the results of Okuma *et al.* [48] on the hockey dataset. Lu *et al.* [42] also incorporate play-by-play as a prior during CRF training. However, the dataset used by Lu *et al.* [42] was not very diverse as it consisted of only 2 teams. Chan *et al.* [56] use a network based on the Long-term Recurrent Convolutional Network (LRCN) [62] to infer jersey numbers from player tracklets. The final tracklet scores are aggregated using a secondary CNN. Although Chan *et al.* [56] obtained an accuracy of 87.01% on a private dataset, the requirement of training a secondary network for inference can be seen as a disadvantage.

2.4 Discussion

Previous player tracking methods in hockey [48, 49] use handcrafted methods [51, 60, 61] for person detection and identification. Therefore in Chapter 3, we compare and contrast several state-of-the-art tracking algorithms and analyze their performance and failure modes in ice hockey.

Unlike sports such as basketball and soccer where player body and facial features can be used for identification [2, 42], this is not achievable in ice hockey due to the players wearing bulky equipment and helmets that occlude body characteristics and skin color. For ice hockey, this leaves jersey numbers as the primary method of performing player identification from game video. In Chapter 4 we introduce a network utilizing a novel multi-task loss function for recognizing jersey numbers from static images in ice hockey. Since only utilizing static images leaves out the temporal information present in video data, therefore, in Chapter 5 we introduce networks to identify jersey number from player tracklets that outperform the current state-of-the-art [56].

Although several studies address the problem of player tracking [48, 49] and player

identification [56] in hockey separately, to the best of our knowledge, the two problems have not been combined and studied in a single pipeline. In Chapter 6 we build a novel holistic pipeline composed of player tracking component (Chapter 3), player identification component (Chapter 4 and Chapter 5) and team identification component while also utilizing the player roster and shift data to improve overall identification accuracy. Finally, we augment the pipeline with an off the shelf homography registration model [63] and puck tracking model (Appendix B) to track both players and puck on the ice-rink in broadcast NHL video.

Chapter 3

Player tracking

From Chapter 2 we found that in ice hockey, prior published research [48, 49] perform player tracking with the help of handcrafted features for player detection and re-identification.

In this chapter, we first discuss the new hockey player tracking dataset developed for training and testing tracking models. The dataset statistics along with the annotation software used are explained. We then track and identify hockey players in broadcast NHL videos and analyze performance of five state-of-the-art deep learning based tracking models on the new ice hockey player tracking dataset. We also perform error analysis and identify the major sources of tracking errors.

3.1 Dataset

The player tracking dataset consists of a total of 84 broadcast NHL game clips with a frame rate of 30 frames per second (fps) and resolution of 1280×720 pixels. The average clip duration is 36 seconds. The 84 video clips in the dataset are extracted from 25 NHL games. The duration of the clips is shown in Fig. 3.2. Each video frame in a clip is annotated with player and referee bounding boxes and player identity consisting of player name and jersey number. The annotation is carried out with the help of the open source computer vision annotation tool (CVAT) [1]. An illustration of an annotation job using the CVAT tool is shown in Fig. 3.1. The dataset is split such that 58 clips are used for training, 13 clips for validation, and 13 clips for testing. To prevent any game-level bias affecting the results, the split is made at the game level, such that the training clips are obtained from

[1]Found online at: https://github.com/openvinotoolkit/cvat

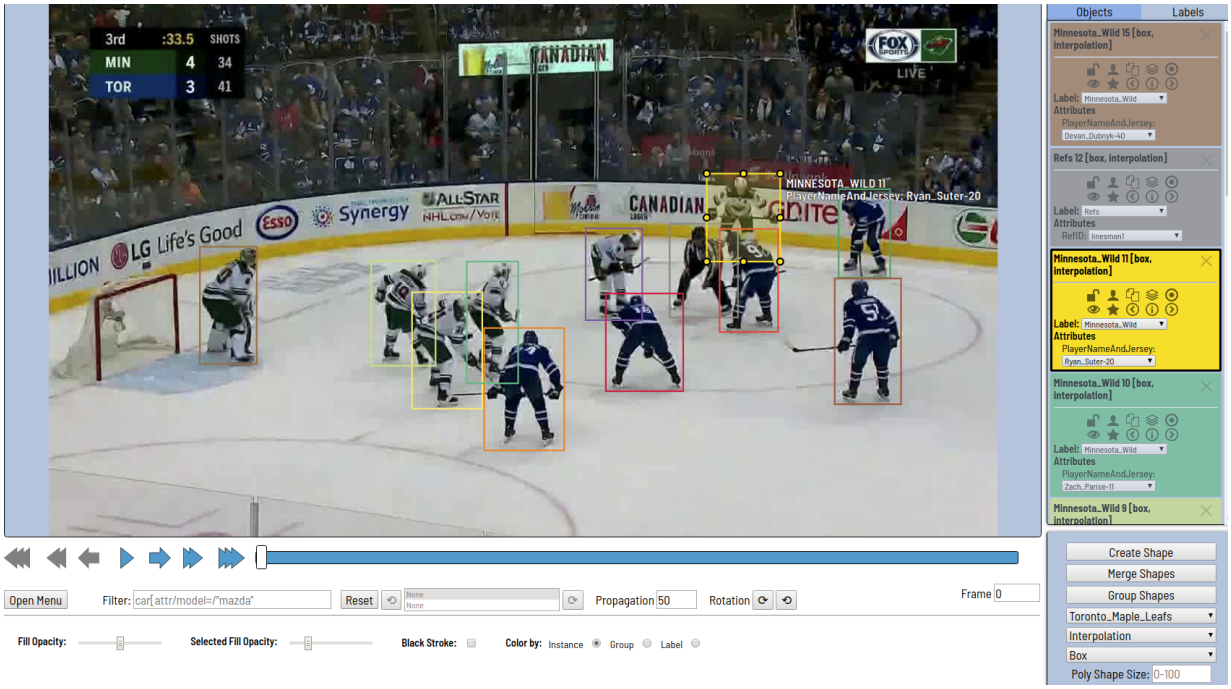


Figure 3.1: CVAT tool used for tracking annotations. The tool offers the ability to annotate bounding boxes with each box having one label - home or away team. Each player bounding box has player name and jersey number as attributes. CVAT also offers an interpolation mode which alleviates the need to draw bounding boxes multiple times for adjacent frames.

17 games, validation clips from 4 games and test split from 4 games respectively.
Table 3.1 compares the size of the dataset with other tracking datasets in literature. The hockey player tracking dataset is comparable in size with other tracking datasets used in literature. As compared to pedestrian datsets (MOT 16 [64] and MOT20 [65]), the bounding boxes per frame is less in our dataset since the maximum number of players on the screen can be 12, with usually less than 12 players actually in broadcast camera field of view (FOV). The NHL game videos used to create this dataset have been obtained from Stathletes Inc. with permission.

Table 3.1: Comparison of hockey tracking dataset with other tracking datasets in literature. Our hockey player tracking dataset is comparable to other multi-object tracking datasets commonly used in literature.

Dataset	Videos/sequences	Frames	Bounding boxes	Domain
MOT16 [64]	14	11,235	292,733	Pedestrians
MOT20 [65]	8	13,410	2,102,385	Crowded pedestrian scenes
KITTI-T [66]	50	10,870	65,213	Autonomous driving
Ours	84	91,807	773,545	Ice hockey players

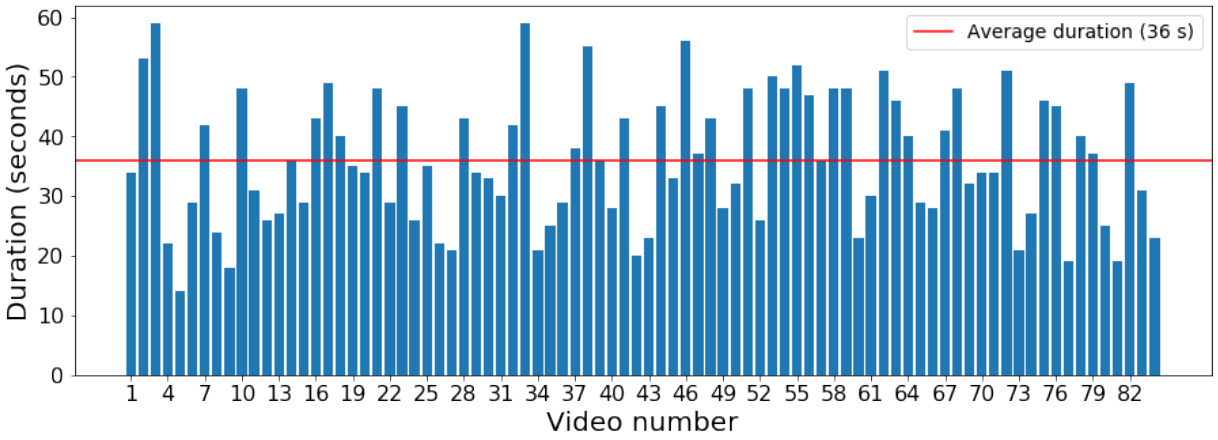


Figure 3.2: Duration of videos in the player tracking dataset. The average clip duration is 36 seconds. The red horizontal line represents the average clip duration.

3.1.1 Annotation process

15 annotators annotated the whole dataset using the CVAT tool. The average time taken to annotate one minute of video is 10.45 minutes. The total time taken to annotate all 84 videos is 527 minutes. The manual annotation was done such that a bounding box as tight as possible was drawn around a player/referee. Linear interpolation was used to interpolate bounding box positions. Additionally, unlike other tracking datasets such as MOT16 [64] and MOT20 [65], the same ground truth identity was assigned to a player leaving a camera FOV at a particular frame and re-entering after some time. If a player was occluded by board or another player, the bounding box was annotated based on the best guess of the tightest box enclosing the full body of the player. For quality control, all bounding boxes were checked to make sure each box has label-name(name of the player).

When a player enters/exits the scene, his bounding box was labeled even if he was partially in camera FOV. Whenever players were occluded by other players, revision of annotations was performed to ensure high quality.

3.2 Methodology

We experimented with five state-of-the-art tracking algorithms [1, 13, 19, 29, 35] on the hockey player tracking dataset. The algorithms include four online tracking algorithms [13, 19, 29, 35] and one offline tracking algorithm [1]. SORT [13], deep SORT [29] and MOT Neural Solver [1] are tracking by detection (TBD) algorithms. Tracktor [19] and FairMOT [35] are joint detection and tracking (JDT) algorithms.

In tracking by detection, the input is a set of object detections $O = \{o_1, ...o_n\}$, where n denotes the total number of detections in all video frames. A detection o_i is represented by $\{x_i, y_i, w_i, h_i, I_i, t_i\}$, where x_i, y_i, w_i, h_i denotes the coordinates, width, and height of the detection bounding box. I_i and t_i represent the image pixels and timestamp corresponding to the detection. The goal is to find a set of trajectories $T = \{T_1, T_2....T_m\}$ that best explains O where each T_i is a time-ordered set of observations. The MOT Neural Solver models the tracking problem as an undirected graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$ is the set of n nodes for n player detections for all video frames. In the edge set E, every pair of detections is connected so that trajectories with missed detections can be recovered. The problem of tracking is now posed as splitting the graph into disconnected components where each component is a trajectory T_i. After computing each node (detection) embedding and edge embedding using a CNN, the model then solves a graph message passing problem. The message passing algorithm classifies whether an edge between two nodes in the graph belongs to the same player trajectory.

The differences and similarities between the five tracking algorithms are summarized in Table 3.2. We refer the readers to the publications of the respective tracking papers [1, 13, 19, 29, 35] for more detail.

3.3 Results

Player detection is performed using a Faster-RCNN network [67] with a ResNet50 based Feature Pyramid Network (FPN) backbone [68] pre-trained on the COCO dataset - a large scale object detection, segmentation, and captioning dataset, popular in computer vision

Table 3.2: Tracking algorithms compared for hockey player tracking.

Algorithm	Description
SORT [13]	Kalman filter with simple IOU based re-id.
Deep SORT [29]	Kalman filter with deep CNN based re-id.
Tracktor [19]	JDT algorithm with separate detection and re-id networks.
FairMOT [35]	JDT algorithm with combined object detection and re-id network.
MOT Neural Solver [1]	Tracking using graph message passing with edge classification.

Table 3.3: Player detection results on the test videos. AP stands for Average Precision. AP_{50} and AP_{75} are the average precision at an Intersection over Union (IoU) of 0.5 and 0.75 respectively.

AP	AP_{50}	AP_{75}
70.2	95.9	87.5

[69] and fine tuned on the hockey tracking dataset. The object detector obtains an average precision (AP) of 70.2 on the test videos (Table 3.3). The accuracy metrics for tracking used are the CLEAR MOT metrics [70] and Identification F1 score (IDF1) [71]. A ground truth object missed by the trackers is called a false negative (FN) whereas a false alarm is called a false positive (FP). For any tracker, a low number of false positives (FP) and false negatives (FN) are favoured. An important metric is the number of identity switches (IDSW), which occurs when a ground truth ID i is assigned a tracked ID j when the last known assignment ID was $k \neq j$. A low number of identity switches is an indicator of accurate tracking performance. For sports player tracking, the IDF1 is considered a better accuracy measure than Multi Object Tracking accuracy (MOTA) since it measures how consistently the identity of a tracked object is preserved with respect to the ground truth identity. The overall results are shown in Table 3.4. The best tracking performance is achieved using the MOT Neural Solver tracking model [1] re-trained on the hockey dataset. The MOT Neural Solver model obtains the highest MOTA score of 94.5 and IDF1 score of 62.9 on the test videos.

Table 3.4: Comparison of the overall tracking performance on test videos of the hockey player tracking dataset. (↓ means lower is better, ↑ mean higher is better)

Method	IDF1↑	MOTA ↑	ID-switches ↓	False positives (FP)↓	False negatives (FN) ↓
SORT [13]	53.7	92.4	673	2403	5826
Deep SORT [29]	59.3	94.2	528	1881	4334
Tracktor [19]	56.5	94.4	687	1706	4216
FairMOT [35]	61.5	91.9	768	1179	7568
MOT Neural Solver [1]	**62.9**	**94.5**	**431**	1653	4394

3.4 Analysis

From Table 3.4 it can be seen that the MOTA score of all methods is above 90%. This is because MOTA is calculated as

$$MOTA = 1 - \frac{\Sigma_t(FN_t + FP_t + IDSW_t)}{\Sigma_t GT_t} \tag{3.1}$$

where t is the frame index and GT is the number of ground truth objects. MOTA metric counts detection errors through the sum $FP+FN$ and association errors through $IDSWs$. Since false positives (FP) and false negatives (FN) heavily rely on the performance of the player detector, the MOTA metric highly depends on the performance of the detector. For hockey player tracking, the player detection accuracy is high because of the sufficiently large size of players in a broadcast video and a reasonable number of players and referees (with a fixed upper limit) to detect in the frame. Therefore, the MOTA score for all methods is high.

The SORT [13] algorithm obtains the least IDF1 score and the highest number of identity switches. This is due to the linear motion model assumption and simple IOU score for re-identification. Deep SORT [18], on the other hand uses features obtained from deep network for re-identification resulting in better IDF1 score and lower identity switches. For JDT based networks, performing detection and re-identification with a single network using a multi-task loss performs better than having separate networks for detection and re-id tasks, evident by better performance of FairMOT [35] compared to Tracktor [19]. JDT tracking algorithms, however, [19, 35] do not not show any significant improvement over deep SORT evident by lower identity switches of deep SORT in comparison. The MOT Neural Solver method achieves the highest IDF1 score of 62.9 and significantly lower identity switches than the other methods. This is because the other trackers use a linear motion model assumption which does not perform well with the motion of hockey players. Sharp changes in player motion often lead to identity switches. The MOT Neural

Table 3.5: Tracking performance of MOT Neural Solver model for the 13 test videos (↓ means lower is better, ↑ means higher is better).

Video #	IDF1↑	MOTA ↑	ID-switches ↓	False positives (FP)↓	False negatives (FN) ↓	Duration (sec.)
1	78.53	94.95	23	100	269	36
2	61.49	93.29	26	48	519	29
3	55.83	95.85	43	197	189	43
4	67.22	95.50	31	77	501	49
5	72.60	91.42	40	222	510	40
6	66.66	90.93	38	301	419	35
7	49.02	94.89	59	125	465	48
8	50.06	92.02	31	267	220	34
9	53.33	96.67	30	48	128	29
10	55.91	95.30	26	65	193	26
11	56.52	96.03	40	31	477	45
12	87.41	94.98	14	141	252	35
13	62.98	94.77	30	31	252	22

Solver model, in contrast, has no such assumptions since it poses tracking as a graph edge classification problem.

Table 3.5 shows the performance of the MOT Neural solver for each of the 13 test videos. We do a failure analysis to determine the cause of identity switches and low IDF1 score in some videos. The major sources of identity switches are severe occlusions and players going out of the camera FOV (due to camera panning and/or player movement). We define a pan-identity switch as an identity switch resulting from a player leaving and re-entering camera FOV due to camera panning. It is very difficult for the tracking model to maintain identity in these situations since players of the same team look identical with features such as, jersey color, helmet model, visor model, stick model, glove model, skate model, tape color etc unidentifiable from bounding boxes cropped from 720p broadcast clips. During a pan-identity switch, a player going out of the camera FOV at a particular point in screen coordinates can re-enter at any other point. We estimate the proportion of pan-identity switches to determine the contribution of panning to total identity switches.

To estimate the number of pan-identity switches, since we have quality annotations, we make the assumption that the ground truth annotations are accurate and there are no missing annotations in the ground truth. Based on this assumption, there is a significant time gap between two consecutive annotated detections of a player only when the player leaves the camera FOV and comes back again. Let $T_{gt} = \{o_1, o_2, ..., o_n\}$ represent a ground truth tracklet, where $o_i = \{x_i, y_i, w_i, h_t, I_i, t_i\}$ represents a ground truth detection. A pan-identity switch is expected to occur during tracking when the difference between timestamps (in frames) of two consecutive ground truth detections i and j is greater than

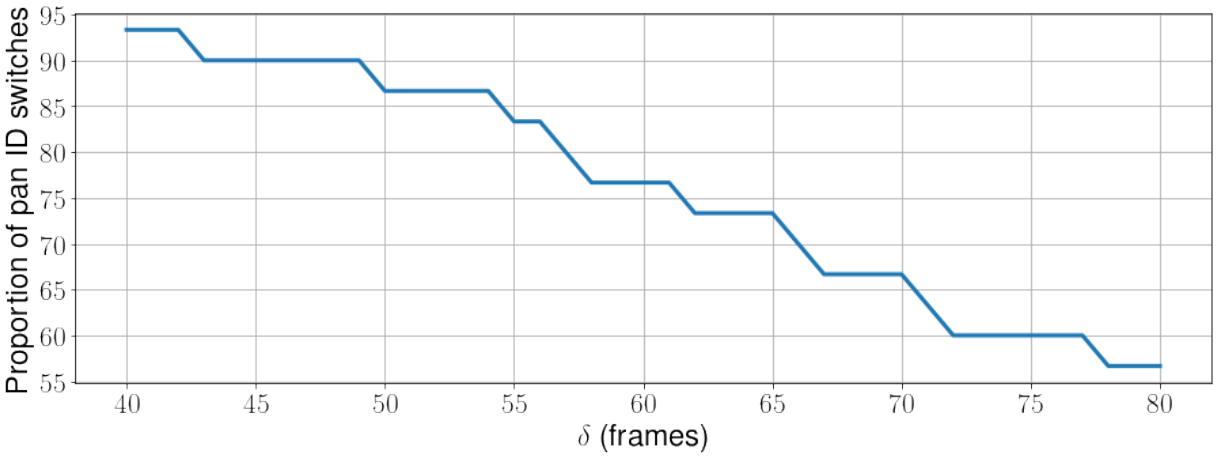


Figure 3.3: Proportion of pan identity switches vs. δ plot for video number 9. Majority of the identity switches (90% at a threshold of $\delta = 40$ frames) occur due to camera panning, which is the main cause of error.

a sufficiently large threshold δ. That is

$$(t_i - t_j) > \delta \tag{3.2}$$

Therefore, the total number of pan-identity switches in a video is approximately calculated as

$$\sum_{G} \mathbb{1}(t_i - t_j > \delta) \tag{3.3}$$

where the summation is carried out over all ground truth trajectories and $\mathbb{1}$ is an indicator function. Consider the video number 9 in Table 3.5 having 30 identity switches and a low IDF1 of 53.33. We plot the proportion of pan identity switches, that is

$$= \frac{\sum_G \mathbb{1}(t_i - t_j > \delta)}{IDSWs} \tag{3.4}$$

against δ, where δ varies between 40 and 80 frames. From Fig. 3.3 it can be seen that majority of the identity switches (90% at a threshold of $\delta = 40$ frames) occur due to camera panning. Visually investigating the video confirmed the statement. Fig. 3.4 shows the proportion of pan-identity switches for all videos at a threshold of $\delta = 40$ frames. On average, pan identity switches account for 65% of identity switches in the videos. This

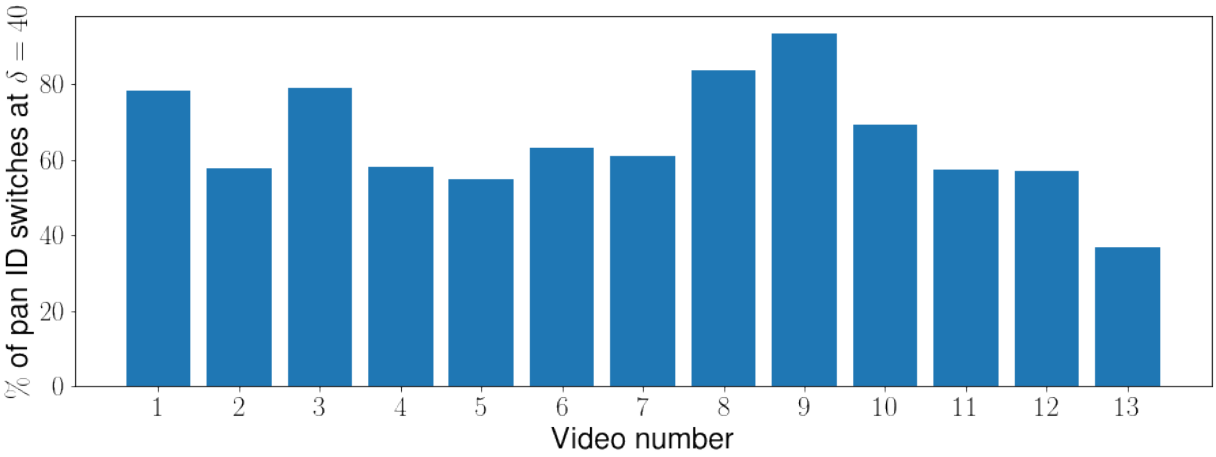


Figure 3.4: Proportion of pan-identity switches for all videos at a threshold of $\delta = 40$ frames. On average, pan-identity switches account for 65% of identity switches.

shows that the tracking model is able to tackle a majority of other sources of errors which include minor occlusions and lack of detections. The primary source or errors are pan-identity switches and extremely cluttered scenes.

3.5 Summary

In this chapter, we test five state-of-the-art tracking algorithms on the ice hockey dataset and analyzed their performance. From the performance of trackers we infer that trackers with a linear motion model do not perform well on the hockey dataset, evident by the high number of identity switches occurring in models with a linear motion assumption. The best performance is obtained by the MOT neural solver model [1], that uses a graph based approach towards tracking without any linear motion model assumption. Also, the IDF1 metric is a better metric for hockey player tracking since the MOTA metric is heavily influenced by player detection accuracy. We find that the main source of error in hockey player tracking in broadcast video are pan-identity switches - identity switches results due to players going outside the broadcast camera FOV.

Chapter 4

Player identification from static images

In the literature, there exist several deep learning approaches for jersey number recognition [52, 53, 54, 57]. These approaches consider jersey number recognition as a classification problem and either (1) consider the jersey numbers as separate classes [52, 57], or (2) treat the two digits in a jersey number as two independent classes [53, 54]. Since learning multiple output representations through multi-task learning can lead to improved regularization [72], in this chapter, we hypothesize that learning both of these representation together in a multi-task loss can result in better performance.

We introduce a network to recognise jersey number for static images. The network utilizes multi-task learning for simultaneously learning the digit-wise and holistic jersey number representations for improving network generalization. Experimental results demonstrate the effectiveness of the multi-task learning formulation by obtaining better performance than the constituent single task settings.

4.1 Methodology

In this section we present the details of the network and the multi-task loss function designed to infer jersey number from static images. We also discuss the experiment settings used to train the network.

4.1.1 Network design

To solve the previously described problem, i.e., players' jersey number recognition in broadcast ice hockey videos, a network with a multi-task loss, as shown in Fig. 4.1, is designed and implemented. The input image of dimension 300×300 pixels is passed through a Resnet34 [73] network to obtain 512-dimensional features from the pre-final layer. The features are then passed through three linear layers followed by softmax layers to output three probabilities. The first linear layer outputs an 81-dimensional vector $p \in \mathbb{R}^{81}$ representing the probability distribution over the 81 jersey number classes. The second and third linear layers output an 11-dimensional vectors $p_1, p_2 \in \mathbb{R}^{11}$ representing the probability of the first and second digit respectively. The one additional class in the 11-dimensional vector denotes the absence of a jersey number. Let $y \in \mathbb{R}^{81}, y_1 \in \mathbb{R}^{11}$ and $y_2 \in \mathbb{R}^{11}$ denote the ground truth vectors corresponding to the jersey number, first digit and second digit respectively.

The multi-task loss consists of three components:

1. The holistic loss $\mathcal{L}$.

$$\mathcal{L} = -\sum_{i=1}^{81} y^i \log p^i \tag{4.1}$$

2. The first digit loss $\mathcal{L}_1$.

$$\mathcal{L}_1 = -\sum_{j=1}^{11} y_1^j \log p_1^j \tag{4.2}$$

3. The second digit loss $\mathcal{L}_2$.

$$\mathcal{L}_2 = -\sum_{k=1}^{11} y_2^k \log p_2^k \tag{4.3}$$

Each of the three losses is a cross-entropy loss between the ground truth and the predicted distribution. The overall loss $\mathcal{L}_{tot}$ is given by

$$\mathcal{L}_{tot} = \alpha * \mathcal{L} + \beta * \mathcal{L}_1 + \gamma * \mathcal{L}_2 \tag{4.4}$$

where α, β, γ denote weights given to each loss such that $\alpha + \beta + \gamma = 1$. Also,

$$\beta * \mathcal{L}_1 + \gamma * \mathcal{L}_2 \tag{4.5}$$

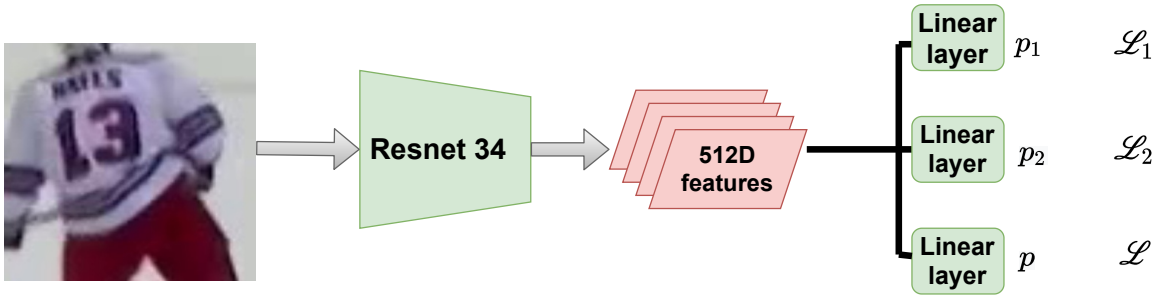


Figure 4.1: The input image is passed through a Resnet 34[73] network after which the 512 dimensional features are extracted from the pre-final layer. $\{p_i, i \in \{1, 2\}\}$ and p are 11 and 81-dimensional vectors representing the digit probabilities and holistic number probabilities respectively. $\mathcal{L}_1$ and $\mathcal{L}_2$ denotes the individual first and second digit loss respectively and $\mathcal{L}$ denotes the holistic loss.

is the overall digit-wise loss and $\beta + \gamma$ is the total weight given to the digit-wise loss.

4.1.2 Training details

For data augmentation, we perform color jittering with high values of the *hue* parameter. Affine transformations are however not performed since they led to a decrease in performance. This is because transformations such as scaling can often make a jersey number not visible since each image has a different scale. The training is done for $10,000$ iterations with an Adam optimizer initial learning rate of .001 and $L2$ weight decay of .001. The learning rate is decreased by a factor of 0.33 after $2000, 4000, 6000$ and 7000 iterations. A batch size 100 is used on a single 1080Ti GPU.

4.2 Experiments

In this section we describe the dataset developed for recognizing jersey numbers from static images. The dataset is compared with other datasets in the literature[52, 54, 74]. The results obtained by testing the network developed in Section 4.1.1 on the dataset are also discussed. Finally we present some ablation studies to explain the impact of parameters such as the backbone network and value of loss weights α, β, γ.

(a) (b) (c) (d)

Figure 4.2: Examples of images from the dataset. The dataset covers many real-game scenarios such as (a) occlusions from external objects, (b)(c) motion blur, and (d) self-occlusion.

4.2.1 Dataset

Datasets used in recent works [52, 53, 54, 57] are not publicly available, hence we created our own dataset. The dataset consists of $54,251$ player bounding boxes obtained from 25 National Hockey League (NHL) games. The NHL game videos are of resolution 1280×720 pixels. The dataset contains a total of 81 jersey number classes, including an additional null class for no jersey number visible. The dataset is much bigger than the datasets used in other works such as Gerke *et al.* [52] with 8,281 images and Liu *et al.* [54] with 3,567 images and 6,293 digit instances (Table 4.1). Although the dataset used in Li *et al.* [53] has $215,036$ images, 90% of the images are negative samples (no jersey number present). Hence, our dataset has more images with a non-null jersey number than Li *et al.* [53].

The player head and bottom of the images are cropped such that only the jersey number is visible. Fig. 4.2 shows some example images from the dataset. A number was considered readable when both constituent digits were visible, however, images with partial occlusion due to motion blur and jersey kinks were included in the dataset since those situations are very common and a model working in sports scenarios should handle those situations. A digit was considered unreadable when either one/both of its constituent digits was fully occluded/invisible. Two annotators annotated the entire dataset.
Images from 17 games are used for training, four games for validation and four games for testing. The exact number of images in the splits is shown in Table 4.2. The splits are constructed at a game level, so that there is no inherent in-game bias present during validation or testing. The dataset is highly imbalanced such that the ratio between the most frequent and least frequent class is 92. The class distribution in the dataset is illustrated in Fig. 4.3. Fig. 4.4 shows the distribution of individual digits in the dataset. The dataset

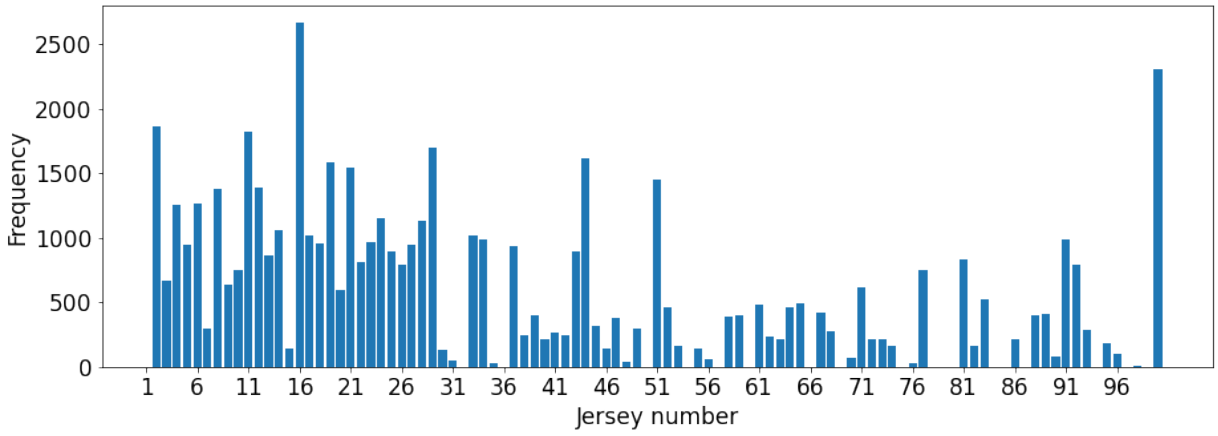


Figure 4.3: Class example distribution in the dataset. The total number of classes is 81 and this includes the "not visible" class. The dataset is highly imbalanced such that the ratio between the most frequent and least frequent class is 92.

Table 4.1: Comparison of datasets in literature

Dataset	Number of images
Gerke *et al.* [52]	8,281
Liu *et al.* [54]	3,567
Ours	**54,251**

Table 4.2: Number of images in train, validation and test set

Train	Validation	Test
38,456	6,770	9,025

covers a range of real-game scenarios such as occlusions, motion blur and self occlusions. We plan on making the dataset publicly available in future.

4.2.2 Results and discussion

We compare the proposed multi-task loss with holistic and digit-wise losses by simply removing the other loss branch from the network. For the digit-wise setting, a predicted number is classified correctly when both of its digits are classified correctly. From Table 4.3,

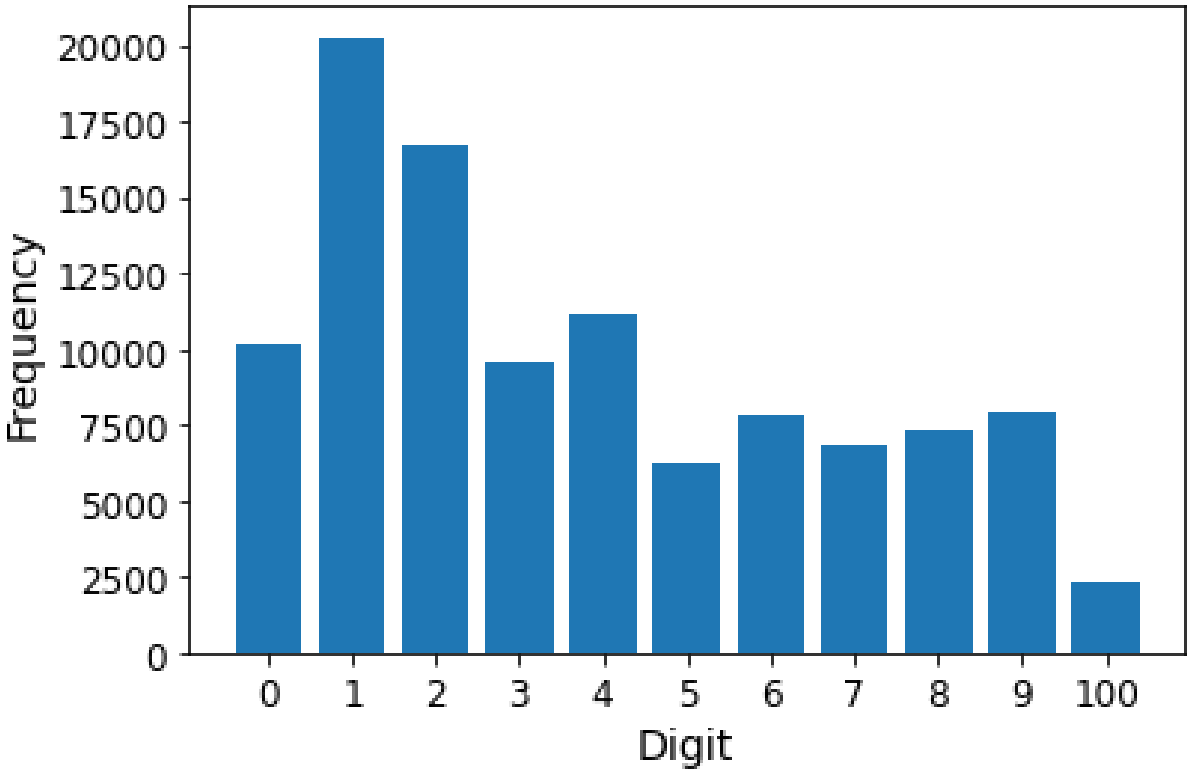


Figure 4.4: Digit distribution in the jersey number recognition dataset. 100 denotes the null class.

the multi-task loss gives an accuracy of 89.6% and a macro averaged F1 score of 91.2% and outperforms the holistic (accuracy 87.6%) and digit-wise losses (accuracy 88.1%). Fig. 4.5 shows the validation accuracy for the three settings during $10,000$ training iterations. The multi-task loss outperforms holistic and digit-wise losses during training.

We implemented the Gerke *et al.* [52] model on our dataset and found the performance low (45.7% test accuracy). We believe that the reasons for this low performance are: (1) The much bigger size of our dataset compared to Gerke *et al.* [52] that lowered the generalizability Gerke *et al.*; and (2) Ice hockey is a more challenging domain for jersey number identification than soccer due to high motion blur from fast moving-camera.

We also implemented the version of Li *et. al.* [53] on our dataset without using spatial transformer localization loss since it requires 'quadrangle' annotations as mentioned in

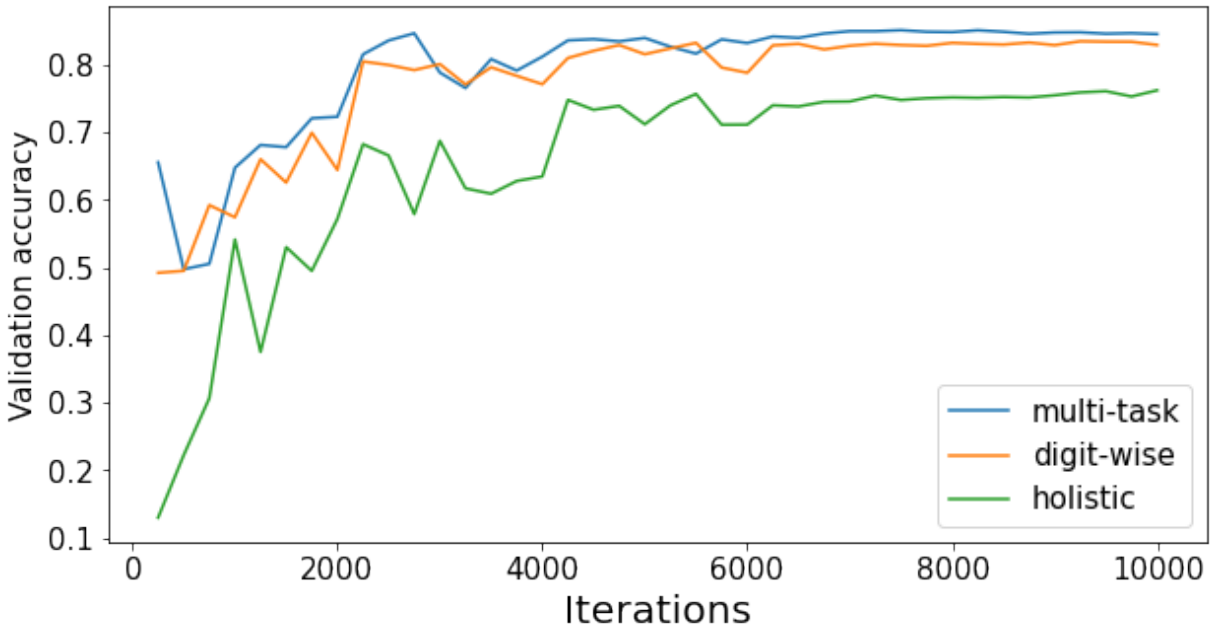


Figure 4.5: Validation accuracy vs number of iterations for the multi-task learning(MTL), holistic and digit-wise loss settings. The multi-task setting shows the best performance among the three settings.

Li *et. al.* [53]. The accuracy obtained was 80.0% with F1 score of 82.5% (Table 4.3). We further replaced the classification cross entropy loss function in Li *et al.* [53] with the proposed loss (Section 4.1.1) function and found an improvement in accuracy of 1.6% (81.6% accuracy) and F1 score of 1.2% (83.7% F1 score) demonstrating the effectiveness of the proposed loss function. We could not compare our model with Liu *et al* [54] since training Liu *et al* [54] model requires digit level bounding boxes and human keypoint annotations which our dataset does not have and there are no trained models provided by the authors to be used publicly for testing.

Fig. 4.6 shows some interesting failure cases. Partial occlusions are common and can result in misinterpretation of jersey numbers (Fig. 4.6 part a). Other sources of failures are folding of the jersey leading to errors (Fig. 4.6 part b), jersey numbers not fully present in player bounding boxes (Fig. 4.6 part c) and jersey number occluded due to camera viewpoints (Fig. 4.6 part d).

Table 4.3: Comparison of accuracy values with holistic, digit-wise and multi-task settings

Method	Test Acc	Precision	Recall	F1 score
Holistic	87.6	90.9	87.7	88.7
digit-wise	88.1	92.5	88.1	89.9
multi-task	**89.6**	**93.6**	**89.6**	**91.2**
Li *et al.* [53]	80.0	87.1	80.0	82.5
Li *et al.* [53](proposed loss)	81.6	87.9	81.6	83.7
Gerke *et al.* [52]	45.7	58.5	45.7	48.2

(a) GT:24; Predicted:21 (b) GT:16; Predicted:18 (c) GT:12; Predicted:2 (d) GT:72; Predicted:77

Figure 4.6: Some common sources of error are (a) occlusions from external sources, (b) folding of jersey, (c) faulty bounding boxes, and (d) camera viewpoints not covering the whole jersey.

4.2.3 Ablation study

We perform an ablation study on the loss weights α, β, and γ to determine how the digit-wise and holistic losses affect accuracy. The analysis can be seen in Table 4.5. We observe that giving a higher weight to the digit-wise loss ($\beta + \gamma = 0.7$) gives the highest accuracy (89.6%) and F1 score (91.2%). However, having a high value of holistic loss weight ($\alpha = 0.8$) results in a lower accuracy(87.8%) and F1 score (89.0%). This makes sense because on its own, the digit-wise loss gives better accuracy compared to holistic loss (Table 4.3). However, as $\beta + \gamma$ is further increased to 0.9 the accuracy decreases (89%). This demonstrates that holistic and digit-wise losses complement each other when an appropriate weight is given to both losses. The accuracy is maximized when the digit-wise loss is given slightly more than double the weight of the holistic loss. The best values are

Table 4.4: Comparison of accuracy values with different backbone networks

Backbone	Test Acc	Precision	Recall	F1 score
Mobilenetv2	87.9	91.8	87.9	89.3
Resnet18	89.1	92.5	89.1	90.3
Resnet34	**89.6**	**93.6**	**89.6**	**91.2**

Table 4.5: Comparison of accuracy values with different values of loss weight coefficients for the multi-task setting

α	β	γ	Test Acc	Precision	Recall	F1 score
1	0	0	87.6	90.9	87.7	88.7
0.8	0.1	0.1	87.8	92.0	87.3	89.0
0.5	0.25	0.25	89.1	92.3	89.1	90.2
0.33	0.33	0.33	88.4	92.7	88.4	90.0
0.3	0.35	0.35	**89.6**	**93.6**	**89.6**	**91.2**
0.2	0.4	0.4	89.6	92.8	89.6	90.9
0.1	0.45	0.45	89.0	92.9	89.07	90.6
0	0.5	0.5	88.1	92.5	88.1	89.9

$\alpha = 0.3, \beta = 0.35$ and $\gamma = 0.35$.

We also do an ablation study on the backbone network used in the experiment in Table 4.4. Two additional backbones were tested: Resnet18 [73], Mobilenetv2 [75], while keeping other parameters including the loss weights α, β, γ fixed to their optimal values of $0.3, 0.35, 0.35$. Resnet 34 showed the best performace followed by Resnet18 and Mobilenetv2. We did not test bigger networks such as Resnet 50 since it could not fit a batch size of 100 on a single GPU.

4.3 Summary

In this chapter, we introduce a simple multi-task learning network for player's jersey number recognition in ice hockey broadcast video frames. We also create a new dataset with more than $50,000$ images to test the network. The network learns both the holistic and

digit-wise representations of jersey number labels which resulted in improved regularization and accuracy. The methodology is however, task agnostic and can be used in other number recognition tasks.

Chapter 5

Player identification from tracklets

In the previous chapter, we inferred player jersey number from static images. However, inferring jersey number from static images does not take into account the valuable temporal information present in sports videos. The temporal information present in sports broadcast can be leveraged to recognize player jersey numbers. Suppose a player with a two-digit jersey number has been tracked using a player tracking model. Often, only one of the digits is visible in the tracklet due to occlusion and varying camera angles. The lack of temporal context poses a challenge for jersey number recognition from static images since a network recognizing jersey number from static images has access to only one image at a time. Therefore, in this chapter, we introduce networks leveraging temporal information by processing multiple tracklet images for jersey number recognition. In order to train and test the networks developed, a tracklet identification dataset is used where each tracklet is manually annotated with a ground truth jersey number. We first introduce a temporal 1D CNN model for tracklet identification in Section 5.1. We thoroughly discuss the training and inference techniques and data augmentations used. In Section 5.2, we introduce a transformer network that improves upon the temporal 1D CNN model and uses weakly-supervised learning for faster training and convergence. Both the temporal 1D CNN and transformer model improve upon the previous state of the art [56] and do not require training of a secondary network for inference [56].

5.1 Temporal CNN model

In this section, we describe the temporal 1D CNN developed for identifying jersey numbers from tracklets. First, we examine the network architecture, training details and the infer-

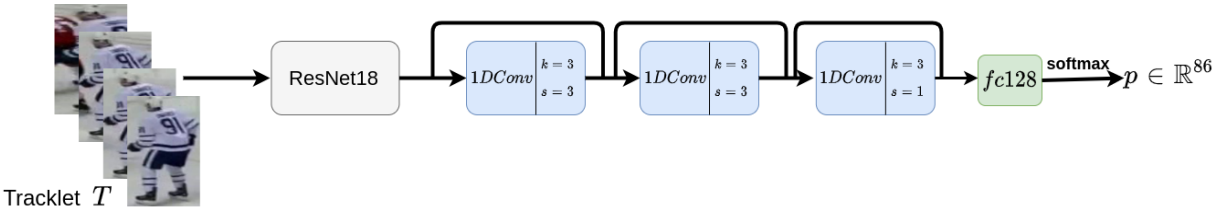


Figure 5.1: Network architecture for the player identification model. The networks accepts a player tracklet as input. Each tracklet image is passed through a ResNet18 to obtain time ordered features F. The features F are input into three 1D convolutional blocks, each consisting of a 1D convolutional layer, batch normalization, and ReLU activation. In this figure, k and s are the kernel size and stride of convolution operation. The activations obtained from the convolutions blocks are mean-pooled and passed through a fully connected layer and a softmax layer to output the probability distribution of jersey number $p \in \mathbb{R}^{86}$.

ence technique used. Then we explain the dataset developed and the experimental results. Finally, ablations studies on the inference techniques and data augmentations used are discussed.

5.1.1 Network architecture

Let $\mathbf{T} = \{o_i\}_{i=1}^{n}$ denote a player tracklet where each o_i represents a player bounding box. The player head and bottom in the bounding box o_i are cropped such that only the jersey number is visible. Each resized image $I_i \in \mathbb{R}^{300\times\ 300\times 3}$ corresponding to the bounding box o_i is input into a backbone 2D CNN, which outputs a set of time-ordered features $\mathbf{F} = \{f_i\}_{i=1}^{n}, f_i \in \mathbb{R}^{512}$. The features $\mathbf{F}$ are input into a 1D temporal convolutional network that outputs probability $p \in \mathbb{R}^{86}$ of the tracklet belonging to a particular jersey number class.

The network consists of a ResNet18 [73] based 2D CNN backbone pretrained on the player identification image dataset described in Chapter 4, Section 4.2.1. The weights of the ResNet18 backbone network are kept frozen while training. The 2D CNN backbone is followed by three 1D convolutional blocks each consisting of a 1D convolutional layer, batch normalization, and $ReLU$ activation. Each block has a kernel size of three and dilation of one. The first two blocks have a larger stride of three, so that the initial layers have a larger receptive field to take advantage of a large temporal context. Residual skip connections are added to aid learning. The exact architecture is shown in Table 5.1.

Finally, the activations obtained are pooled using mean pooling and passed through a fully connected layer with 128 units. The logits obtained are softmaxed to obtain jersey number probabilities. Note that the model accepts fixed length training sequences of length $n = 30$ frames as input, but the training tracklets are hundreds of frames in length (Fig. 5.2). Therefore, $n = 30$ tracklet frames are sampled with a random starting frame from the training tracklet. This serves as a form of data augmentation since at every training iteration, the network processes a randomly sampled set of frames from an input tracklet.

Table 5.1: Network architecture for the temporal 1D player identification model. k, s, d and p denote kernel dimension, stride, dilation size and padding respectively. Ch_i, Ch_o and b denote the number of channels going into and out of a block, and batch size, respectively.

Input: Player tracklet $b \times 30 \times 3 \times 300 \times 300$
ResNet18 backbone
Layer 1: Conv1D $Ch_i = 512, Ch_o = 512$ (k = 3, s = 3, p = 0, d = 1) Batch Norm 1D ReLU
Layer 2: Conv1D $Ch_i = 512, Ch_o = 512$ (k = 3, s = 3, p = 1, d = 1) Batch Norm 1D ReLU
Layer 3: Conv2D $Ch_i = 512, Ch_o = 128$ (k = 3, s = 1, p = 0, d = 1) Batch Norm 1D ReLU
Layer 4: Fully connected $Ch_i = 128, Ch_o = 86$
Output $b \times 86$

5.1.2 Training details

To address the severe class imbalance present in the tracklet dataset, the tracklets are sampled intelligently such that the *null* class is sampled with a probability $p_0 = 0.1$. The network is trained with the help of cross entropy loss. We use Adam optimizer for training with a initial learning rate of .001 with a batch size of 15. The learning rate is reduced by a factor of $\frac{1}{5}$ after iteration numbers 2500, 5000, and 7500. Several data augmentation

techniques such as random cropping, color jittering, and random rotation are also used. All experiments are performed on two Nvidia P-100 GPUs.

5.1.3 Inference

During inference, we need to assign a single jersey number label to a test tracklet of k bounding boxes $\mathbf{T_{test}} = \{o_1, o_2....o_k\}$. Here k can be much greater than $n = 30$. So, a sliding window technique is used where the network is applied to the whole test tracklet $\mathbf{T_{test}}$ with a stride of one frame to obtain window probabilities $\mathbf{P} = \{p_1, p_2, ...p_k\}$ with each $p_i \in \mathbb{R}^{86}$. The probabilities $\mathbf{P}$ are aggregated to assign a single jersey number class to a tracklet. To aggregate the probabilities $\mathbf{P}$, we filter out the tracklets where the jersey number is visible. To do this we first train a ResNet18 classifier C^{im} (same as the backbone of discussed in Section 5.1.1) on the player identification image dataset. The classifier C^{im} is run on every image of the tracklet. A jersey number is assumed to be absent on a tracklet if the probability of the absence of jersey number C^{im}_{null} is greater than a threshold θ for each image in the tracklet. The threshold θ is determined using the player identification validation set. In the tracklets where the jersey number is visible, the probabilities are averaged to obtain a single probability vector p_{jn}, which represents the probability distribution of the jersey number in the test tracklet $\mathbf{T_{test}}$. For post-processing, only those probability vectors p_i are averaged for which $argmax(p_i) \neq null$.

The rationale behind using visibility filtering and post-processing step is that a large tracklet with hundreds of frames may have the number visible in only a few frames and therefore, a simple averaging of probabilities $\mathbf{P}$ will often output *null*. The proposed inference technique allows the network to ignore the window probabilities corresponding to the *null* class if a number is visible in the tracklet. The whole algorithm is illustrated in Algorithm 1.

5.1.4 Tracklet dataset

The player identification tracklet dataset consists of 3510 player tracklets. The tracklet bounding boxes and identities are annotated manually. The manually annotated tracklets simulate the output of a tracking algorithm. The tracklet length distribution is shown in Fig. 5.2. The average length of a player tracklet is 191 frames (6.37 seconds in a 30 frame per second video). It is important to note that the player jersey number is visible in only a subset of tracklet frames. Fig. 5.4 illustrates two tracklet examples from the dataset. The dataset is divided into 86 jersey number classes with one *null* class representing no

Algorithm 1: Algorithm for inference on a tracklet.

Input: Tracklet $\mathbf{T_{test}} = \{o_1, o_2....o_k\}$, image-wise jersey number classifier C^{im}, Tracklet id model $\mathbf{P}$, Jersey number visibility threshold θ
Output: Identity Id, p_{jn}
Initialize: $vis = false$
$P = \mathcal{P}(\mathbf{T_{test}})$ `// using sliding window`
for $o_i \quad in \quad \mathbf{T_{test}}$ **do**
    if $C^{im}_{null}(o_i) < \theta$ **then**
        $vis = true$
        $break$
    end
end
if $vis == true$ **then**
    $\mathbf{P'} = \{p_i \in \mathbf{P} : argmax(p_i) \neq null\}$ `// post-processing`
    $p_{jn} = mean(\mathbf{P'})$
    $Id = argmax(p_{jn})$
end
else
    $Id = null$
end

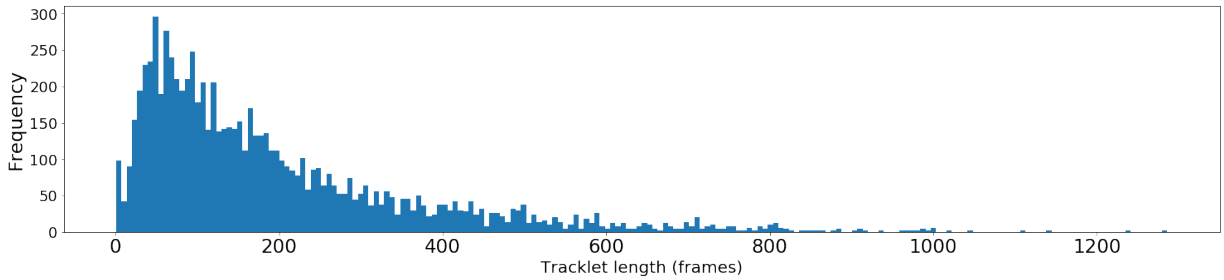


Figure 5.2: Distribution of tracklet lengths (in frames) of the player identification dataset. The distribution is positively skewed with the average length of a player tracklet as 191 frames.

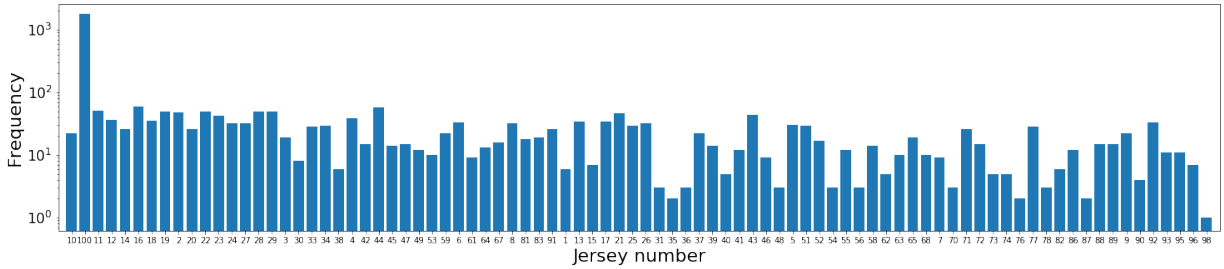


Figure 5.3: Class distribution in the player tracklet identification dataset. The dataset is heavily imbalanced with the *null* class (denoted by class 100) consisting of 50.4% of tracklet examples.

jersey number visible. The class distribution is shown in Fig. 5.3. The dataset is heavily imbalanced with the *null* class consisting of 50.4% of tracklet examples. The training set contains 2829 tracklets, 176 validation tracklets and 505 test tracklets. The game-wise training/testing data split is identical to the tracking dataset in Chapter 3 and the image-based player identification dataset in Chapter 4 such that 17 games are used for training and 8 games are used for validation and testing.

5.1.5 Results

The proposed player identification network attains an accuracy of 83.17% on the test set. We compare the network with Chan *et al.* [56] who use a secondary CNN model for

Figure 5.4: Examples of two tracklets in the player identification dataset. **Top row:** Tracklet represents a case when the jersey number 12 is visible in only a subset of frames. **Bottom row:** Example when the jersey number is never visible over the whole tracklet.

aggregating probabilities on top of an CNN+LSTM model. Our proposed inference scheme, on the contrary, does not require any additional network. Since the code and dataset for Chan *et al.* [56] is not publicly available, we re-implemented the model by scratch and trained and evaluated the model on our dataset. The proposed network performs 9.9% better than Chan *et al.* [56]. The network proposed by Chan *et al.* [56] processes shorter sequences of length 16 during training and testing, and therefore exploits less temporal context than the proposed model with sequence length 30. Also, the secondary CNN used by Chan *et al.* [56] for aggregating tracklet probability scores easily overfits on our dataset evident by a high training accuracy of 98% with low testing accuracy of 73.27%. Adding $L2$ regularization while training the secondary CNN proposed in Chan *et al.* [56] on our dataset also did not improve the performance. This is because our dataset is half the size and is more skewed than the one used in Chan *et al.* [56], with the *null* class consisting of half the examples in our case. The higher accuracy indicates that the proposed network and training methodology involving intelligent sampling of the *null* class and the proposed inference scheme works better on our dataset. Additionally, temporal 1D CNNs have been reported to perform better than LSTMs in handling long range dependencies [76], which is verified by the results.

The network is able to identify digits during motion blur and unusual angles (Fig. 5.6). Upon inspecting the error cases, it is seen that when a two digit jersey number is misclassified, the predicted number and ground truth often share one digit. This phenomenon is observed in 85% of misclassified two digit jersey numbers. For example, 55 is misclassified

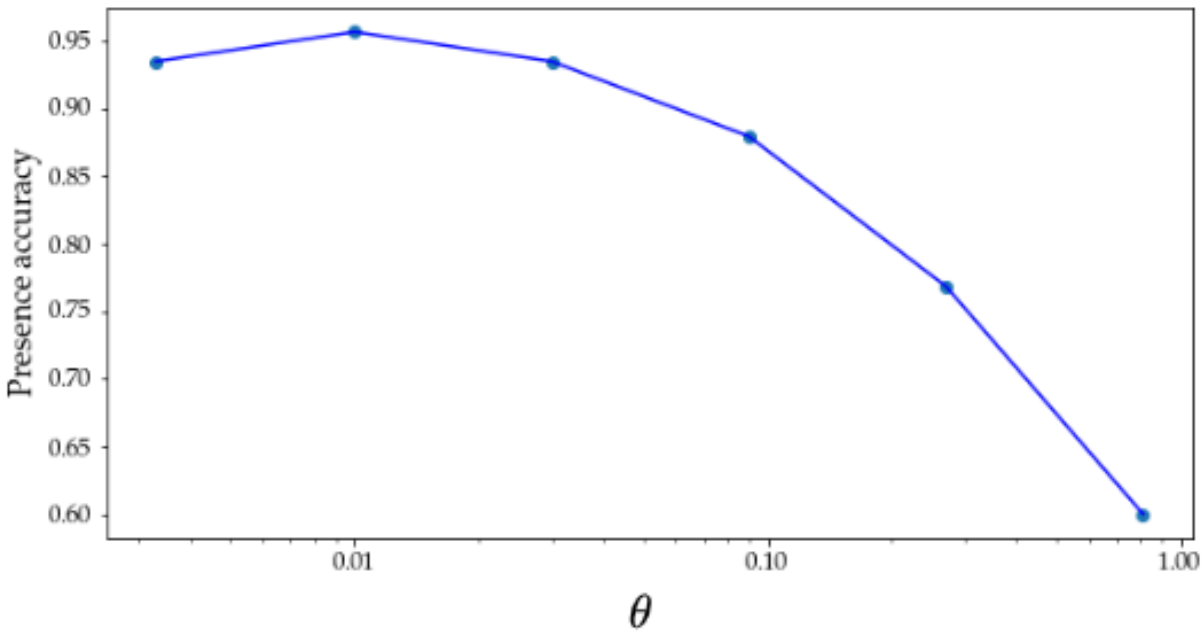


Figure 5.5: Jersey number presence accuracy vs. θ (threshold for filtering out tracklets where jersey number is not visible) on the validation set. The values of θ tested are $\theta = \{0.0033, 0.01, 0.03, 0.09, 0.27, 0.81\}$. The highest accuracy is attained at $\theta = 0.01$.

as 65 and 26 is misclassified as 28 since 6 often looks like 8 (Fig. 5.7) because of occlusions and folds in player jerseys.

The value of θ (threshold for filtering out tracklets where jersey number is not visible) is determined using the validation set. In Fig 5.5, we plot the percentage of validation tracklets correctly classified for the presence of jersey number versus the parameter θ. The values of θ tested are $\theta = \{0.0033, 0.01, 0.03, 0.09, 0.27, 0.81\}$. The highest accuracy of 95.64% at $\theta = 0.01$. A higher value of θ results in more false positives for jersey number presence. A θ lower than 0.01 results in more false negatives. We therefore use the value of $\theta = 0.01$ for doing inference on the test set.

5.1.6 Ablation studies

We perform ablation studies in order to study how backbone pretraining, data augmentation and inference techniques affect the player identification network performance.

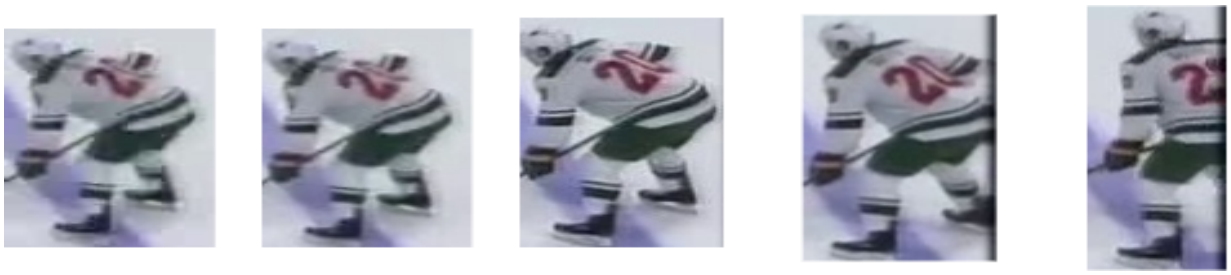


Figure 5.6: Some frames from a tracklet where the model is able to identify the number 20 where 0 is at a tilted angle in majority of bounding boxes.

Figure 5.7: Some frames from a tracklet where 6 appears as 8 due to motion blur and folds in the player jersey leading to error in classification.

Backbone pretraining

In Section 5.1.1, the Resnet18 [73] network used as a backbone for temporal 1D CNN network was pretrained on the player identification image dataset (Section 4.2.1). We perform an ablation study to understand the effect of pretraining by considering two more settings (1) removing the pretrained Resnet18 and replacing it with an identical Resnet18 without any pretraining (random weight initialization) and (2) using Resnet18 pretrained on Imagenet dataset [77] which is a widely used practice in literature. From Fig. 5.8, it can be seen that the player identification model trained using the randomly initialized backbone converges to a low training accuracy of 10%. For the player identification model trained using the backbone pretrained on Imagenet dataset, the learning is very slow (green curve in Fig. 5.8). The fastest learning and convergence is shown by the model using the backbone pretrained on player identification image dataset (Section 4.2.1) as seen by the orange curve in Fig.5.8. This is because the backbone pretrained on player

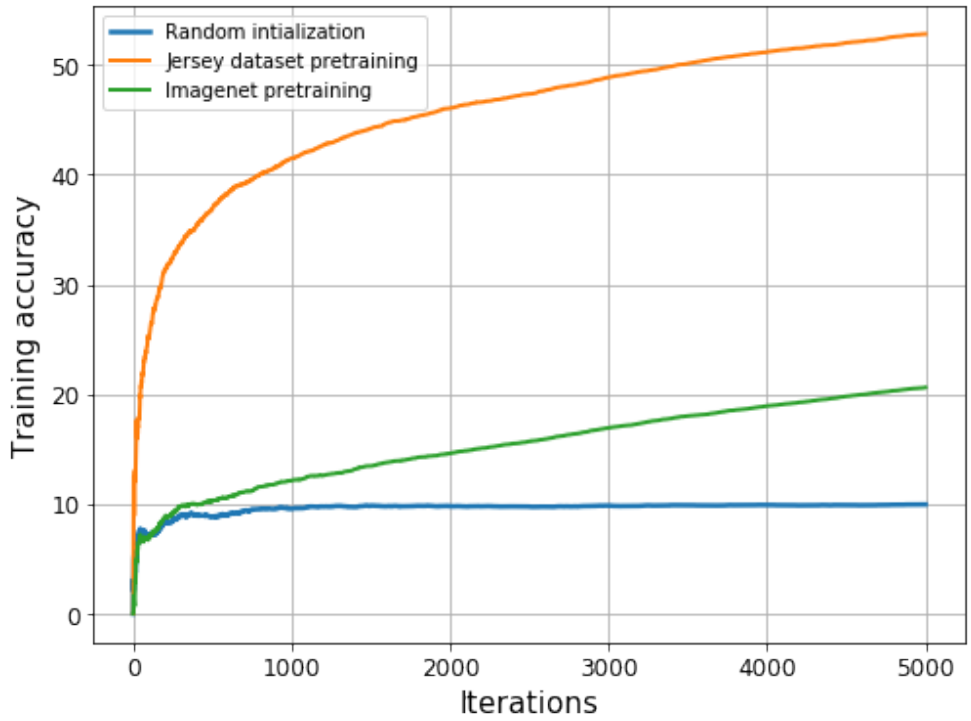


Figure 5.8: Effect of backbone pretraining on the training of player identification network. With a randomly initialized backbone network, the player identification network converges to a low training accuracy of 10%. The fastest training and convergence is obtained by the model using the backbone network pretrained on jersey number dataset.

identification image dataset possesses important domain knowledge of jersey numbers that readily transfers to the task of recognizing jersey numbers from video.

Data augmentation

We perform several data augmentation techniques to boost player identification performance such data color jittering, random cropping, and random rotation by rotating each image in a tracklet by ± 10 degrees. Note that since we are dealing with temporal data, these augmentation techniques are applied per tracklet instead of per tracklet-image. In this section, we investigate the contribution of each augmentation technique to the overall accuracy. Table 5.2 shows the accuracy and weighted macro F1 score values after remov-

Table 5.2: Ablation study on different kinds of data augmentations applied during training. Removing any one of the applied augmentation techniques decreases the overall accuracy and F1 score.

Accuracy	F1 score	Color	Rotation	Random cropping
83.17%	**83.19%**	✓	✓	✓
81.58%	82.00%	✓	✓	
81.58%	81.64%	✓		✓
81.00%	81.87%		✓	✓

ing these augmentation techniques. It is observed that removing any one of the applied augmentation techniques decreases the overall accuracy and F1 score.

Inference technique

We perform an ablation study to determine how our tracklet score aggregation scheme of averaging probabilities after filtering out tracklets based on jersey number presence compares with other techniques. Recall from section 5.1.3 that for inference, we perform *visibility filtering* of tracklets and evaluate the model only on tracklets where jersey number is visible. We also include a *post-processing* step where only those window probability vectors p_i are averaged for which $argmax(p_i) \neq null$. The other baselines tested are described:

1. Majority voting: after filtering tracklets based on jersey number presence, each window probability $p_i \in \mathbf{P}$ for a tracklet is argmaxed to obtain window predictions after which a simple majority vote is taken to obtain the final prediction. For post-processing, the majority vote is only done for those window predictions with are not the *null* class.

2. Only averaging probabilities: this is equivalent to our proposed approach without visibility filtering and post-processing.

The results are shown in Table 5.3. We observe that our proposed aggregation technique performs the best with an accuracy of 83.17% and a macro weighted F1 score of 83.19%. Majority voting shows lower performance with accuracy of 80.59% even after the visibility filtering and post-processing are applied. This is because majority voting does not take into account the overall window level probabilities to obtain the final prediction since it

Table 5.3: Ablation study on different methods of probability aggregation.

Method	Accuracy	F1 score	Visiblility filtering	Postprocessing
Majority voting	80.59%	80.40%	✓	✓
Probability averaging	75.64%	75.07%		
Proposed w/o postprocessing	80.80%	79.12%	✓	
Proposed w/o visibility filtering	50.10%	48.00%		✓
Proposed	**83.17%**	**83.19%**	✓	✓

applies the argmax operation to each probability vector p_i separately. Simple probability averaging without visibility filtering and post-processing obtains a 7.53% lower accuracy demonstrating the advantage of visibility filter and post-processing step. The proposed method without the post-processing step lowers the accuracy by 2.37% indicating post-processing step is of integral importance to the overall inference pipeline. The proposed inference technique without visibility filtering performs poorly when post-processing is added with an accuracy of just 50.10%. This is because performing post-processing on every tracklet irrespective of jersey number visibility prevents the model to assign the *null* class to any tracklet since the logits of the *null* class are never taken into aggregation. Hence, tracklet filtering is an essential precursor to the post-processing step.

5.2 Transformer model

Transformers [78] are the existing standard in natural language processing (NLP) and are swiftly gaining traction in computer vision [79, 80, 81]. Motivated by the increasing success of transformers in computer vision, in this section, we introduce a transformer network for recognizing players through their jersey numbers. The transformer model takes player tracklets as input and outputs the probabilities of jersey numbers present.

In previous works [56] all images in a tracklet are annotated with the same label with the tracklet consisting of hundreds of frames. As a result, when sampling a fixed number of frames for training, it is possible that the frames may not have a jersey number visible. This leads to inconsistent and slow training. To address the issue, we implement a weakly-supervised training approach by generating approximate frame-level labels for jersey number presence (Section 5.2.3) and use the frame-level labels for faster training. The proposed transformer network performs better than the temporal 1D CNN model discussed in the previous section.

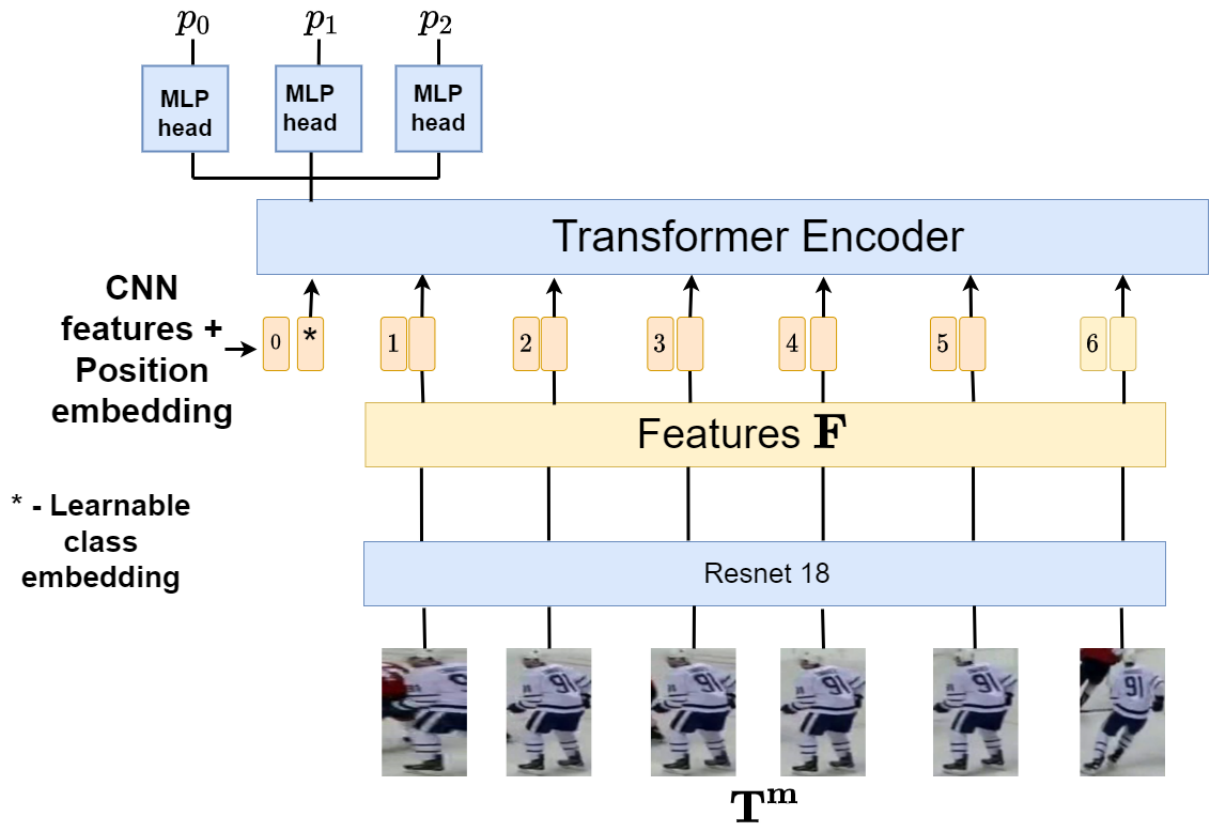


Figure 5.9: Network architecture for the proposed network. The input to the network is a temporal sequence of m images $\mathbf{T^m}$. Each image in the tracklet is passed through a ResNet18 network to obtain 512 dimensional features $\mathbf{F}$. The features are prepended with the [`class`] token and combined with learnable positional encoding.

5.2.1 Network architecture

The input to the network is a temporal sequence of m images $\mathbf{T^m} = \{I_i \in \mathbb{R}^{3\times300\times300}\}_{i=1}^{m}$ sampled from a player tracklet $\mathbf{T} = \{I_k : I_k \in \mathbb{R}^{300\times300\times3}\}_{k=1}^{n}$ of n images. The m images are randomly sampled from the tracklet T serving as a form of data augmentation. The sampling technique is discussed in Section 5.2.3. The images $\mathbf{T^m}$ are passed through a 2D CNN (Resnet18 [73]) to obtain m features $\mathbf{F} = \{f_i \in \mathbb{R}^{512}\}_{i=1}^{m}$. The Resnet18 is pretrained on static jersey number images using the image based jersey number dataset introduced in Chapter 4. The features $\mathbf{F}$ are input into a transformer encoder consisting of l layers with h multi-headed self-attention heads per layer. Each attention head has a constant dimension of $D_h \in \mathbb{R}^{64}$. Positional encoding $p_i \in \mathbb{R}^{512}$ are added to the features f_i. Instead of using fixed positional encoding, the positional encoding is learned. As per the Vision transformer [82], a `[class]` token similar to BERT [83] is prepended to the CNN features $\mathbf{F}$. The state of the `[class]` token at the final transformer layer is fed to three multi-layer perceptron (MLP) heads consisting of a layernorm [84] and linear layer. The output of the three MLP heads are three vectors. The first vector $p_0 \in \mathbb{R}^{86}$ denotes the probability distribution of the predicted jersey number considering each jersey number in the dataset as a separate class. The other two vectors $p_1 \in \mathbb{R}^{11}$ and $p_2 \in \mathbb{R}^{11}$ denote the probability distribution of the first and second digit of the predicted jersey number. The one additional class in the 11-dimensional vectors p_1 and p_2 denotes the absence of a jersey number

We utilize the multi-task loss for jersey number recognition introduced in Chapter 4 for training the network. Concretely, we let $y_0 \in \mathbb{R}^{86}$ denote the ground truth vector for the holistic jersey number class, and we let $y_1 \in \mathbb{R}^{11}$ and $y_2 \in \mathbb{R}^{11}$ denote the first digit and second digit ground truth vectors respectively. Let

$$\mathcal{L}_0 = -\sum_{i=1}^{86} y_0^i \log p_0^i \tag{5.1}$$

be the holistic jersey number component of the loss and

$$\mathcal{L}_1 = -\sum_{j=1}^{11} y_2^j \log p_1^j \tag{5.2}$$

and

$$\mathcal{L}_2 = -\sum_{j=1}^{11} y_1^j \log p_2^j \tag{5.3}$$

be the digit-wise losses. Instead of using fixed weights for the three losses, the loss weights are learned using the technique introduced in Kendall *et al.* [85], with the overall loss $\mathcal{L}$ given by:

$$\mathcal{L} = \frac{1}{\sigma_1^2}\mathcal{L}_0 + \frac{1}{\sigma_2^2}\mathcal{L}_1 + \frac{1}{\sigma_3^2}\mathcal{L}_2 + \log(\sigma_1) + \log(\sigma_2) + \log(\sigma_3) \tag{5.4}$$

where $\{\sigma_i\}_{i=1}^3$ are trainable parameters. The overall network architecture is illustrated in Fig 5.9.

5.2.2 Training details

Same as the temporal 1D CNN in the previous section, for handling the severe class imbalance in the dataset, the *null* class tracklets are sampled with a probability of $p_s = 0.1$. The network is trained with an Adam optimizer with an initial learning rate of 0.0001 and a batch size of 16. The learning rate is reduced by a factor of $\frac{1}{5}$ after 2500 iterations and again after 5000 iterations. Several data augmentation techniques such as random rotation by ± 10 degrees, randomly cropping 300×300 pixel patches from the tracklet images and color jittering are used while training. Each augmentation technique is used on a per-tracklet basis instead of a per-frame basis. The experiments are performed on two NVIDIA P-100 GPUs.

5.2.3 Training with approximate labels

The tracklets present in the training set can contain hundreds of frames such that the jersey number is only visible in a small subset of frames. Previous approaches in the literature [56] sample a fixed number of frames randomly from a tracklet without any information of where the jersey number is actually visible. Therefore certain sampled tracklets with a non-null jersey number class may not have a jersey number visible. A toy example depicting such a scenario is shown in Fig. 5.10. This leads to inconsistent training signals which results in slow/unstable training as we demonstrate in experiments. To address this issue, we create frame-level labels indicating the frames in the tracklet where the jersey number is visible.

To generate these frame level labels, let $\mathcal{M}$ be a model trained to predict a jersey number in static images and let $\mathbf{T} = \{I_k : I_k \in \mathbb{R}^{300\times300\times3}\}_{k=1}^n$ be a training tracklet consisting of n images I_k. The model $\mathcal{M}$ is run on every image I_k to obtain the probability

p_k of whether a jersey number is visible in the image I_k. This gives n probability scores $\{p_k \in [0,1]\}_{k=1}^{n}$. The n probability scores are thresholded with a binary threshold ϕ to obtain n binary values $\mathbf{B} = \{b_k \in \{0,1\}\}_{k=1}^{n}$. The value of b_k denotes the presence of jersey number in a tracklet frame.

$$b_k = 1 \text{ if jersey number present in frame} \tag{5.5}$$

$$b_k = 0 \text{ otherwise} \tag{5.6}$$

The algorithm to obtain approximate labels in summarized in Algorithm 2. The model $\mathcal{M}$ is a ResNet18 [73] pretrained on a a jersey number dataset consisting of static images introduced in Chapter 4.

After precomputing $\mathbf{B}$, let $\mathbf{T^m} = \{I_i \in \mathbb{R}^{300\times300\times3}\}_{i=l}^{l+m}$ where $l >= 1$ and $l + m <= n$ be the m images randomly sampled from a tracklet $\mathbf{T}$ for training. The corresponding $\mathbf{B^m} = \{b_i \in \{0,1\}\}_{i=l}^{l+m}$ where $l > 1$ and $l + m <= n$ has at least one $b_i = 1$. This ensures that at least one image with a visible jersey number is present in the sampled tracklet.

For implementation, we let $\mathbf{I}$ denote the indices in the vector B for which $b_k = 1$. We randomly sample an index *start_idx* from $\mathbf{I}$ and then sample m frames from the tracklet $\mathbf{T}$ starting from index *start_idx* to *start_idx* $+ m$. A random offset $o \in [0, m)$ is subtracted from *start_idx* to ensure that the sampled tracklet $\mathbf{T^m}$ may have a non-zero jersey number label at any sampled frame (and not necessarily always at the beginning). The algorithm is provided in Algorithm 3.

Algorithm 2: Algorithm for creating approximate frame-wise jersey number labels.

Input: Player tracklet $\mathbf{T}$, Image-wise jersey number model $\mathcal{M}$, Threshold ϕ
Output: Frame for labels $\mathbf{B}$
Initialize: $\mathbf{B} = null$
for $I_k \in \mathbf{T}$ **do**
    $p_k = \mathcal{M}(I_k)$
    if $p_k > \phi$ **then**
        $\mathbf{B}$.append(1)
    else
        $\mathbf{B}$.append(0)
    end
end

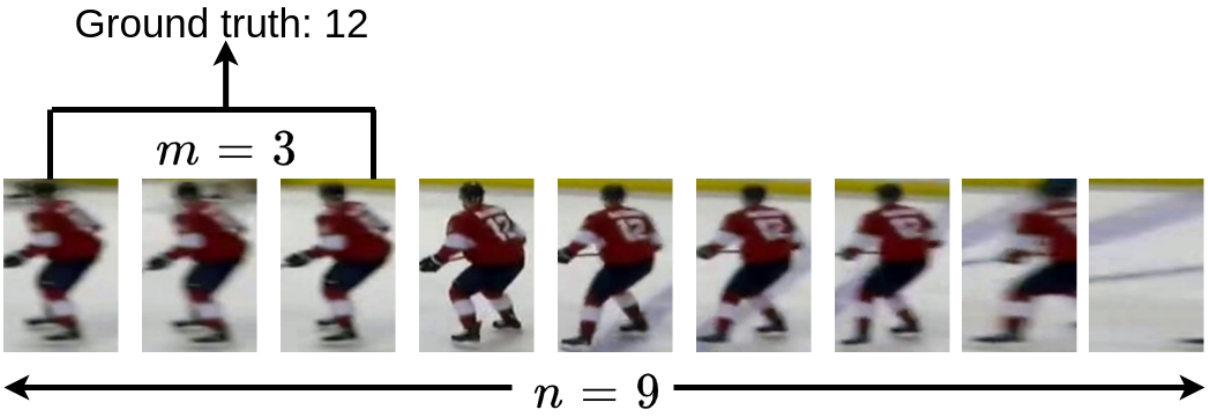


Figure 5.10: Toy exampling for a tracklet (of length $n = 9$ frames) where sampling $m = 3$ consecutive frames from the start leads to a sequence with ground truth 12 with no jersey number visible.

5.2.4 Results

The tracklet dataset discussed in Section 5.1.4 is used for training and testing. The inference for each tracklet is carried out using the interference technique explained in Section 5.1.3. We compare the performance of the proposed network the temporal 1D CNN model. The network performs better demonstrating the effectiveness of the proposed approach. The results are shown in Table 5.4.

We also re-implement Chan *et al.* [56] from scratch due to the unavailability of publicly-available code and dataset. The proposed approach obtains 10.1% more accuracy than Chan *et al.*. The reasons for better accuracy of the proposed approach compared to Chan *et al.* are: (1) Chan *et al.* use a temporal receptive field of only 16 frames whereas the proposed approach has a more than double receptive field of 40 frames; (2) lack of data augmentation such as such as random rotation, color jittering in Chan *et al.*; (3) the dataset used in our work is half the size and much more skewed (50.4% *null* class) compared to Chan et al. due to which their late fusion network overfits on our dataset and (4) Chan *et al.* does not incorporate techniques to handle dataset class imbalance.

We also compare the proposed weakly-supervised training scheme making use of approximate labels to sampling frames randomly from any point in the tracklet (not using approximate frame labels) [56]. The proposed scheme of training with the help of approximate labels improves the training convergence as illustrated in Fig. 5.11. The validation

Algorithm 3: Algorithm for sampling m frames $\mathbf{T^m}$ for a tracklet T.

Input: Player tracklet $\mathbf{T}$, Frame-wise jersey number labels $\mathbf{B}$, Sampling sequence length m
Output: Sampled tracklet images $\mathbf{T^m}$
Initialize: $\mathbf{T^m} = null$
`// numpy function`
$\mathbf{I} = \texttt{np.where}(\mathbf{B} == 1)$
$start\_idx = \texttt{random\_sample}(\mathbf{I})$
$o = \texttt{randint}(m)$
$start\_idx = \texttt{max}(0, start\_idx - o)$
$T_m = \mathbf{T}[start\_idx : start\_idx + m]$

Table 5.4: The result of the best performing model compared to temporal 1D CNN.

Model	Accuracy	F1 score
Proposed	**83.37 %**	**84.14 %**
Temporal 1D CNN	83.17%	83.19%

accuracy curves are shown in Fig. 5.12. The reason for improved convergence with the proposed training scheme is that all the tracklet mini-batches sampled using approximate labels have the jersey number visible which results in a consistent training signal.

5.2.5 Ablation studies

The number of transformer layers l, the number of attention heads h and length of sequence for training/evaluation m are important parameters affecting the overall performance. Hence, an ablation study is performed to determine the best value for each parameter.

Attention heads

We perform an ablation study to determine to best value of the number of attention heads per transformer layer h. The values of $h \in \{2, 4, 6, 8, 10\}$ were tested while keeping the number of transformer layers l and sequence length for training/evaluation m constant ($l = 2, m = 30$). The value of $h = 8$ showed the best performance with an accuracy

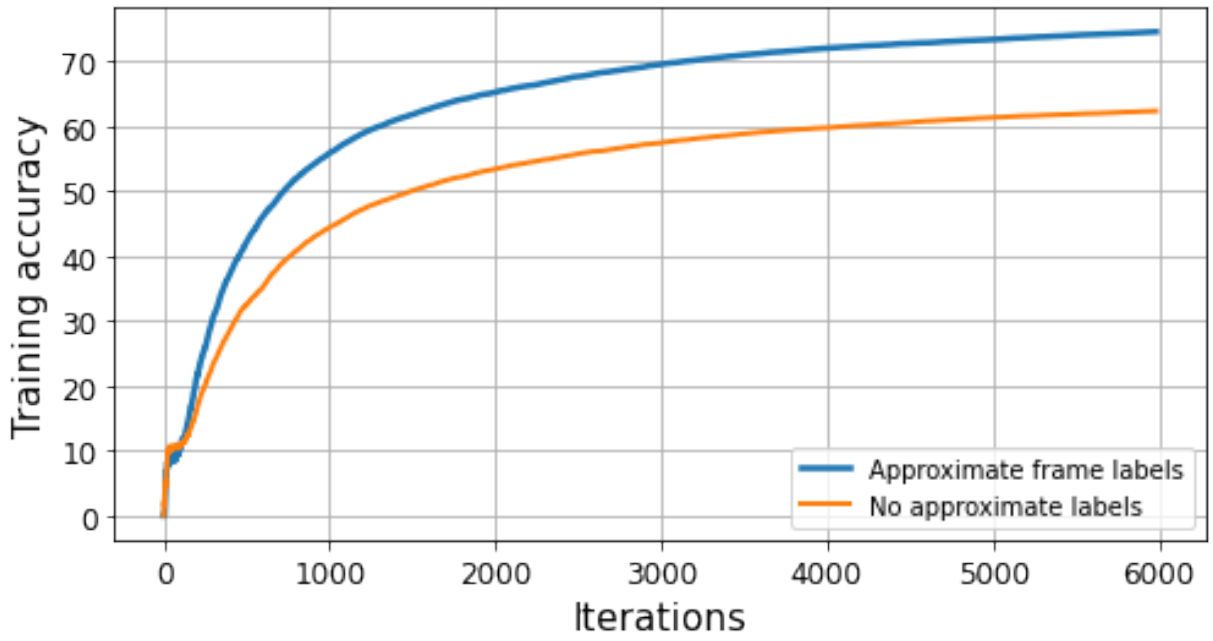


Figure 5.11: Training curves corresponding to a network with transformer layers $l = 2$, attention heads per layers $h = 8$ and training sequence length $m = 40$ Training with approximate labels makes the network converge faster while training.

of 83.6% and a weighted F1 score of 84.2%. Table 5.5 shows the accuracy and F1 score values at the different values of h tested. Using more than 8 attention heads resulted in a performance decrease due to overfitting.

Transformer layers

We determine to best value of the number of transformer layers l by testing $l \in \{2, 4, 6, 8\}$ while keeping the number of attention heads per layer h and the sequence length used for training/evaluation m constant ($h = 8, m = 30$). From Table 5.6, the best accuracy value of 83.37% and F1 score of 83.85% was obtained with $l = 2$. The performance of the network declines after increasing the transformer layers from $l = 2$ to $l = 8$. This is because of overfitting since the number of parameters in the model increases around four times from $\sim$ 3.2 million when $l = 2$ to $\sim$ 12.6 million when $l = 8$ with no significant improvement in accuracy.

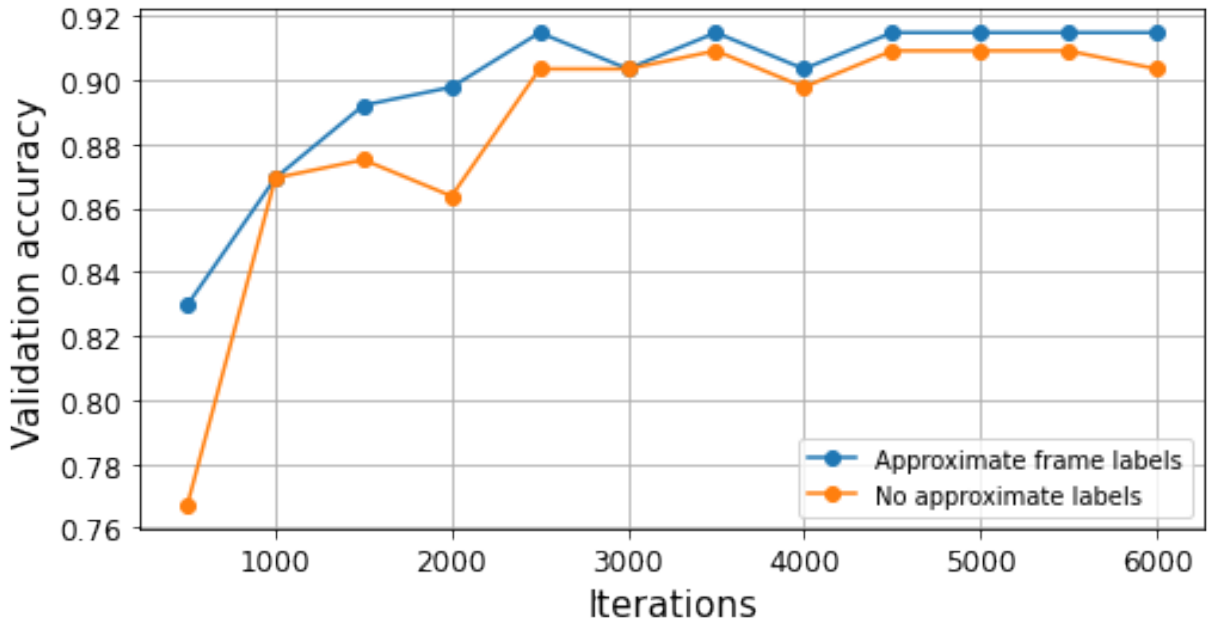


Figure 5.12: Validation accuracy curves corresponding to a network with transformer layers $l = 2$, attention heads per layers $h = 8$ and training sequence length $m = 40$. The initial accuracy at iteration number 500 is 6.2% higher when training with approximate labels (blue color curve). The network also converges faster and obtains a higher accuracy value with approximate label based training.

Sequence length

We determine the best value of the training and evaluation sequence length m by keeping the transformer layers and number of attention heads per layer constant. The values of $m \in \{10, 20, 30, 40, 50\}$. From Table 5.7, the lowest performance was shown by $m = 10$ with an accuracy of 81.58%. Increasing m to 20 improved the accuracy and F1 score due to to increase in receptive field of the network. However, the accuracy between $m = 20$ to $m = 50$ remained the same. The best performance was obtained by $m = 40$ with an accuracy of 83.37% and F1 score of 84.14%. Further increasing sequence length m beyond 40 did not improve performance.

Table 5.5: Ablation study to determine the best value of attention heads per layer h keeping number of layers l and sequence length m constant ($l = 2, m = 30$).

h	Accuracy	F1 score
2	82.97%	83.65%
4	82.97%	83.32%
6	83.17%	83.74%
8	**83.37 %**	**83.85%**
10	82.38%	82.90%

Table 5.6: Ablation study to determine the best value layers l keeping number of attention heads h and sequence length m constant ($h = 8, m = 30$).

l	Accuracy	F1 score
2	**83.37 %**	**83.85%**
4	81.98%	82.74%
6	81.58%	82.17%
8	82.77%	83.17%

Table 5.7: Ablation study to determine the best value of training and evaluation sequence length m keeping number of attention heads h and number of layers l constant ($h = 8, l = 2$).

m	Accuracy	F1 score
10	81.58%	81.75%
20	83.37%	83.76%
30	83.37%	83.85%
40	**83.37**%	**84.14**%
50	83.37%	84.07%

5.3 Summary

In this chapter we introduced two networks for identifying player jersey numbers from player tracklets to incorporate temporal information available in broadcast video. The

first network is a 1D temporal CNN network that makes use of a series of residual 1D temporal convolutional blocks to output the probability of the jersey number present in a tracklet. An inference technique based on visibility filtering is implemented that outperforms majority voting and probability averaging. Additionally, data augmentation methods such as random cropping, color jittering and random rotation are also implemented to improve overall accuracy by approximately 2%. Appropriate ablation studies are conducted to demonstrate the effectiveness of the inference technique and data augmentations used. The second network is a transformer that takes tracklet frame CNN features and output the jersey number probability. The transformer makes use of the same data augmentation and inference techniques used in the temporal 1D CNN network. In addition, weakly supervised learning is performed by generating labels for visibility of jersey numbers in tracklet images which leads to faster training and convergence. The two networks are tested on a new player tracklet dataset where player tracklets are manually annotated with the jersey number present. Experimental results demonstrate that the transformer network performs better than the previous state-of -the-art[56] by almost 10%.

Chapter 6

Overall system

The player tracking and identification techniques introduced in the previous chapters discussed the result of each component separately. In this chapter, we integrate the player tracking (Chapter 3) and player identification (Chapter 5) methods into a holistic pipeline (Fig 6.4). Concretely, the player tracklets obtained from the tracking model and used as input for player identification. Remarking that the team roster and/or player shifts are often available through the NHL play-by-play data, we incorporate the roster and shift data into the pipeline to reduce the search space of the player identification method. However, incorporating team roster requires team affiliations of each player. Therefore, in this chapter, we introduce a team-identification method which is integrated into the pipeline. For team identification, away-team jerseys are grouped into a single class and home-team jerseys are grouped in classes according to their jersey color. A convolutional neural network is then trained on the team identification dataset. Finally, the player tracking done in image coordinate system is converted to ice rink coordinated using an automatic registration network [63] to obtain position of players on ice rink.

6.1 Team identification

In this section, we present the team identification model used to get team affiliations. We discuss the dataset used, methodology including training details for team identification and the results obtained.

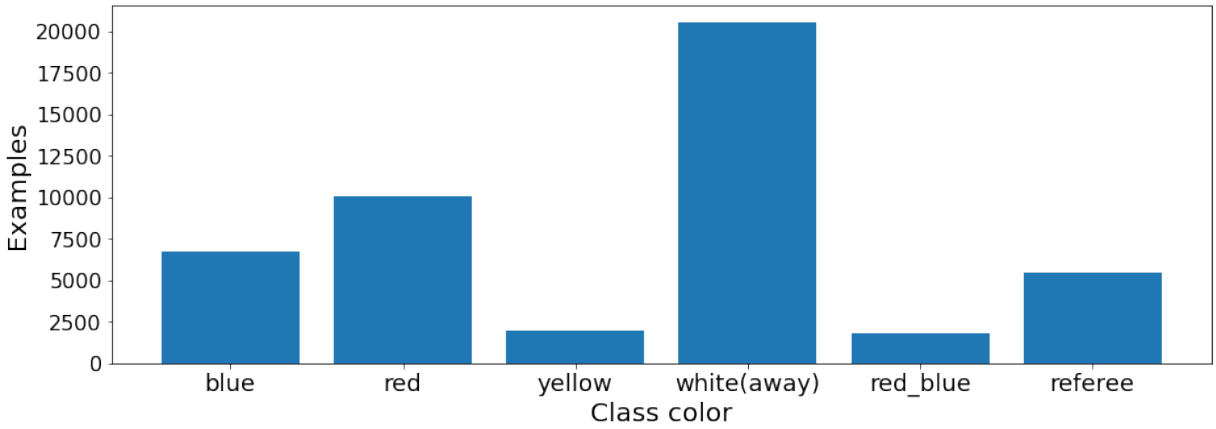


Figure 6.1: Classes in team identification and their distribution. The 'ref' class denotes referees.

6.1.1 Dataset

The team identification dataset is obtained from the same games and clips used in the player tracking dataset 3). The train/validation/test splits are also identical to player tracking data. We take advantage of the fact that the away team in NHL games usually wear a predominantly white colored jersey with color stripes and patches, and the home team wears a dark colored jersey. For example, the Toronto Maple Leafs and the Tampa Bay Lightning both have dark blue home jerseys and therefore can be put into a single 'Blue' class. We therefore build a dataset with five classes (blue, red, yellow, white, red-blue and referees) with each class composed of images with same dominant color. The data-class distribution is shown in Fig. 6.1. Fig. 6.2 shows an example of the blue class from the dataset. The training set consists of 32419 images. The validation and testing set contain 6292 and 7898 images respectively.

6.1.2 Methodology

For team identification, we use a ResNet18 [73] pretrained on the ImageNet dataset [77], and train the network on the team identification dataset by replacing the final fully connected layer to output six classes. The images are scaled to a resolution of 224×224 pixels for training. During inference, the network classifies whether a bounding box belongs to the away team (white color), the home team (dark color), or the referee class. For inferring

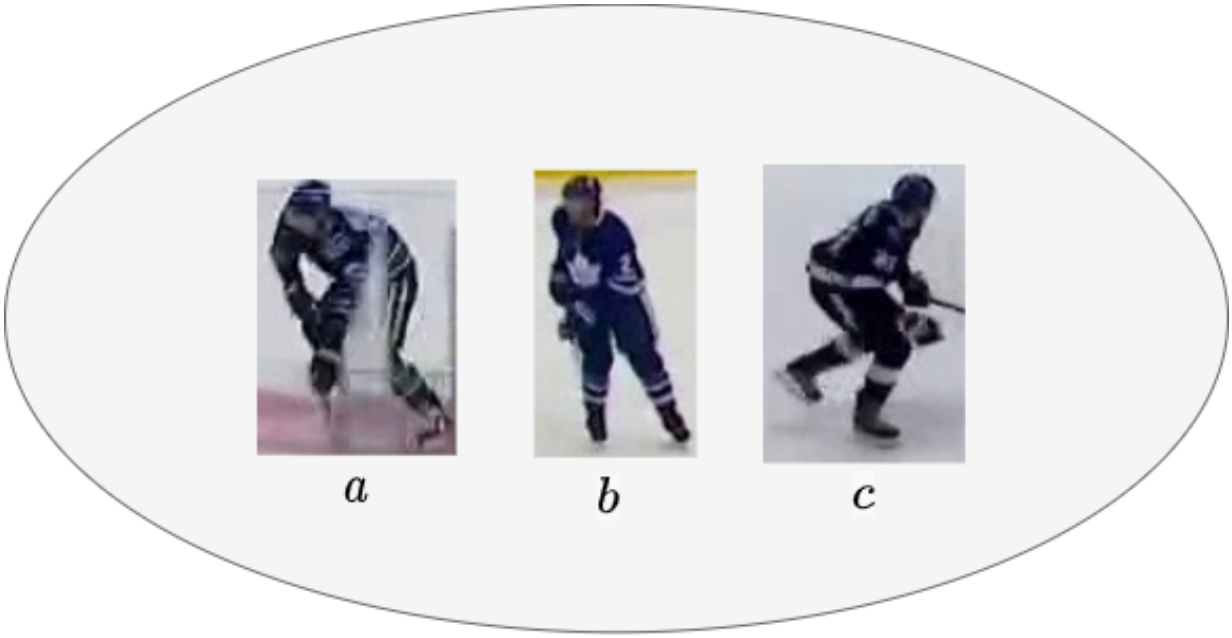


Figure 6.2: Examples of 'blue' class in the team identification dataset. Home jersey of teams such as (a) Vancouver Canucks (b) Toronto Maple Leafs and (c) Tampa Bay Lightning are blue in appearance and hence are put in the same class.

the team for a player tracklet, the team identification model is applied to each image of the tracklet and a simple majority vote is used to assign a team to the tracklet. This way, the tracking algorithm helps team identification by resolving errors in team identification.

6.1.3 Training details

We use the Adam optimizer with an initial learning rate of .001 and a weight decay of .001 for optimization. The learning rate is reduced by a factor of $\frac{1}{3}$ at regular intervals during the training process. We do not perform data augmentation since performing color augmentation on white away jerseys makes it resemble colored home jerseys.

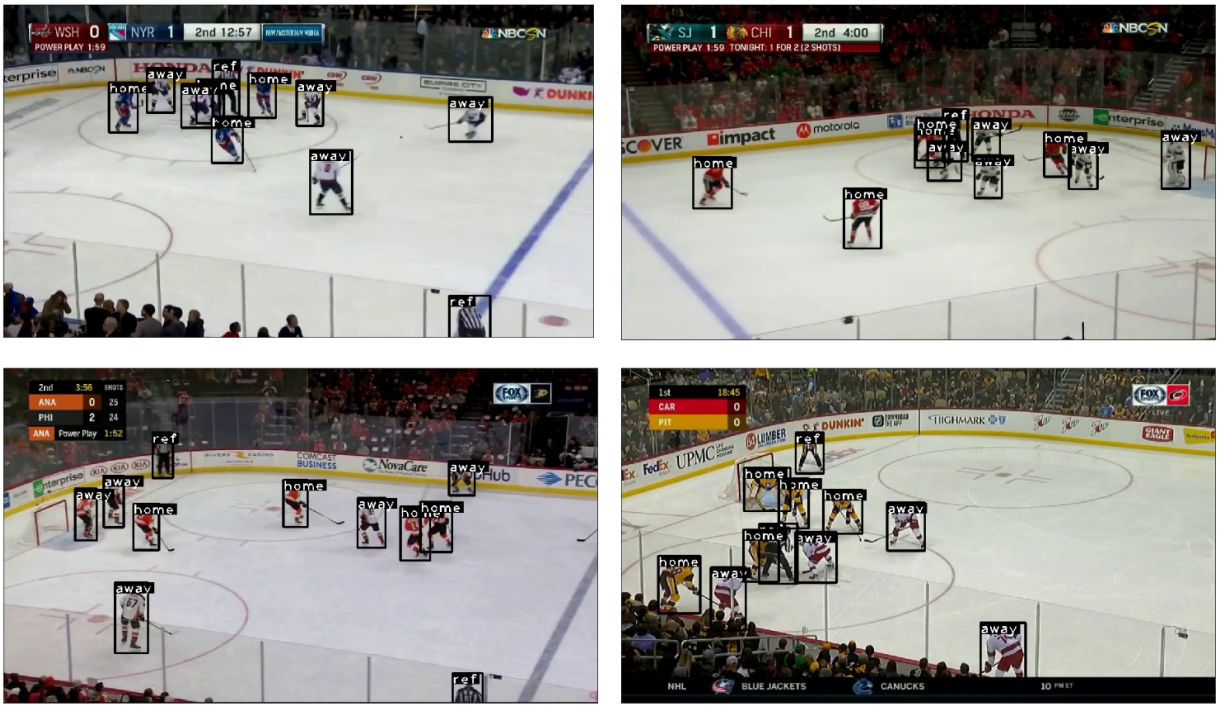


Figure 6.3: Team identification results from four different games that are each not present in the team identification dataset. The model performs well on data not present in dataset, which demonstrates the ability to generalize well on out of sample data points.

6.1.4 Results

The team identification model obtains an accuracy of 96.6% on the team identification test set. Table 6.1 shows the macro averaged precision, recall and F1 score for the results. The model is also able to correctly classify teams in the test set that are not present in the training set. Fig. 6.3 shows some qualitative results where the network is able to generalize on videos absent in training/testing data. We compare the model to color histogram features as a baseline. Each image in the dataset was cropped such that only the upper half of jersey is visible. A color histogram was obtained from the RGB representation of each image, with n_{bins} bins per image channel. Finally a support vector machine (SVM) with an radial basis function (RBF) kernel was trained on the normalized histogram features. The optimal SVM hyperparameters and number of histogram bins were determined using grid search by doing a five-fold cross-validation on the combination of training and validation

Table 6.1: Team identification accuracy on the team-identification test set.

Method	Accuracy	Precision	Recall	F1 score
Proposed	**96.6**	**97.0**	**96.5**	**96.7**
SVM with color histogram	82.0	81.7	81.5	81.5

set. The optimal hyperparameters obtained were $C = 10$, $\gamma = .01$ and $n_{bins} = 12$. Compared to the SVM model, the deep network approach performs 14.6% better on the test set demonstrating that the deep network (CNN) based approach is superior to simple handcrafted color histogram features.

6.2 Holistic pipeline

In this section, we explain the holistic pipeline combining the tracking, team identification and player identification models. We discuss the methodology used, results obtained and the failure cases for the pipeline.

6.2.1 Methodology

Given a test video, the player tracking, team identification, and player identification methods discussed are combined together for tracking and identifying players and referees in broadcast video shots. Given a test video shot, we first run player detection and tracking to obtain a set of player tracklets $\tau = \{T_1, T_2, ...T_n\}$. For each tracklet T_i obtained, we run the player identification model to obtain the player identity.

To incorporate player shifts for improving player identification performance, the game time in the video needs to be synced with the player shifts database, denoted by $\mathcal{S}$. $\mathcal{S}$ contains player shifts according to game time along with the corresponding jersey number and team affiliations. To read game time from broadcast video clips, the EasyOCR[1] library was used. Let t_s denote the starting game time and t_e denote the ending game time of a short video clip obtained using OCR. The player shifts S' that are present in the game time between t_s and t_e are extracted from the player shift database $\mathcal{S}$. The set $\mathcal{S}'$ can be expressed as a union $S' = S_h \cup S_a$ where S_h and S_a are the subsets of home and away shifts present in the set S'. Let the sets $\mathcal{H}$ and $\mathcal{A}$ denote the jersey numbers corresponding to S_h and S_a respectively. We then construct *shift vectors* $v_h \in \mathbb{R}^{86}$ and $v_a \in \mathbb{R}^{86}$ that encode

[1]Found online at: https://github.com/JaidedAI/EasyOCR

the jersey numbers present in the home and away teams. Let *null* denote the no-jersey number class and j denote the index associated with jersey number n_j in p_{jn} vector.

$$v_h[j] = 1, \text{if } n_j \in \mathcal{H} \cup \{null\} \tag{6.1}$$
$$v_h[j] = 0, \ otherwise \tag{6.2}$$

similarly,

$$v_a[j] = 1, \text{if } n_j \in \mathcal{A} \cup \{null\} \tag{6.3}$$
$$v_a[j] = 0, \ otherwise \tag{6.4}$$

Based on whether the player tracklet belongs to the home or the away team, the final player identity Id is computed as

$$Id = argmax(p_{jn} \odot v_h) \tag{6.5}$$

,(where $\odot$ denotes element-wise multiplication) if the tracklet belongs to the home team, otherwise,

$$Id = argmax(p_{jn} \odot v_a) \tag{6.6}$$

, if the player belongs to the away team. If instead of the player shifts, the game roster is available, sets $\mathcal{H}$ and $\mathcal{A}$ denote the jersey numbers of the home and away team respectively present in the roster. The overall algorithm is summarized in Algorithm 4. Fig. 6.4 depicts the overall system.

Algorithm 4: Holistic algorithm for player tracking and identification.

Input: Input Video V, Tracking model T_r , Team ID model $\mathcal{T}$, Player ID model $\mathcal{P}$, v_h , v_a
Output: Identities $\mathcal{ID} = \{Id_1, Id_2.....Id_n\}$
Initialize: $\mathcal{ID} = \phi$
$\tau = \{T_1, T_2, ...T_n\} = T_r(V)$
for $T_i \quad in \quad \tau$ **do**
    $team = \mathcal{T}(T_i)$
    $p_{jn} = \mathcal{P}(T_i)$
    if $team == home$ **then**
        $Id = argmax(p_{jn} \odot v_h)$
    else if $team == away$ **then**
        $Id = argmax(p_{jn} \odot v_a)$
    else
        $Id = ref$
    end
    $\mathcal{ID} = \mathcal{ID} \cup Id$
end

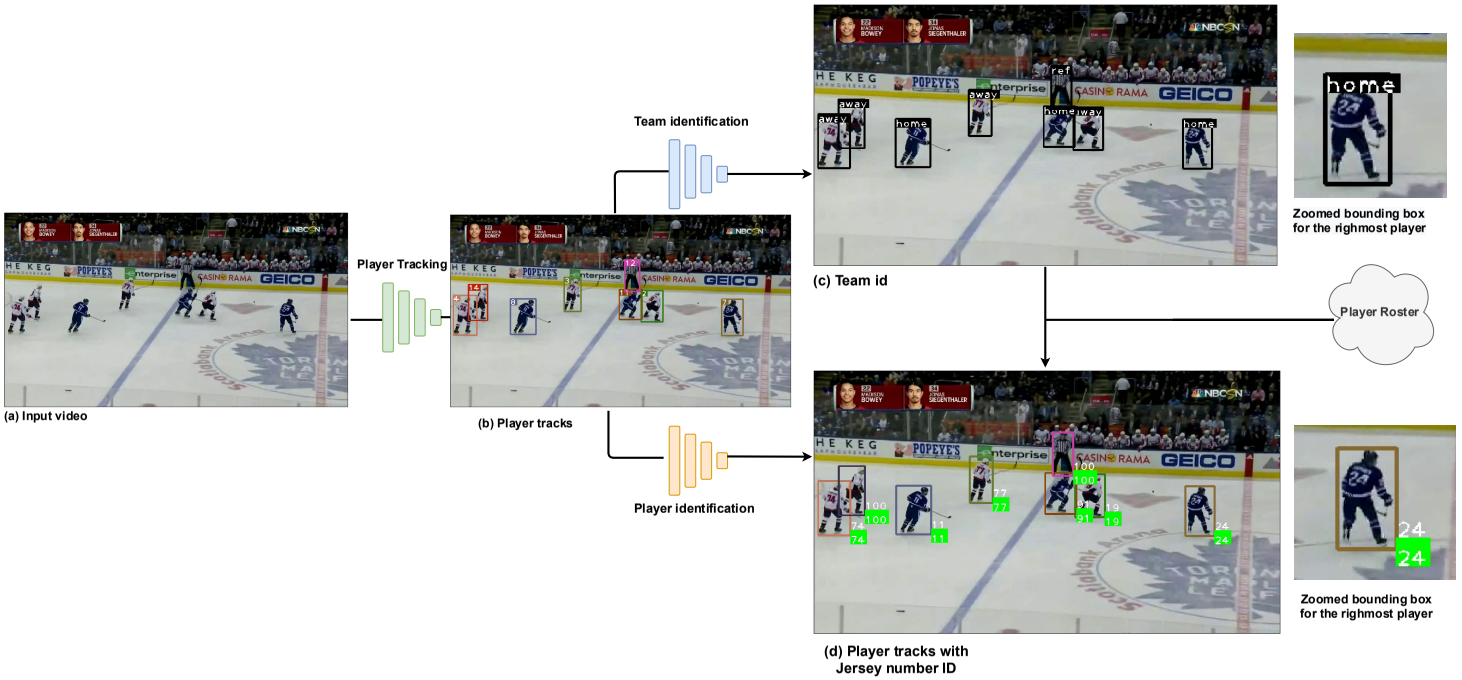


Figure 6.4: Overview of the player tracking and identification system. The tracking model takes a hockey broadcast video clip as input and outputs player tracks. The team identification model takes the player track bounding boxes as input and identifies the team of each player along with identifying the referees. The player identification model utilizes the player tracks, team data and game roster data to output player tracks with jersey number identities.

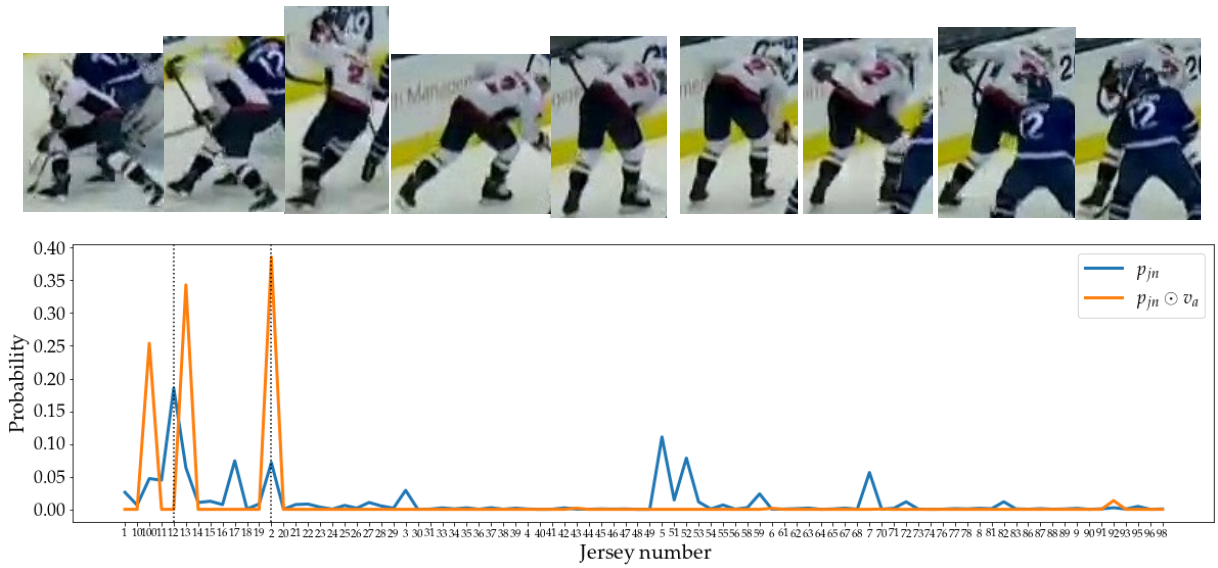


Figure 6.5: **Top row:** Example of a 'hard' tracklet where the ground truth jersey number 2 is tilted. There is also a heavy occlusion with the opposition player with jersey number 12. Note that the original tracklet contains 89 frames, however, only a subset of frames is shown here due to space constraints. **Bottom row:** For the tracklet shown in the top row, p_{jn} is the probability of jersey number present in the tracklet (blue color). Orange color line is the normalized probability $p_{jn} \odot v_a$, i.e, the probability of jersey number multiplied by the shift vector v_a. For p_{jn} the highest confidence value exists for jersey number 12 (first vertical line from left), which is incorrect. Multiplying with the shift vector v_a corrects the mistake by making the system focus only on the jersey number present in the away team during the game shift, after which the probability of the correct jersey number 2 (second vertical line from left) becomes the greatest.

6.2.2 Results

We evaluate the network on the player tracklets obtained by running a tracking algorithm [1, 86] on the 13 test videos. This evaluation is different from the evaluation done in Chapter 5, since the player tracklets are now obtained from the player tracking algorithm (rather than being manually annotated). The accuracy obtained by incorporating player shifts using OCR into player identification is compared to two baselines: (1) not incorporating any kind of roster/shift information, and (2) using player rosters available at the start of the game instead of player shifts.

From Table 6.2, not using any shifts/roster data using the transformer network obtains a mean accuracy of 82.02%, that is 4.12% greater than the temporal CNN network . With the transformer network, incorporating player shifts obtains the best mean accuracy of 87.97%, which is $\sim 6\%$ more than not using any shift or roster data. In fact, every video except the first video in the test set obtains equal or more accuracy when using the player shift data. This is because using player shifts helps the algorithm focus on a smaller subset of possible players present at a particular time. The lower accuracy of the first test video is due to inaccuracies in the shifts database. Using the player roster with transformer network obtains an accuracy 86.32%, which is just 1.65% lower than the accuracy obtained when using player shifts, which demonstrates that even if player shifts are not available, using the available roster can provide performance comparable to using player shift data. Fig. 6.5 shows an example of a tracklet where incorporating player shifts corrects the prediction of the model that does not use any shift or roster information.

Figure 6.6: Example of a tracklet where the same identity is assigned to two different players due to an identity switch. These kind of errors in player tracking gets carried over to player identification, since a single jersey number cannot be associated with this tracklet.

6.2.3 Failure cases

There are three main sources of error:

1. Identity switches of the tracking model, where the same ID is assigned to two different player tracks. These are illustrated in Fig. 6.6;
2. Misclassification of the player's team, as shown in Fig. 6.7, which causes the player jersey number probabilities to get multiplied by the incorrect roster vector; and
3. Incorrect jersey number prediction by the network.

Table 6.2: Overall player identification accuracy for 13 test videos. The mean accuracy for identification increases by 5.95% after including the player shift data. Note that Tformer stands for transformer.

Video number	Tformer w/ shift data	Tformer w/ roster data	Tformer w/o shift/roster data	Temporal 1D CNN w/o shift/roster data
1	90.70%	95.35%	90.60%	90.60%
2	**91.43%**	85.71%	74.29%	57.1%
3	**87.72%**	87.72%	84.2%	84.2%
4	**80.00%**	76.0%	72.00%	74.0%
5	**83.33%**	83.33%	81.48%	79.6%
6	**90.00%**	90.0%	90.00%	88.0%
7	**85.07%**	80.60%	73.13%	68.6%
8	**93.75%**	93.75%	91.6%	91.6%
9	**94.45%**	93.18%	88.6%	88.6%
10	**93.02%**	88.37%	83.72%	86.04%
11	**82.22%**	80.00%	71.11%	44.44%
12	**84.85%**	84.85%	84.85%	84.85%
13	**86.11%**	83.33%	80.56%	75.0%
Mean	**87.97%**	86.32%	82.02%	77.9%

Figure 6.7: Example of a tracklet where the team is misclassified. Here, the away team player (white) is occluded by the home team player (red), which causes the team identification model to output the incorrect result. Since the original tracklet contains hundreds of frames, only a subset of tracklet frames is shown.

6.3 Tracking on ice-rink

In the previous section players were tracked in the image pixel coordinate system. However, in order to calculate on-rink metrics such as player velocities, analyzing player formation and also for downstream computer vision tasks such as game event recognition, the player locations need to be obtained in ice-rink coordinates. Therefore in this section, we explain how a homography registration system [63] can be used in conjugation with the system developed to obtain player locations on ice. A puck localization system (Appendix B) may also be used to obtain locations puck on the ice-rink.

To calculate the position of the player on ice rink, in a video frame f_i, let $X_{img} \in \mathbb{R}^2$ denote the position of the player in image coordinates. Let $H \in \mathbb{R}^{3\times3}$ be the homography matrix obtained using the automatic homography registration model from Fani *et. al.* [63] that maps the image pixel coordinates to ice rink coordinates in each video frame f_i using a Resnet18 [73] based regressor. Then the rink location of the player $X_{rink} \in \mathbb{R}^2$ is calculated as:

$$X_{rink} = HX_{img} \tag{6.7}$$

Fig. 6.10 shows some qualitative results of the player tracking and identification system combined with a homography registration model on a test set video. Red circles denote home team player, cyan circles represent away team player and yellow circles denote the referee. Fig 6.8 shows the magnified version of the first image in Fig 6.10. The numbers in each frame denote the player jersey number. In Fig 6.8, the system identifies the player with jersey number 24 when the jersey number is not yet visible. This is because the jersey number becomes visible in a future frame after which the system assign the number to the

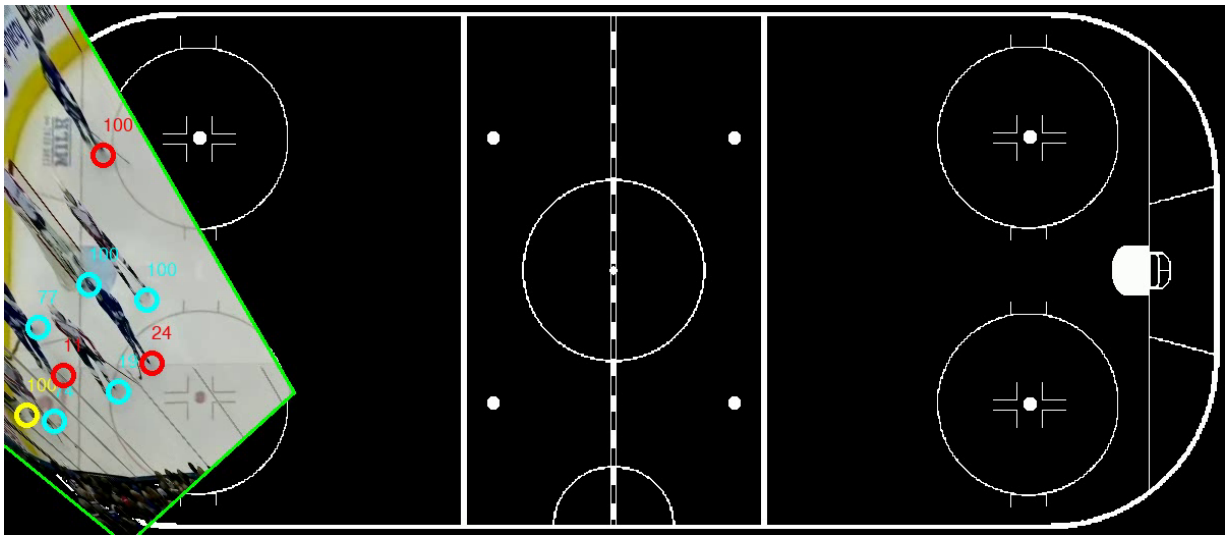


Figure 6.8: The system identifies the player with jersey number 24 when the jersey number is not yet visible. This is because the jersey number becomes visible in a future frame after which the system assign the number to the whole tracklet.

whole tracklet. The system allows a seamless analysis of the players skating on the ice ring along with their team affiliations and identities. Additionally, Fig. 6.9 shows the puck trajectory in the video.

6.4 Summary

In this chapter, we have introduced and implemented an automated offline system combining player tracking, team identification and player identification models for the challenging problem of player tracking and identification in ice hockey. If available, the systems makes use of game roster or shift data to further increase player identification accuracy by allowing the system to focus on a smaller subset of possible players present at a particular time for identification. The system takes as input broadcast hockey video clips from the main camera view and outputs player trajectories on screen along with their teams and identities. In order to obtain player trajectory location on the ice rink, an automatic homography registration system can be used. Additionally, the location of puck on the ice rink can be obtained by using a separate puck localization model. In this way both player

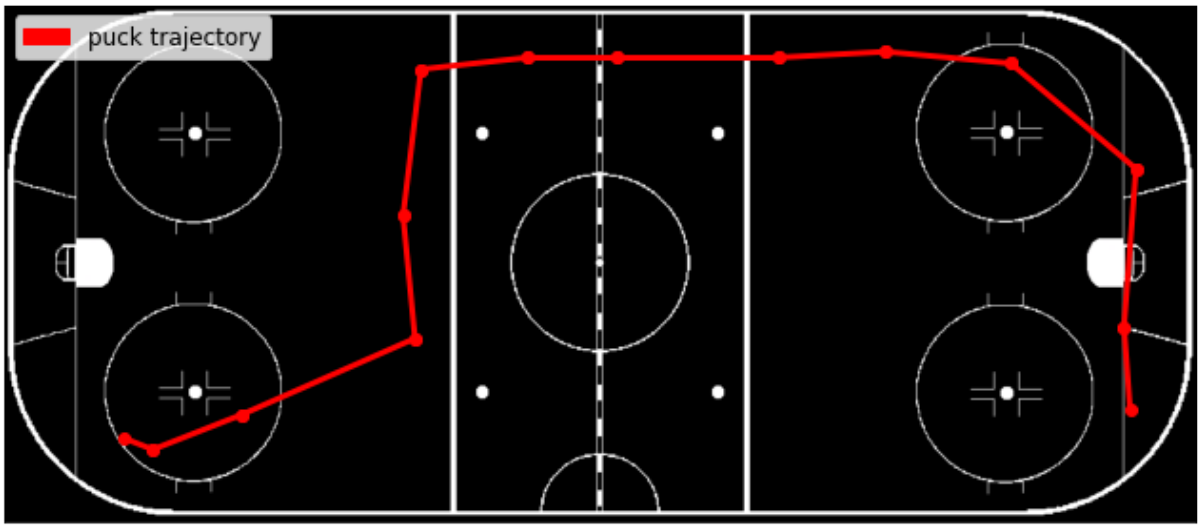


Figure 6.9: The predicted puck trajectory for the test video.

and puck locations on the ice rink may be obtained from broadcast video which can be used for further high level analysis by analysts and scouts.

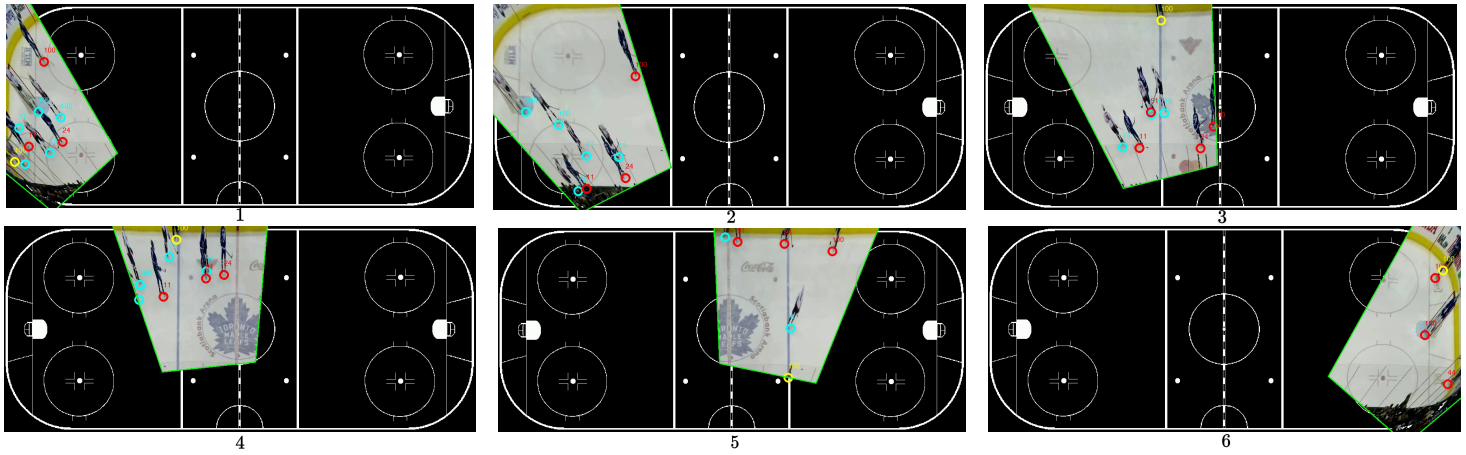


Figure 6.10: Qualitative results of the player tracking and identification system combined with a homography registration model on a test set video. The frames are sampled at 0.5 fps because of space constraints. Red circles denote home team player, cyan circles represent away team player and yellow circles denote the referee.

Chapter 7

Conclusion and future work

In this chapter, we summarize the contributions of the thesis. We discuss the limitations of the proposed system and also consider the possible future research directions.

7.1 Summary of contributions

The contributions of this thesis can be summarized as as follows:

1. **Holistic tracking and identification system:** A player tracking and identification system is designed and implemented to track and identify players in broadcast NHL videos. NHL broadcast videos are input to the systems after which players are tracked and identified. A network trained to identify player team affiliations is further incorporated to make use of the game roster and player shifts data, further improving the overall accuracy by 6%. Further, a trained automatic homography registration model [63] and puck localization model (Appendix B) are utilized to obtain the positions of the players and puck on the ice rink.

2. **Hockey player tracking :** Five state-of-the-art algorithms are tested on hockey dataset and their performance and failure cases are analysed. It is concluded that algorithms with a simple linear motion model do not perform well for hockey player tracking in broadcast video, demonstrated by the best performance of the graph formulation based MOT neural solver model [1] on the hockey dataset. The main source of player re-id failure were identity switches resulting from player going in and out of broadcast camera field-of-view (accounting for 65% of total identity switches).

3. **Multi-task network for jersey number identification:** A multi-task loss function is designed and implemented to recognize jersey numbers from images. Remarking that the player jersey number in ice hockey are either one or two digit numbers, they can wither be modelled as two separate classes or a single holistic class for classification. After hypothesizing that learning both representation together is a multi-task loss function could improve generalization [72], the multi-task network was implemented and it was experimentally demonstrated that the multi-task network performed better than the digit-wise and holistic setting. The experiments were followed by a thorough ablation study further verifying the hypothesis.

4. **Transformer based tracklet identification network:** A transformer based tracklet identification network is introduced and implemented to identify player jersey number for player tracklets obtained from the player tracking model. Weak labels indicating the presence of the jersey number in individual tracklet frames are generated. Utilizing the weak labels for training led to faster training and convergence. The network is compared to Chan *et al.* [56] and a temporal CNN network developed as a baseline. Experiments results demonstrate better performance of the proposed network compared to Chan *et al.* [56] and the temporal CNN network.

5. **New dataset for player tracking and identification:** There are no publicly available datasets for hockey player tracking, jersey number identification from images, tracklet identification and team identification. Therefore, new datasets for the aforementioned problems are created. The player tracking dataset is obtained from 25 NHL games encompassing 84 videos. The dataset created for player identification from static images consisting of $50,000$ images is the biggest dataset of such kind in the literature.

7.2 Limitations

Although the tracking and identification system developed in this thesis achieves good results, it is not perfect. This section discusses the major limitations associated with the system.

7.2.1 Pan identity switches

As previously mentioned in Chapter 3 identity switches which result from the player going outside and coming back into the camera field of view due to panning pose a big challenge

Figure 7.1: In online tracking and identification systems, jersey number can only be inferred for the frames after the number becomes visible.

to the tracking system. This is because when the player goes out of the field of view on screen, there are no reliable features to determine when and where the player will re-enter.

7.2.2 Identification in absence of jersey number

Although the player identification model introduced in the thesis achieves good results, especially when player shift data is available (87%), the identification problems becomes much more difficult when player jersey number is not visible in a given tracklet. This issue can be resolved by incorporating features such as player handedness for player identification explained in the next section.

7.2.3 Offline nature of the system

The player tracking and identification system introduced in this thesis is offline by design. The player detection model is first used to generate player detections for input into the tracking model. The overall system uses an offline tracking model that makes use of future video to generate player tracklets. Also, the player identification is also offline since the tracklet identification model is run once the full tracklets are obtained from tracking results. However, an offline systems is unsuitable for real-time tasks where tracking and identification results are desired *on the go* while the broadcast game video is streaming. The online setting is a more difficult problem since unlike the system developed in this

Figure 7.2: General pseudo labelling technique for semi supervised learning. A model is first trained on limited labeled data, which is used to infer labels on unlabeled data. Then the model is re-trained on combined data with data augmentations applied. The process is repeated in cyclic manner to obtain acceptable results.

thesis, the algorithm has no knowledge of the future video information. In online tracking, if the jersey number is encountered in the middle of a player track, the identity can't simply be propagated in the previous frames, however, propagating the result is straightforward in offline setting since tracklets identity has to be computed once the whole tracklet has been observed (Fig 7.1).

7.2.4 Propagation of errors in pipeline

The system developed is a pipeline where there is likely that an error in one component affects the other. For instance, in Chapter 6, Section 6.2.3, the failure cases of the overall pipeline included identity switches arising from tracking and incorrect team identification. Such errors may be minimized by a single network for tracking, team identification and player identification such that a single multi-task loss function is optimized for the three problems. A possible formulation of such a system is briefly explained in the next section.

7.3 Future work

The research done in this thesis opens several directions for future work. The important ones are listed in this section:

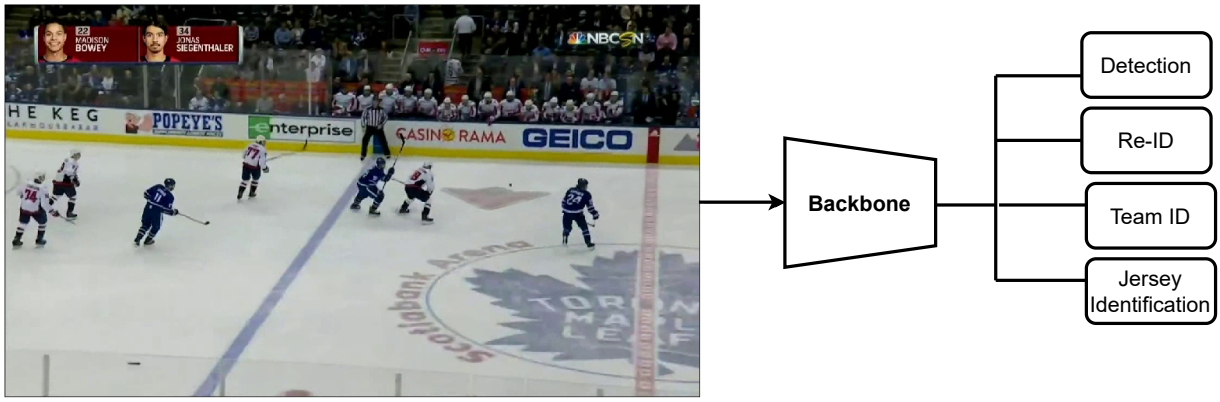


Figure 7.3: Potential architecture for a unified tracking and identification network. Two additional heads for jersey number identification and team identification can be added to a backbone network so that a single multi-task loss function can be optimized for player tracking and identification

7.3.1 Player handedness for player identification

Techniques such as player handedness can help in situations where the jersey number is not visible. Similar to other racquet/stick based sports such as tennis and cricket, an ice hockey player can left-handed, right-handed or ambidextrous. Although unique identification would not be possible **only** on the basis of handedness, however, the search space for identification maybe reduced by using player handedness as a feature for identification. In addition, player handedness features can also be combined with RGB features to improve player identification model accuracy. The challenge with identifying handedness is that the hockey stick blade is often occluded and it is hard to estimate the orientation of the blade with accuracy. Player handedness based features can further be used to recover identities lost due to pan identity switches discussed in Section 7.2.1.

7.3.2 Unsupervised/ semi-supervised learning techniques

In this thesis, the player and team identification networks used are fully supervised. This means that the models are trained and tested on a fixed, annotated train and test set. However, for testing the models on broader data consisting of new seasons and leagues unsupervised/semi-supervised methods may be used to improve generalization. Assuming

that there is availability of large amount of unlabelled data, semi-supervised learning techniques such as pseudo labelling [87, 88, 89] (Fig. 7.2) and active learning [90, 91] techniques can be use to improve the generalization of player identification and team identification models. Additionally fully unsupervised techniques such as clustering can also be used for team identification.

7.3.3 Unified network for tracking and identification

The limitation discussed in Section 7.2.4 may be addressed by using a single network for player tracking, team identification and player identification. One realization of this approach can be the addition of a team and player identification branch into a joint detection and tracking (JDT) based network. Concretely, several JDT approaches [19, 33, 34, 35] have separate detection and re-identification heads for detecting objects and learning re-identification embeddings. Two additional heads for jersey number identification and team identification can be added to the network so that a single multi-task loss function can be optimized for player tracking and identification (Fig. 7.3).

References

[1] G. Braso and L. Leal-Taixe, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] A. Senocak, T.-H. Oh, J. Kim, and I. So Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[3] IIHF, "Survey of Players," 2018. Available online: https://www.iihf.com/en/static/5324/survey-of-players.

[4] A. C. Thomas, "The impact of puck possession and location on ice hockey strategy," *Journal of Quantitative Analysis in Sports*, vol. 2, no. 1, 2006.

[5] K. Vats, M. Fani, P. Walters, D. A. Clausi, and J. Zelek, "Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[6] Y. Luo, O. Schulte, and P. Poupart, "Inverse reinforcement learning for team sports: Valuing actions and players," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 3356–3363, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[7] Sportlogiq Inc. https://sportlogiq.com/en/hockey.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, (Washington, DC, USA), pp. 1440–1448, IEEE Computer Society, 2015.

[9] R. B. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[10] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2015.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), p. 91–99, MIT Press, 2015.

[13] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and real-time tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[14] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *European Conference on Computer Vision (ECCV) Workshops*, 2016.

[15] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 3212–3232, Nov 2019.

[16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1515–1522, Sep. 2009.

[17] D. Mitzel and B. Leibe, "Real-time multi-person tracking with detector assisted structure propagation," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 974–981, Nov 2011.

[18] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 748–756, IEEE, 2018.

[19] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *CVPR*, 2021.

[21] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 4225–4232, AAAI Press, 2017.

[22] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 300–311, Oct 2017.

[23] C. Kim, F. Li, M. Alotaibi, and J. M. Rehg, "Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9548–9557, 2021.

[24] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 14324–14334, IEEE Computer Society, jun 2021.

[25] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: siamese cnn for robust target association," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR). DeepVision: Deep Learning for Computer Vision.*, 2016.

[26] Li Zhang, Yuan Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[27] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3701–3710, July 2017.

[28] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Subgraph decomposition for multi-target tracking," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5033–5041, IEEE, June 2015.

[29] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[30] A. Roshan Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), (Berlin, Heidelberg), pp. 343–356, Springer Berlin Heidelberg, 2012.

[31] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[32] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 146, pp. 2–7, Feb 1999.

[33] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proceedings of the European Conference on Computer Vision*, 2020.

[34] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *The European Conference on Computer Vision (ECCV)*, 2020.

[35] 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*, pp. 1–19, 2021.

[36] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *European Conference on Computer Vision (ECCV)*, 2020.

[37] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019.

[38] W. Jiang, J. C. G. Higuera, B. Angles, W. Sun, M. Javan, and K. M. Yi, "Optimizing through learned errors for accurate sports field registration," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2020.

[39] R. Sanford, S. Gorji, L. G. Hafemann, B. Pourbabaee, and M. Javan, "Group activity detection from trajectory and video data in soccer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[40] N. Mehrasa, Y. Zhong, F. Tung, L. Bornn, G. Mori, and S. Fraser, "Deep learning of player trajectory representations for team activity analysis," 2018.

[41] A. A. Sangüesa, C. Ballester, and G. Haro, "Single-camera basketball tracker through pose and semantic feature fusion," *ArXiv*, vol. abs/1906.02042, 2019.

[42] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.

[43] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognition*, vol. 102, p. 107260, 2020.

[44] R. Theagarajan and B. Bhanu, "An automated system for generating tactical performance statistics for individual soccer players from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 632–646, 2021.

[45] S. Hurault, C. Ballester, and G. Haro, "Self-supervised small soccer player detection and tracking," in *Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports*, MMSports '20, (New York, NY, USA), p. 9–18, Association for Computing Machinery, 2020.

[46] J. Theiner, W. Gritz, E. Müller-Budack, R. Rein, D. Memmert, and R. Ewerth, "Extraction of positional player data from broadcast soccer videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 823–833, January 2022.

[47] C. A. Gadde and C. Jawahar, "Transductive weakly-supervised player detection using soccer broadcast videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 965–974, January 2022.

[48] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Computer Vision - ECCV 2004* (T. Pajdla and J. Matas, eds.), (Berlin, Heidelberg), pp. 28–39, Springer Berlin Heidelberg, 2004.

[49] Y. Cai, N. de Freitas, and J. J. Little, "Robust visual tracking for multiple targets," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 107–118, Springer Berlin Heidelberg, 2006.

[50] Vermaak, Doucet, and Perez, "Maintaining multimodality through mixture tracking," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1110–1116 vol.2, Oct 2003.

[51] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, Dec 2001.

[52] S. Gerke, K. Müller, and R. Schäfer, "Soccer jersey number recognition using convolutional neural networks," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 734–741, 2015.

[53] G. Li, S. Xu, X. Liu, L. Li, and C. Wang, "Jersey number recognition with semi-supervised spatial transformer network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1864–18647, 2018.

[54] H. Liu and B. Bhanu, "Pose-guided R-CNN for jersey number recognition in sports," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2457–2466, 2019.

[55] A. Nady. and E. Hemayed., "Player identification in different sports," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pp. 653–660, INSTICC, SciTePress, 2021.

[56] A. Chan, M. D. Levine, and M. Javan, "Player identification in hockey broadcast videos," *Expert Systems with Applications*, vol. 165, p. 113891, 2021.

[57] S. Gerke, A. Linnemann, and K. Müller, "Soccer player recognition using spatial constellation features and jersey number recognition," *Computer Vision and Image Understanding*, vol. 159, pp. 105 – 115, 2017. Computer Vision in Sports.

[58] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9365–9374, 2019.

[59] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *International Conference on Computer Vision (ICCV)*, 2019.

[60] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004. British Machine Vision Computing 2002.

[61] G. LoweDavid, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[62] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.

[63] M. Fani, P. Walters, D. A. Clausi, J. S. Zelek, and A. Wong, "Localization of ice-rink for broadcast hockey videos," *ArXiv*, vol. abs/2104.10847, 2021.

[64] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *ArXiv*, vol. abs/1603.00831, 2016.

[65] P. Dendorfer, H. Rezatofighi, A. Milan, J. Q. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taix'e, "Mot20: A benchmark for multi object tracking in crowded scenes," *ArXiv*, vol. abs/2003.09003, 2020.

[66] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[67] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[68] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[69] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft *coco*: Common objects in context," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.

[70] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, 01 2008.

[71] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops* (G. Hua and H. Jégou, eds.), (Cham), pp. 17–35, Springer International Publishing, 2016.

[72] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[74] K. Li, S. Wang, X. Zhang, Y. Xu, W. Xu, and Z. Tu, "Pose recognition with cascade transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1944–1953, June 2021.

[75] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[76] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv:1803.01271*, 2018.

[77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[79] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 213–229, Springer International Publishing, 2020.

[80] 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.

[81] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. M. Snoek, "Actor-transformers for group activity recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 836–845, 2020.

[82] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[84] J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *ArXiv*, vol. abs/1607.06450, 2016.

[85] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[86] K. Vats, P. Walters, M. Fani, D. A. Clausi, and J. S. Zelek, "Player tracking and identification in ice hockey," *ArXiv*, vol. abs/2110.03090, 2021.

[87] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[88] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[89] L.-C. Chen, R. G. Lopes, B. Cheng, M. D. Collins, E. D. Cubuk, B. Zoph, H. Adam, and J. Shlens, "Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation," in *Computer Vision – ECCV 2020: 16th Eu-*

ropean Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX, (Berlin, Heidelberg), p. 695–714, Springer-Verlag, 2020.

[90] D. Yoo and I. S. Kweon, "Learning loss for active learning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 93–102, 2019.

[91] R. Caramalau, B. Bhattarai, and T.-K. Kim, "Sequential graph convolutional network for active learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9583–9592, June 2021.

[92] K. Vats, M. Fani, D. A. Clausi, and J. Zelek, "Multi-task learning for jersey number recognition in ice hockey," in *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*, MMSports'21, (New York, NY, USA), p. 11–15, Association for Computing Machinery, 2021.

[93] K. Vats, H. Neher, D. A. Clausi, and J. Zelek, "Two-stream action recognition in ice hockey using player pose sequences and optical flows," in *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 181–188, 2019.

[94] R. Girdhar, J. J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 244–253, IEEE Computer Society, jun 2019.

[95] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6836–6846, October 2021.

[96] W. McNally, K. Vats, T. Pinto, C. Dulhanty, J. McPhee, and A. Wong, "Golfdb: A video database for golf swing sequencing," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2553–2562, 2019.

[97] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[98] Z. Cai, H. Neher, K. Vats, D. A. Clausi, and J. Zelek, "Temporal hockey action recognition via pose and optical flows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[100] H. Pidaparthy and J. H. Elder, "Keep your eye on the puck: Automatic hockey videography," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1636–1644, 2019.

[101] N. Homayounfar, S. Fidler, and R. Urtasun, "Sports field localization via deep structured models," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4012–4020, 2017.

[102] R. A. Sharma, B. Bhat, V. Gandhi, and C. V. Jawahar, "Automated top view registration of broadcast football videos," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 305–313, 2018.

[103] X. Zhang, T. Zhang, Y. Yang, Z. Wang, and G. Wang, "Real-time golf ball detection and tracking based on convolutional neural networks," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2808–2813, 2020.

[104] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6047–6056, 2018.

[105] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Computer Vision – ECCV 2016 Workshops* (G. Hua and H. Jégou, eds.), (Cham), pp. 100–111, Springer International Publishing, 2016.

[106] J. R. Munkres, "Algorithms for the assignment and transportation problems," in *Journal of the Society for Industrial and Applied Mathematics, 5, 32-38.*, 1957.

[107] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai, "Online multi-object tracking with convolutional neural networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 645–649, Sep. 2017.

[108] V. Renò, N. Mosca, R. Marani, M. Nitti, T. D'Orazio, and E. Stella, "Convolutional neural networks based ball detection in tennis games," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1839–18396, 2018.

[109] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games," in *Object recognition supported by user interaction for service robots*, vol. 1, pp. 303–306 vol.1, 2002.

[110] Y. Ariki, Tetsuya Takiguchi, and Kazuki Yano, "Digital camera work for soccer video production with event recognition and accurate ball tracking by switching search method," in *2008 IEEE International Conference on Multimedia and Expo*, pp. 889–892, 2008.

[111] R. Voeikov, N. Falaleev, and R. Baikulov, "Ttnet: Real-time temporal and spatial video analysis of table tennis," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3866–3874, 2020.

[112] N. Ishii, I. Kitahara, Y. Kameda, and Y. Ohta, "3d tracking of a soccer ball using two synchronized cameras," in *Advances in Multimedia Information Processing – PCM 2007* (H. H.-S. Ip, O. C. Au, H. Leung, M.-T. Sun, W.-Y. Ma, and S.-M. Hu, eds.), (Berlin, Heidelberg), pp. 196–205, Springer Berlin Heidelberg, 2007.

[113] X. Wang, V. Ablavsky, H. Ben Shitrit, and P. Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play," *Computer Vision and Image Understanding*, vol. 119, 01 2013.

[114] J. Komorowski, G. Kurzejamski, and G. Sarwas, "Deepball: Deep neural-network ball detector," *ArXiv*, vol. abs/1902.07304, 2019.

[115] M. Yakut and N. Kehtarnavaz, "Ice-hockey puck detection and tracking for video highlighting," *Signal, Image and Video Processing*, vol. 10, 03 2015.

[116] X. Yu, C. . Sim, J. R. Wang, and L. F. Cheong, "A trajectory-based ball detection and tracking algorithm in broadcast tennis video," in *2004 International Conference on Image Processing, 2004. ICIP '04.*, vol. 2, pp. 1049–1052 Vol.2, 2004.

[117] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[118] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *ArXiv*, vol. abs/1609.01775, 2016.

[119] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

APPENDICES

Appendix A

Accuracy metrics for tracking

To evaluate the effectiveness of any MOT algorithm, many metrics have been introduced in the literature [117, 118]. Among these metrics, the most widely used are the Clear MOT metrics [117], and ID-based metrics [118]. The details of these metrics are explained next:

A.1 Clear MOT metrics

Consider a frame t of a video. Let $\{h_0, h_1, ..h_n\}$ be the set of hypotheses for targets $\{o_0, o_1, ..o_n\}$ present in the frame t. Each hypothesis $\{h_i : i \in [0..n]\}$ and target $\{o_i : i \in [0..n]\}$ is represented by bounding boxes. The objective in mind is to develop a set of metrics that:

1. A correspondence between $\{h_0, h_1, ..h_n\}$ and $\{o_0, o_1, ..o_n\}$ needs to be established for every frame t. The correspondence has to be as good as possible.
2. The errors that need to be calculated are :
    - Localization error between the ground truth and detections.
    - **False positives(FP)** - Detections not suitable for being assigned to any target.
    - **False negatives (FN)**- Target location without any hypothesis
    - **Identity switches(IDSW)** - Switching of hypotheses for a given target compared to the previous frames.

Figure A.1: Motivation for ID-based measures. CLEAR MOT metrics charge one identity switch error to (a) and two identity switch errors to (b). However applications such as sports player tracking favour (b) instead of (a) since (b) *maintains* the identity for a longer time.

The correspondence $\{h_i, o_j\}$ in a frame t is labelled as valid if the distance $d_{i,j}$ is less than a threshold T. The distance $d_{i,j}$ is generally defined as the IOU intersection between the hypothesis h_i and the object o_j bounding boxes. The mapping procedure is as follows:

1. Let Ψ_t denote the mappings $\{h_i, o_j\}$ until frame t. If the hypothesis h_i is present in frame $t+1$ along with the object o_j, make the correspondence between h_i and o_j provided that the distance $d_{i,j}$ does not exceed the threshold T.

2. For all the objects o_j for which a correspondence could not be made, use the Hungarian algorithm [106] for making the minimum weight assignment which minimizes the hypothesis distance between the objects and the hypothesis. In case the hypothesis for a particular object changes between Ψ_t and Ψ_{t+1}, it is counted as an identity switch (IDSW).

3. Label the targets with no matches as False negatives (FN), detections without any matched target as False positive (FP).

4. Let c_t denote the number of correspondences(matches) at time t. For these matches calculate the distance(IOU overlap) d_t^i between the target o_i and its corresponding hypothesis.

The multi-object tracking accuracy (MOTA) is then defined as :

$$MOTA = 1 - \frac{\Sigma_t(FN_t + FP_t + IDSW_t)}{\Sigma_t GT_t} \tag{A.1}$$

where FN_t, FP_t, $IDSW_t$ and GT_t denote the false negatives, false positives, identity switches and ground truth objects in frame t respectively. The multi-object tracking precision (MOTP) is defined as:

$$MOTP = \frac{\Sigma_{i,t} d_t^i}{\Sigma_t c_t} \tag{A.2}$$

A.2 Identity preserving metrics

In many real world scenarios such as airport security surveillance and sports tracking, one is more interested in preserving identities of a particular object rather than merely minimizing the number of identity switches. Consider Fig. A.1, let A denote the identity of a sports player who needs to be tracked and let **1** and **2** denote the predicted identities. Since Fig. A.1 (a) has fewer identity switches than A.1 (b), CLEAR MOT metrics report higher MOTA score for (a) than (b), even though (b) provides the position of the player for a longer time and hence, is more likely to be preferred by a sports player tracking system.

To calculate identity preserving metrics, first of all, a bipartite graph $G = (V_t, V_c, E)$ is constructed such that the set V_t has a regular node for each true trajectory and a false positive node for each computed trajectory. The vertex set V_c contains a regular node for each computed trajectory and a false negative mode for each true trajectory. An edge $e \in E$ connects two regular nodes if their trajectories overlap in time. Connections are also made between a regular true node and corresponding false positive node and a regular computed node and corresponding false negative node.

After constructing the graph, appropriate weights are assigned to the graph edges. When one of the nodes corresponding to an edge is an irregular node(false positive or false negative node), then a miss is counted. An edge connecting two regular nodes in sets V_c and V_t is counted as a miss if the overlap between them is less than a threshold Δ. The weight of an edge is defined as the number of binary misses incurred on connecting an edge. A minimum weight matching on the above graph defines a one-to-one mapping minimizing the overall misses. A match between two regular nodes is a True Positive ID (IDTP), a match between a regular and an irregular node is either a False Positive ID (IDFP) or a False Negative ID (IDFN). A match between two irregular nodes is counted as a True Negative ID (IDTN). The ID preserving F1 score (IDF1) is defined as:

$$IDF1 = \frac{2IDTP}{2IDTP + IDFN + IDFP} \tag{A.3}$$

Appendix B

Puck localization

We introduce a network for localizing hockey puck on the ice rink. Remarking that humans can locate the puck position from video with the help of contextual cues and temporal information, our method incorporates temporal information in the form of RGB video and leverages player location information with heatmaps using an attention mechanism to help localize the puck (Fig. B.1). As such, the network developed is tasked with simultaneously (1) localizing the puck in RGB video and (2) learning the homography between the broadcast camera and the static rink coordinate system. Additionally, our method differs from Pidaparthy *et al.* [100] in that instead of annotating data on a frame-by-frame basis, we utilize the existing NHL data available on a play-by-play basis annotated by expert annotators.

Experimental results demonstrate that the network is able to locate the puck with an AUC of 73.1% on the test set. The network is able to localize the puck during player and board occlusions. At test-time, the network is able to perform inference using a sliding window approach in previously unseen untrimmed broadcast hockey video at 5 frame per second (fps).

B.1 Methodology

In this section we discuss the components of the network designed to localize the puck from broadcast video. We also discuss the loss function used along with other parameters used during training.

B.1.1 Network architecture

The overall network architecture consists of four components: Video branch, Player branch, Attention and Output. The architecture is illustrated in Fig. B.1. The next four subsections explain the components in detail.

Video branch

The purpose of the video branch is to obtain relevant spatio-temporal information to estimate puck location. The video branch takes as input 16 frames $\{f_i \in R^{256\times256\times3}, i \in \{1..16\}\}$ sampled from a short video clip V of two second duration. The frames are passed through a backbone network consisting of four layers of R(2+1)D network [119] to obtain features $F_v \in R^{4\times32\times32\times256}$ to be used for further processing. The R(2+1)D network consists of (2+1)D blocks which splits spatio-temporal convolutions into spatial 2D convolutions followed by a temporal 1D convolution.

Player branch

The location of puck on the ice rink is correlated with the location of the players since the puck is expected to be be present where the player "density" is high. We make the assumption that the location of players remains approximately the same in a short two second video clip. In order to encode the spatial player location, we take the middle frame f_m of the video V and pass it through a FasterRCNN [12] network to detect players. After player detection, we draw a Gaussian with a standard deviation of σ_p at the centre of the player bounding boxes to obtain the player location heatmap H. An advantage of using this representation is that the player location variability in the video clip can be expressed through the Gaussian variance. The player location heatmap H is passed through a player location backbone network to output player location features $F_p \in R^{32\times32\times8}$. Please refer to Table B.1 for the exact configuration of the player location backbone. The player location features F_p are passed to the attention block for further processing.

Attention

The purpose of attention is to make the network incorporate player locations by considering the relationship between video features F_v and player location features F_p. The player

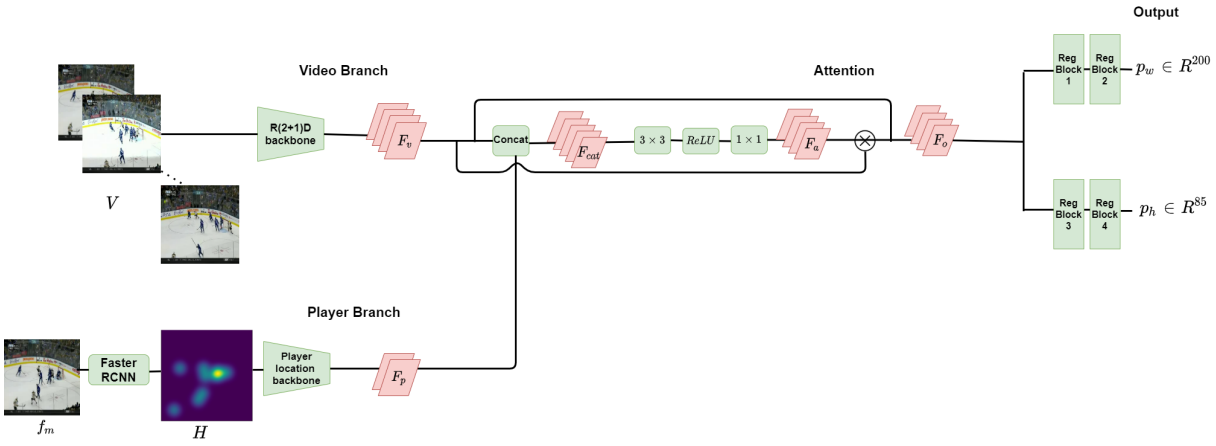


Figure B.1: The overall network architecture. Green represents model layers while pink represents intermediate features. The network consists of four components: (1) Video Branch, (2) Player Branch, (3) Attention, and (4) Output. The Video Branch extracts spatio-temporal features from raw hockey video. The Player Branch extracts play location information from player Gaussian heatmaps. The Attention component fuses the player location and spatio-temporal video information. The Output component produces the puck location output from the features obtained from the attention component.

location features F_p and video features F_v are concatenated along the the channel axis by repeating the player location features along the temporal axis. The concatenated features $F_{cat} \in R^{4\times32\times32\times264}$ are then passed through a variation of the squeeze and excitation [**? ?**] network consisting of a 3×3 convolution, non-linear excitation and 1×1 convolution. The 3×3 squeeze operation learns the spatial relationships between player locations on the rink and video features. The squeeze operation outputs features $F'_{cat} \in R^{4\times32\times32\times132}$. The squeeze operation is followed by non linear activation and 1×1 convolution to obtain features $F_a \in R^{4\times32\times32\times256}$. The 1×1 convolution learns the channel wise relationships between the feature maps in F'_{cat}. Finally, the output of the attention block is the hadamard product of the attention features F_a and the video features F_v followed by a skip connection.

$$F_o = F_a \otimes F_v + F_v \tag{B.1}$$

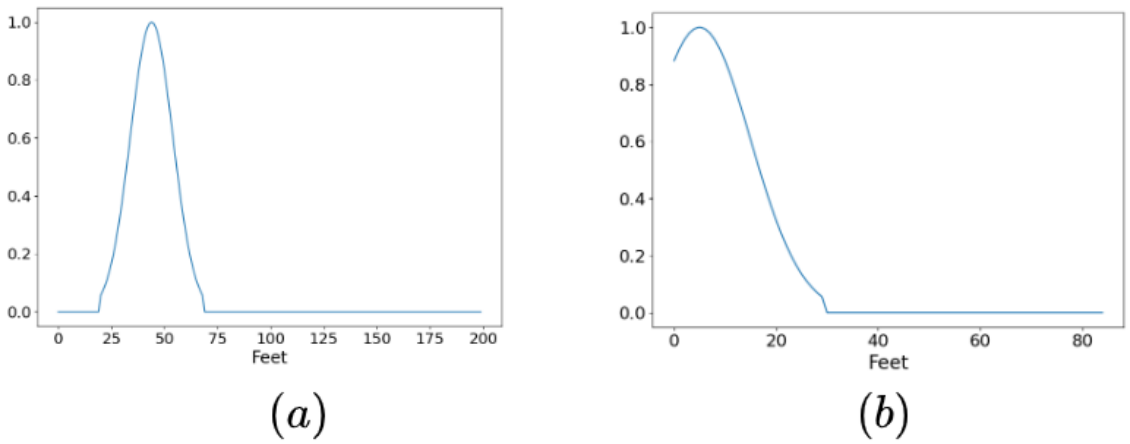


Figure B.2: Construction of ground truth for a training sample with puck located at $w = 44\ ft$ and $h = 5\ ft$. (a) Ground truth distribution vector $w_{gt} \in R^{200}$ (b) Ground truth distribution vector $h_{gt} \in R^{85}$

Output

The features F_o obtained from the attention component are finally passed through two RegBlocks to output the probability of puck location on the ice rink. Global average pooling is done at the end of the two RegBlocks to squash the intermediate output to one dimensional vectors. This is done independently for rink width and height dimensions through two separate branches. The overall network outputs two vectors, $p_w \in R^{200}$ and $p_h \in R^{85}$, in accordance with the dimension of the NHL rink. The exact details of RegBlocks 1 and 2 are given in Table B.2. Regblocks 3 and 4 have a similar architecture, the only difference is that instead of a R^{200} vector p_w, a R^{85} vector p_h is output by changing the output channels to 85.

B.1.2 Training details

We use the cross entropy loss to train the network. In order to create the ground truth, we use a one dimensional Gaussian with mean at the ground truth puck location and a standard deviation σ for both directions. The Gaussian variance encodes the variability in ball location in the short video clip (Fig. B.2) . The total loss L_{puck} is the sum of the loss

Table B.1: Network architecture of player location backbone. k,s and p denote kernel dimension, stride and padding respectively. Ch_i, Ch_o and b denote the number of channels going into and out of a block and batch size respectively.Additionally each layer contained a residual-skip connection with a 1×1 convolution.

Input: Player heatmap $b \times 256 \times 256$
Layer 1
Conv2D
$Ch_i = 1, Ch_o = 2$
(k = 3×3, s = 2, p = 1)
Batch Norm 2D
ReLU
Layer 2
Conv2D
$Ch_i = 2, Ch_o = 4$
(k = 2×2, s = 2, p = 0)
Batch Norm 2D
ReLU
Layer 3
Conv2D
$Ch_i = 4, Ch_o = 8$
(k = 2×2, s = 2, p = 0)
Batch Norm 2D
ReLU
Output $b \times 32 \times 32 \times 8$

in horizontal axis L_w and vertical axis L_h, which is given by:

$$L_{puck} = L_w + L_h \tag{B.2}$$

$$L_{puck} = -\frac{1}{200}\sum_{i=1}^{200} w_{gt} \log p_w - \frac{1}{85}\sum_{j=1}^{85} h_{gt} \log p_h \tag{B.3}$$

Where $w_{gt} \in R^{200}$ and $h_{gt} \in R^{85}$ denote the ground truth probabilities and $p_w \in R_{200}$ and $p_h \in R^{85}$ denote the predicted probabilities.
For data augmentation, each frame is sampled from a uniform distribution $U(0, 60)$ so that the network sees different frames of the same video when the video sampled different times. The data augmentation technique is used is all experiments unless stated otherwise. We use the Adam optimizer with an initial learning rate of .0001 such that the learning rate is reduced by a factor of $\frac{1}{5}$ at iteration number 5000. The batch size is 15.

Table B.2: Network architecture of Regblocks 1 and 2 for output $p_w \in R^{200}$. k,s and p denote kernel dimension, stride and padding respectively. Ch_i, Ch_o and b denote the number of channels going into and out of a block and batch size respectively. Additionally each layer contained a residual-skip connection with a $1 \times 1 \times 1$ convolution.

Input: F_0 $b \times 4 \times 32 \times 32 \times 256$
Reg Block 1
Conv3D
$Ch_i = 256, Ch_o = 200$
(k $= 2 \times 2 \times 2$, s $= 2 \times 2 \times 2$, p $= 0$)
Batch Norm 3D
ReLU
Reg Block 2
Conv3D
$Ch_i = 200, Ch_o = 200$
(k $= 2 \times 2 \times 2$, s $= 2 \times 2 \times 2$, p $= 0$)
Batch Norm 3D
ReLU
Global average pooling
Sigmoid activation
Output $b \times 200$

B.2 Experiments

In this section we describe the dataset created to train and evaluate the puck tracking network. We also discuss the accuracy metric used and the results obtained. Finally be perform an ablation study the study the influence of various factors on the network performance.

B.2.1 Dataset

The dataset consists 8,987 broadcast NHL videos of two second duration with a resolution of 1280 $\times$ 720 pixels and a framerate of 30 fps with the approximate puck location on the ice rink annotated. The annotations are rough and approximate such that the puck location corresponds to the whole two second video clip rather than a particular frame. The videos are split into 80% samples for training and 10% samples each for validation and testing. Fig B.3 shows the distribution of a subset of puck location data. The videos are also annotated with an event label which can be either Faceoff, Advance (dump in/out), Play (player moving the puck with an intended recipient e.g., pass, stickhandle) or Shot.

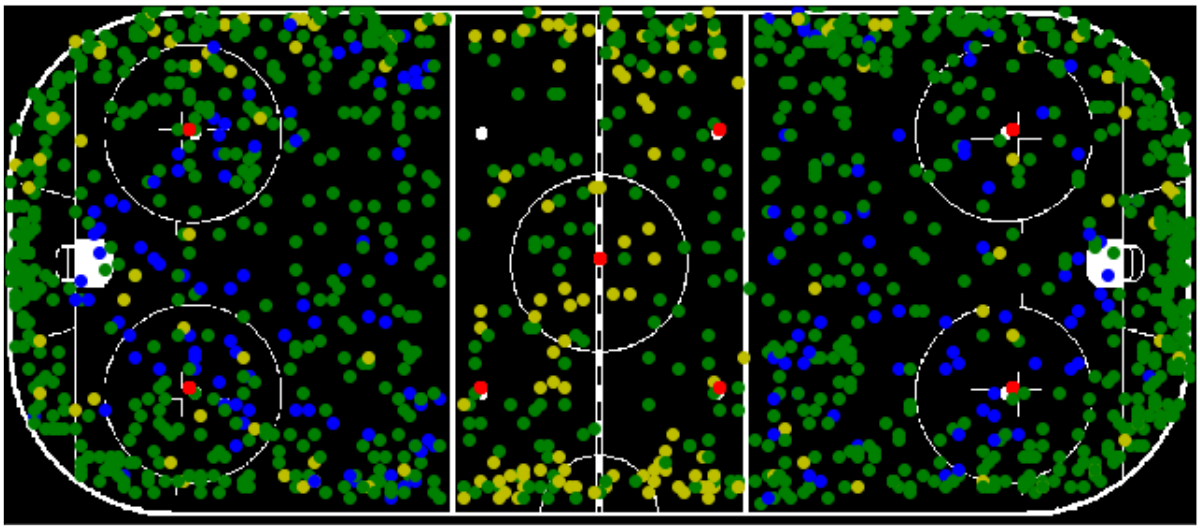


Figure B.3: Subset of 1500 puck locations in the dataset. The puck locations on the ice rink are highly correlated with the event label. Faceoffs(red) are located at the faceoff circles, shots(blue) are located in the offensive zones and dump in/outs (yellow) are presents in the neutral zone.

B.2.2 Accuracy metric

A test video is considered to be correctly predicted at a tolerance t feet if the distance between the ground truth puck location z and predicted puck location z_p is less than t feet. That is $||z - z_p||_2 < t$. Let $\phi(t)$ denote the percentage of examples in the test set with correctly predicted position puck position at a tolerance of t. We define the accuracy metric as the area under the curve (AUC) $\phi(t)$ at tolerance of $t = 5$ feet to $t = 50$ feet.

B.2.3 Results - trimmed video clips

The network attains an AUC of 73.1% on the test dataset illustrated in Fig. B.5 (b). The AUC in the horizontal direction is 81.4% and AUC in vertical direction is 87.8%. From Fig. B.5 (a), at a low tolerance of $t = 12\,ft$, the accuracy in vertical(Y) direction is 76% and the accuracy in horizontal(X) direction is 63%. At a tolerance of $t = 20\,ft$, the accuracy in both directions is greater than 80% .

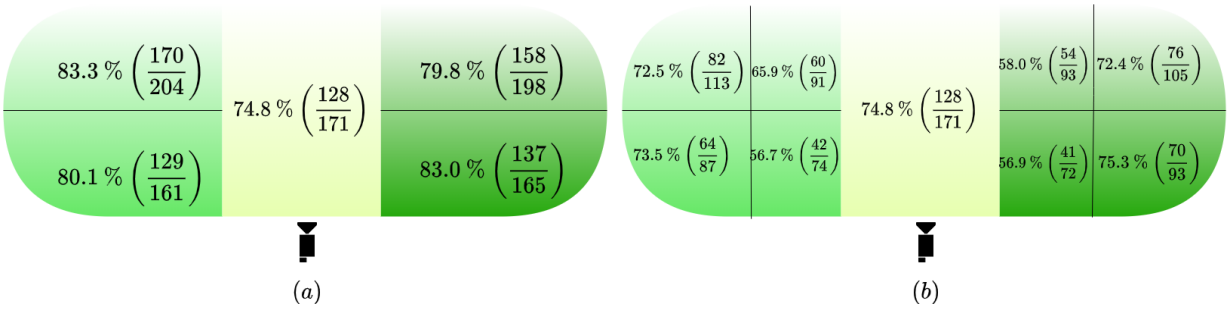


Figure B.4: Zone-wise accuracy. The figure represents the hockey rink with the text in each zone represents the percentage of test examples predicted correctly in that zone. The position of the camera is at the bottom. In (b), the accuracy is low in the lower halves of the defensive and offensive zones since the puck gets occluded by the rink board.

Fig. B.4 show the zone wise accuracy. A test example is classified correctly if the predicted and ground truth puck location lies in the same zone. From Fig. B.4 (a), the network gets an accuracy of $\sim 80\%$ percent in the upper and lower halves of the offensive and defensive zones. From Fig. B.4 (b), after further splitting the ice rink in nine zones, the network achieves an accuracy of more than 70% in five zones. The network also has failure cases. From Fig. B.4 (b), it can be seen that accuracy is low (less than 60%) in the bottom halves of the defensive and offensive zones. This is due to the puck being occluded by the rink boards.

B.2.4 Results- untrimmed broadcast video

We also test the network on untrimmed broadcast videos using a sliding window of length l and stride s. The window length l is the time duration covered by the sliding window and stride s is the time difference between two consecutive application of the sliding window. Due to the difficulty of annotating puck location frame-by-frame in $720p$ videos, we do not possess the frame-by-frame ground truth puck location. Therefore, we perform a qualitative analysis in this section. The videos used for testing are previously unseen video not present in the dataset used for training and testing the network.

To determine the optimal values of stride s validation is performed on a 10 second clip. Some frames from the validation 10 second clip are shown in Fig. B.6. Whenever visible,

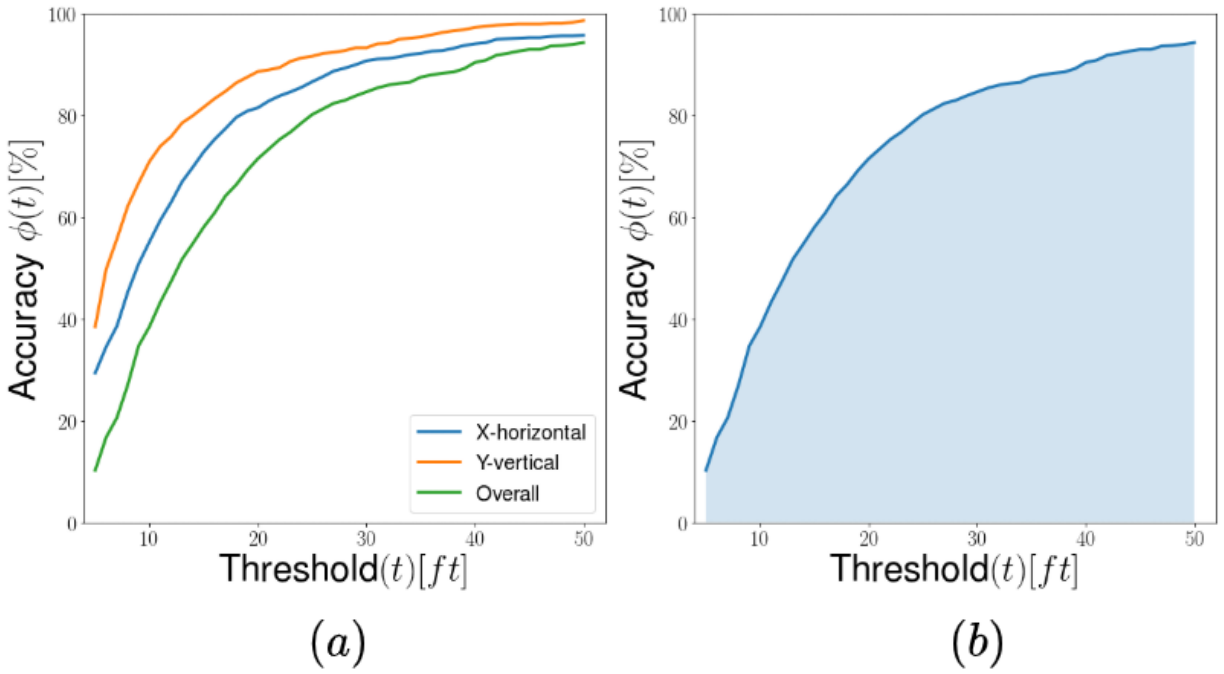


Figure B.5: (a) Accuracy (ϕ) vs threshold (t) curve. (b) The best performing model gets an overall AUC of 73.1% on test set.

the location of the puck is highlighted using a red circle. Fig B.7 (a) shows the trajectories obtained. The network is able to approximately localize the puck in untrimmed video within acceptable visual errors, even though the network is trained on trimmed video clips where puck location is annotated approximately. The puck is not visible during many frames of the video, but the network is still able to guess the puck location. This is because the network takes into account the temporal context and player location. Since the network is originally trained on 2 second clips, the window length l is fixed to $2s$. Fig B.7 (a) , shows that as the stride s is decreased, the puck location estimates become noisy. Since between two passes, the puck motion is linear, we do not decrease stride below $0.5s$ as it leads to very noisy estimates (Fig. B.7 (b)). The optimal stride $s = 1s$ gives the most accurate result. A lower stride results in noisy results and higher strides produces very simple predictions. The inference time of the network on a single GTX 1080Ti GPU with 12GB memory is 5 fps.

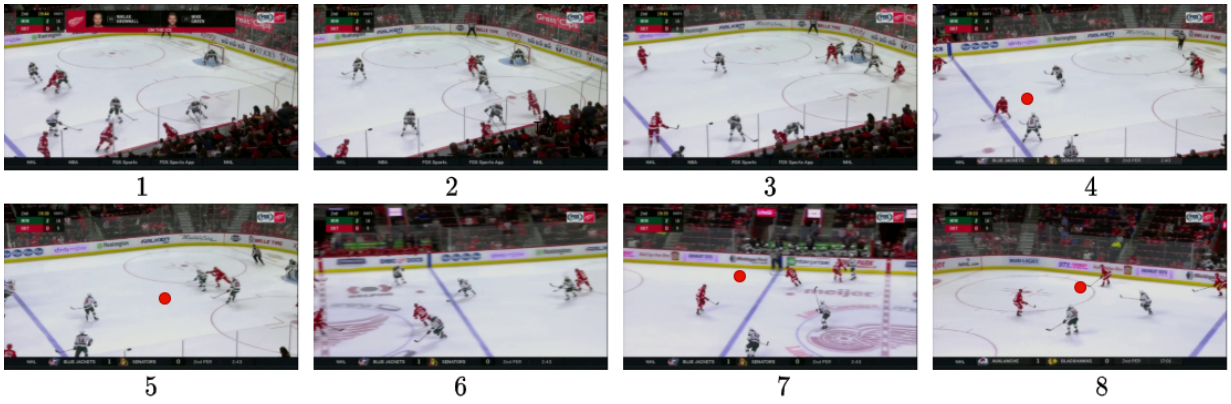


Figure B.6: Some frames from the 10 second validation video clip. Whenever visible, the location of the puck is highlighted using the red circle. The initial portion of the clip is challenging since the puck is not visible in the initial part of the clip.

B.2.5 Ablation studies

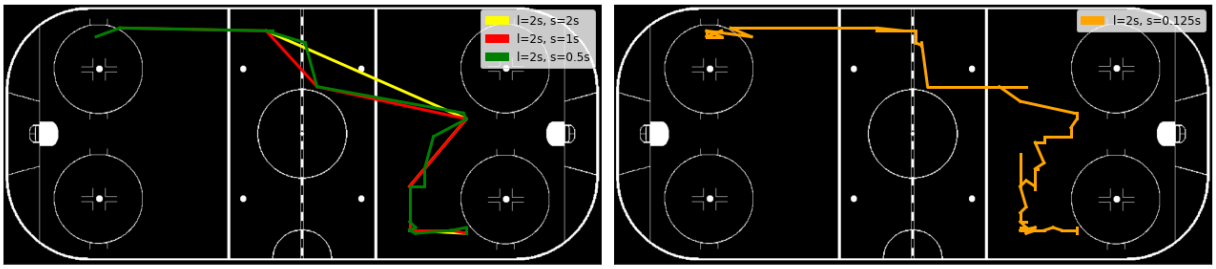
We perform an ablation study on the number of layers in the backbone network, puck ground truth standard deviation, presence/absence of player branch consisting of player locations and data augmentation .

Puck ground truth standard deviation

The best value of standard deviation σ of puck location ground truth 1D Gaussian is determined by varying σ from 20 to 35 in multiples of five. From Table B.3, the number of layers in the backbone is fixed to three while player location based attention is not used. Maximum AUC of 69% is attained with $\sigma = 30$ feet. A lower value of σ makes the ground truth Gaussian more rigid/peaked which makes learning difficult. A value of sigma greater than 30 lowers accuracy since a higher σ makes the ground truth more spread out which reduces accuracy on lower tolerance values.

Layers in backbone

We determine the optimal number of layers in the R(2+1)D backbone network by extracting the video branch features from different layers without using the player location based attention. The puck ground truth standard deviation is set to the optimal value of 30.



(a) (b)

Figure B.7: (a) Puck trajectory on the ice rink for the validation video. The trajectory becomes noisy with $s = 0.5s$ and lower. (b) Puck trajectory for the validation video with a very low stride of 0.125 seconds. The trajectory is extremely noisy and hence is not a good estimate.

Table B.3: Comparison of AUC with different values of σ with a three layer backbone network. Network with $\sigma = 30$ shows the best performance

σ	AUC	AUC(X)	AUC(Y)
20	62.5	71.3	85.07
25	68.5	77.9	85.6
30	**69.0**	78.5	85.5
35	68.9	78.8	85.4

Table B.4: Comparison of AUC with different number of layers of the backbone R(2+1)D network. A four layer backbone shows the best performance.

Layers	AUC	AUC(X)	AUC(Y)
2	56.3	73.2	74.1
3	69.0	78.5	85.5
4	**72.5**	81.3	87.3
5	72.4	81.0	87.3

From Table B.4, the maximum AUC of 72.5% is achieved by using 4 layers of R(2+1)D network. Further increasing the number of backbone layers to 5 causes a decrease of 0.1 in AUC due to overfitting.

Table B.5: Comparison of AUC values with/without player branch. The player branch with $\sigma_p = 15$ shows the best performance.

Player detection	σ_p	AUC	AUC(X)	AUC(Y)
No	-	72.5	81.3	87.3
Yes	15	**73.1**	81.4	87.8
Yes	20	72.8	81.5	87.3
Yes	25	72.2	80.4	87.9

Table B.6: Comparison of AUC values with uniform and random sampling

Sampling method	AUC	AUC(X)	AUC(Y)
Constant interval	70.3	79.4	86.4
Random	**73.1**	81.4	87.8

Player location based attention

We add the player branch and the attention mechanism to the network with 4 backbone layers and $\sigma = 30$. Three values of player location standard deviation $\sigma_p = \{15, 20, 25\}$ are tested. From Table B.5, adding the player location based attention mechanism brought an improvement in the overall AUC by 0.6% with $\sigma_p = 15$. Further increasing σ_p causes the player location heatmap to become more spread out obfuscating player location information.

Data augmentation

We compare the data augmentation technique done using randomly sampling frames from a uniform distribution (explained in Section B.1.2) to sampling frames at a constant interval. From Table B.6, removing random sampling decreases the overall AUC by 3.2% which demonstrates the advantage of the data augmentation technique used.

B.3 Summary

We introduced a network to localize puck in broadcast hockey video. The model makes use of temporal information and incorporated player locations trough an attention mechanism to localize puck. We perform ablation studies on the network parameters and data

augmentation used. We attain an AUC of 73.1% on the test set and qualitatively localize the puck in untrimmed broadcast videos. We also report an ice rink region based average accuracy of 80.2% with the ice rink split into five zones and 67.3% with the rink split into nine regions.