Leveraging Player Tracking for Event Detection in Ice Hockey

Ken M. Nsiempba[1]*, Amir Nazemi[1],
William Jiang[2], John Zelek[1], David Clausi[1]
[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
[2]University of California, Los Angeles
{kmnsiemp, amir.nazemi, jzelek, dclausi}@uwaterloo.ca
{jiangwil}@g.ucla.edu

Abstract

Faceoffs are pivotal events in hockey, marking strategic resets in gameplay that influence team positioning and puck possession. Detecting these events in video data can provide valuable insights for coaches and analysts, enabling the study of player formations and strategies around these critical moments. This work presents a novel framework for detecting ice hockey faceoffs. Our approach processes overhead video sequences using a multi-stage pipeline, incorporating object detection and segmentation to track player trajectories across frames. We employ state of the art detectors and tracking tools, enabling tracking and trajectory analysis for each player. Additionally, preprocessed sequences are used to better ensure accurate player tracking. We demonstrate the framework's effectiveness in automatically identifying faceoffs, with promising results that suggest its potential for broader applications in sports analytics. By enhancing the visibility of faceoffs and player interactions in hockey, this work contributes toward automated sports analytics, providing a robust tool for studying patterns and tactics in high-paced, dynamic sports environments.

Introduction

Detecting events in hockey is crucial for analyzing and improving team strategies, player performance, and overall game understanding. In this article, events are defined by the simultaneous actions of multiple players at a specific instant, focusing on coordinated team activities and formations. Key events like faceoffs, passes, goals, and turnovers often indicate shifts in game dynamics and can reveal patterns that impact the game's outcome. For coaches and analysts, identifying these events enables targeted feedback to players, supports strategic decision-making, and enhances preparation for future games. On a broader scale, event detection in hockey is also valuable in sports analytics research, aiding in the development of automated systems that make sports analysis more efficient and objective.

Ice hockey is characterized by its exceptionally fast pace, with players moving rapidly across the ice, creating motion blur that can obscure important details. This restricts our ability to fully analyze the spatial and temporal relationships among players, an essential factor in understanding team dynamics, assessing player performance, and uncovering tactical patterns. These limitations have traditionally made it difficult to consistently and accurately capture the information needed to rate individual players, identify play patterns, or offer coaching insights.

This article proposes a solution to the challenging task of faceoff detection as an important event in ice hockey games.

Faceoffs in ice hockey are critical points where gameplay is reset, either at the start of a period, after a goal, or following a stoppage in play, typically signaled by a referee's whistle. Faceoffs not only impact puck possession but also reveal underlying strategies, as teams often employ particular formations or player roles depending on the location of the faceoff and the game's context. Recognizing and analyzing faceoffs in video footage can provide insights into team strategies, player positioning, and success rates, adding valuable information to the coaching process.

The main contribution of this paper is its temporal feature structure design, which includes first- and second-order positional information about players on the ice. Our framework's feature design draws inspiration from our prior knowledge of faceoff events. Faceoff events are usually more stable and structural. Thus,

*website: https://botengu.github.io/portfolio/

Figure 1: A video frame from a publicly available overhead view ice hockey game video [1] with players' trajectories and bounding boxes.

player position on the field and player speed could be a discriminative feature for those events. The framework presented here leverages state-of-the-art computer vision tools in order to properly extract the players' locations analyze the trajectories of the players.

In the following sections, we review prior efforts to automate hockey event detection and explain how our approach builds upon and diverges from these methods. We go over the concepts that we have used for our methods and then we detail the implementation in the section . We then present results, showcasing the effectiveness of this framework in identifying faceoffs, and discuss potential implications for broader hockey analytics applications.

Related Works

The authors in [2] developed a pipeline for event detection in sports videos that are only coarsely annotated. Their method involves inputting frames into a CNN to extract spatial features, which are then passed through multiple temporal CNN towers with varying receptive fields to capture events over different time scales. A Softmax layer is applied to obtain event probabilities. Unlike our approach, this method does not utilize tracklets, and it was tested on a different dataset.

In [3], the authors created a ResNet-based model focused on classifying sports images. Their work is strictly spatial, without any temporal aspect, as it does not consider sequential information across frames.

The authors of [4] aimed to improve event-based object detection by selectively processing only the most important data, which reduces computational costs and improves efficiency. Their work focuses on scene-adaptive methods to optimize object detection for high-dynamic events.

The pipeline in [5] targets action spotting using an ac-

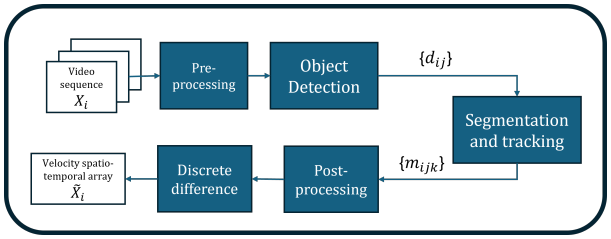

Figure 2: The pipeline which helps us extract spatio-temporal arrays from video sequences

tive learning approach. It includes a prediction function and a clip selection function, which are based on "uncertainty" and "entropy" measures to prioritize annotation. Starting with a small labeled dataset, they iteratively apply the clip selection function to identify the most informative clips for further annotation, thus optimizing the labeling process.

In [6], the authors implemented a player tracking system using the YOLOv8 model, allowing for accurate and efficient tracking of players in sports footage.

The authors in [7] worked on action recognition by using optical flow to capture motion information, providing additional temporal context to improve the recognition of sports actions.

Finally, in [8], the authors leveraged puck localization as a basis for event recognition in hockey. Their method combines video features and player position heatmaps to infer the puck's location, allowing for more accurate identification of events based on puck movement and positioning.

Method

In this section, we outline the methodology used to extract and classify spatio-temporal arrays from video sequences in the context of faceoff and whistle event detection in hockey games. We start by annotating the relevant timestamps in the video, followed by the detection and tracking of players using advanced computer vision techniques. These techniques allow us to capture the dynamic movement of players and segment them accurately. We then proceed to classify the extracted spatio-temporal features using machine learning algorithms to predict event types. The pipeline is designed to facilitate the efficient processing of large video datasets while maintaining high accuracy in detecting events.

Data Labeling

For every video, we have annotated the different timestamps of the video where a faceoff or a whistle would take place. The whistle event can be part of the neg-

ative class because, although it may indicate a stoppage, it is not specifically a faceoff, which typically involves players readying for puck possession at designated spots. Including whistles as a negative class helps the model learn to differentiate faceoffs from other whistle-stopped events, reducing false positives by clearly distinguishing faceoff events from similar stoppage signals in the game. This distinction is essential for accurate faceoff detection and minimizes misclassification in real-world applications.

In our frame work:

- $T = \{t_1, t_2, ..., t_n\}$ is the set of annotated timestamps (specific moments in the video where clips are extracted).
- Δt represents the size of the temporal window.
- X_i is the video sequence going from

$$[t_i - \Delta t/2, t_i + \Delta t/2]$$

- $Y_i \in \{0, 1\}$ is the label (event type) corresponding to clip i, where 0 might represent one event type (e.g., "faceoff") and 1 represents the other event type (e.g., "whistle").

Detection and Tracking

The core of our framework lies in transforming the clip X_i into spatio-temporal arrays $\tilde{X}_i$ (see figure 2). To this extent, we used the object detection tool [9] to detect the players which uses a pre-trained Detection Transformer (DETR) model with a ResNet-50 backbone. The detections d_j are then used as prompts for the input tool Segment Anything Model 2 (SAM2) [10]. SAM2 segments and tracks the objects by refining the mask iteratively, enabling more precise boundaries; which makes it robust to occlusions compared to traditional bounding boxes.

- For sample X_i, using DETR, we can get the detections $\{d_{ij} \mid 0 \leq i \leq n_{samples}, 0 \leq j \leq n_{players}\}$ with $n_{players}$ being the number of players detected (highly dependent on the clarity of the image).
- For each detection, we get a tracklet which is a mask propagating throughout the frames of the sequence. For clip X_i we get the tracklets $\{m_{ijk} \mid i, j, k \in R, 0 \leq i \leq n_{samples}, 0 \leq j \leq n_{players}, 0 \leq l \leq k \leq n_{frames}\}$.
- After a few post-processing steps (padding, data augmentation and discrete differences), we can obtain the velocity spatio-temporal array $\tilde{X}_i$ from the initial clip X_i. Before the discrete difference, we just have a position spatio-temporal array.

We obtain the dataset D, such that $D = \{(\tilde{X}_i, Y_i) \mid 0 \leq i \leq n_{samples}\}$

K-nearest-neighbours

In order to classify the extracted features arrays from the game, K-nearest-neighbours can be used. Given a dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in R^d$ are the feature vectors and $y_i \in R$ are the corresponding labels, the KNN algorithm for a query point $q \in R^d$ is defined as follows:

1. Compute the distance between the query point q and all points in the dataset. Common distance metrics include Euclidean distance:

$$d(x_i, q) = \sqrt{\sum_{k=1}^{d} (x_{i,k} - q_k)^2}$$

2. Sort the distances and select the k-nearest neighbors, i.e., the k smallest values of $d(x_i, q)$.

3. For classification tasks, the predicted label $\hat{y}_q$ for the query point q is determined by the majority vote of the k nearest neighbors:

$$\hat{y}_q = \text{mode}(y_{i_1}, y_{i_2}, \ldots, y_{i_k})$$

4. For regression tasks, the predicted value $\hat{y}_q$ is the average of the labels of the k nearest neighbors:

$$\hat{y}_q = \frac{1}{k} \sum_{j=1}^{k} y_{i_j}$$

where $\{i_1, i_2, \ldots, i_k\}$ are the indices of the k nearest neighbors based on the computed distance.

Our methodology combines advanced object detection and segmentation techniques to create accurate spatio-temporal representations of hockey events. By integrating tools like DETR for player detection and SAM2 for precise tracking, we are able to handle occlusions and noise effectively. The K-nearest-neighbors algorithm serves as a powerful classifier for the labeled spatio-temporal arrays, ensuring reliable event detection. This approach provides a robust framework for automating the analysis of hockey video sequences, allowing for accurate recognition and classification of critical in-game events like faceoffs and whistles.

Experimental Details

The data we had consisted of 24 overhead-view videos provided by our collaborators at Stathletes. They corresponded to multiple 10 minutes video of Ice hockey games at 30 frames per seconds. Each video was

recorded with a fixed camera but the position of the camera changed based on the rink. Each frame was first converted from its original BGR color format to a grayscale image. To minimize noise and smooth the image, a median filter with a kernel size of 31 was applied. This step reduces minor variations and preserves edge integrity, facilitating more accurate contour detection in subsequent steps. The grayscale image was then thresholded with a binary thresholding method. Pixels with intensities above 128 were set to 255 and those below were set to 0. This transformation accentuates the shape of potential objects, simplifying the contour detection process.

We also had to do some cropping to remove the crowd from the view such that focus could solely be on the rink; this would increase the proper detection and tracking of the players.

To isolate the rink, the largest contour was selected based on contour area. The assumption is that the largest contour in the frame represents the most relevant object or feature. A bounding rectangle was generated around the largest contour yielding the coordinates (x, y) of the top-left corner, along with the width (w) and height (h) of the rectangle. This bounding box enables straightforward localization of the rink.

For our DETR model, we filtered detected objects by applying a threshold of 0.95 on the confidence scores to retain only high-confidence detections. The bounding boxes of these selected objects are then passed to the subsequent segmentation stage.

From the initial set of 190 samples, only 94 valid samples remained after processing them through the pipeline.

Here is a breakdown of our approach:

- The dataset, consisting of feature samples (samples) and corresponding labels (labels), is first divided into training data (70%) and a temporary set (30%). The temporary set is further split equally into validation and test sets (15% each of the original dataset). This three-way split ensures that the training data is used for model fitting, the validation set helps with hyperparameter tuning, and the test set provides a final, unbiased evaluation of the model.

- To optimize the KNN model, we use a grid search over a range of possible hyperparameters (specifically, the number of neighbors, k). A KNN classifier is instantiated. A parameter grid is defined to test values of k ranging from 1 to 20. The GridSearchCV function is employed to perform a 5-fold cross-validation for each value of k, using accuracy as the scoring metric. The best value of k is identified based on the cross-validation results, and the corresponding model is retrieved.

- Validation Set Evaluation After determining the optimal number of neighbors (k), the best-performing model from the grid search is evaluated on the validation set: Predictions are made on the validation data using the best KNN model.

- Test Set Evaluation Finally, the model's generalization capability is assessed on the test set: Predictions are made using the optimized KNN model. The test set accuracy score and a detailed classification report, including precision, recall, and F1 scores, are generated and can be shown in Table 1.

Results

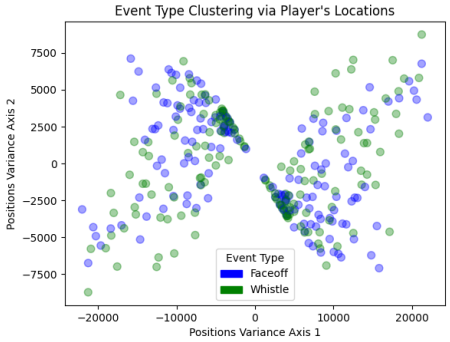

Figure 3: Principal component analysis embeddings of the different location arrays

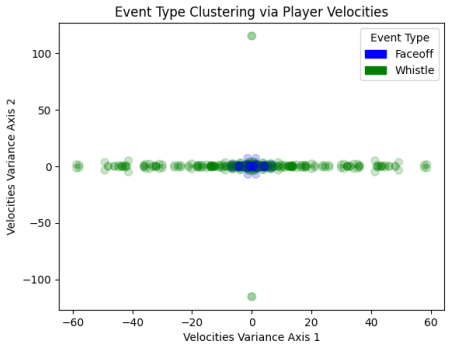

Figure 4: Principal component analysis embeddings of the different velocities arrays

Figure 3 and 4 show the principal component analysis (PCA) embeddings for the two different feature choices; while using trajectory arrays leads to indistinguishable embeddings, velocity arrays lead to embeddings which are more revealing. Faceoff embeddings are tightly clustered near the origin, reflecting minimal velocity variance due to controlled positioning before the puck drop.

In contrast, whistle event embeddings are more dispersed, indicating greater variability in player movements across different gameplay scenarios. This distinction suggests that faceoffs have a unique velocity pattern, which could aid automatic event classification in hockey, while whistles could serve as a negative class for distinguishing faceoffs. The tight clustering of faceoffs indicates they have a unique velocity pattern, potentially useful for automatic event classification in hockey. The wider spread of whistle events highlights the variability in player movements around stoppages not related to faceoffs, reinforcing the idea that whistles could serve as a negative class for identifying faceoffs.

Table 1: Test Set Metrics for trajectory features

Class	Precision	Recall	F1-Score	Support
faceoff	0.62	0.53	0.57	53
whistle	0.49	0.59	0.53	41
Accuracy			0.55	
Macro Avg	0.56	0.56	0.55	94
Weighted Avg	0.56	0.55	0.55	94

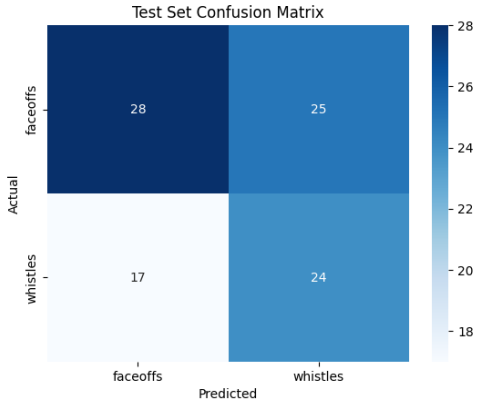

Figure 5: Confusion matrix for set with trajectory features

Table 2: Test Set Metrics for velocity features

Class	**Precision**	**Recall**	**F1-Score**	**Support**
faceoff	0.82	0.92	0.87	53
whistle	0.88	0.73	0.80	41
Accuracy			0.84	
Macro Avg	0.85	0.83	0.83	94
Weighted Avg	0.85	0.84	0.84	94

Table 1 compares the classification accuracy achieved using two different types of arrays: Position and Velocity. The accuracy was obtained as the number of correct prediction over the number of predictions in total.

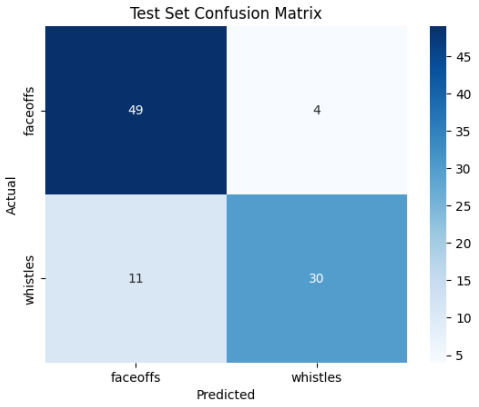

Figure 6: Confusion matrix for set with velocity features

The location array yields an accuracy of 0.55 with a k parameter of 9, suggesting that using only position data provides mediocre predictive capability. In contrast, the velocity array achieves a significantly higher accuracy of 0.84 with k=1, indicating that velocity data is more effective for this classification task. This improvement suggests that velocity captures more informative dynamics relevant to event detection, potentially due to its ability to reflect player movement patterns that are less apparent when only positional data is considered.

The test set confusion matrix in figure 6 confirms that more whistles are taken whistles (False Positives) are predicted to be faceoffs that the reverse, this corroborates the embeddings shown in Figure 4. The variance of the whistles embeddings is greater than the one of the faceoff, there are more change for some whistles to be confused with faceoffs near the center of the plot (where the faceoffs embeddings are concentrated).

Conclusion

In this paper, we introduced a novel framework for automatically detecting faceoff events in ice hockey videos using computer vision and player tracking techniques. By leveraging object detection and segmentation models, we were able to capture and analyze player trajectories and velocity patterns to differentiate between faceoffs and other whistle-stopped events. Our results highlight the effectiveness of using velocity data over positional data for faceoff detection, as it captures the subtle movement dynamics that characterize these events.

In future work, we aim to incorporate homography transformations to obtain real positional data that accounts for the perspective distortion in overhead video footage. This will provide more accurate spatial representations of player positions on the rink. Additionally, we plan to explore the use of broadcasted views in our

analysis, rather than relying solely on overhead views, as broadcasted views tend to be the most abondant types of views. These enhancements will contribute to creating a more robust, adaptable tool for sports analytics in ice hockey.

We can also make this study more robust by quantifying the detection rate, and mask propagation rate to quantify the validity of our approach. Furthermore, we also intent on aggregating images embeddings with tracking data embeddings to increase the likelihood of detecting events.

Acknowledgements

We would like to acknowledge Stathletes for their support and the footage provided for these experiments and we would also like to acknowledge the IBET fellowship for their support.

References

[1] A. Software, "hockey player tracking," 2018, [Accessed: 2024-11-08]. [Online]. Available: https://www.youtube.com/watch?v=zI_DPGOafKY

[2] 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," 2020, pp. 882–883. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2020/html/w53/Vats_Event_Detection_in_Coarsely_Annotated_Sports_Videos_via_Parallel_Multi-Receptive_CVPRW_2020_paper.html

[3] K. Singh Gill, V. Anand, S. Malhotra, and S. Devliyal, "Sports Game Classification and Detection Using ResNet50 Model Through Machine Learning Techniques Using Artificial Intelligence," in *2024 3rd International Conference for Innovation in Technology (INOCON)*, Mar. 2024, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10511858

[4] Y. Peng, H. Li, Y. Zhang, X. Sun, and F. Wu, "Scene Adaptive Sparse Transformer for Event-based Object Detection," Apr. 2024, arXiv:2404.01882 [cs]. [Online]. Available: http://arxiv.org/abs/2404.01882

[5] S. Giancola, A. Cioppa, J. Georgieva, J. Billingham, A. Serner, K. Peek, B. Ghanem, and M. Van Droogenbroeck, "Towards Active Learning for Action Spotting in Association Football Videos," 2023, pp. 5098–5108. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023W/CVSports/html/Giancola_Towards_Active_Learning_for_Action_Spotting_in_Association_Football_Videos_CVPRW_2023_paper.html

[6] A. Katić, V. Matić, and V. Papić, "Detection and Player Tracking on Videos from SoccerTrack Dataset," in *2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH)*, Mar. 2024, pp. 1–6, iSSN: 2767-9470. [Online]. Available: https://ieeexplore.ieee.org/document/10495998/?arnumber=10495998

[7] M. Cao, M. Yang, G. Zhang, X. Li, Y. Wu, G. Wu, and L. Wang, "SpotFormer: A Transformer-based Framework for Precise Soccer Action Spotting," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2022, pp. 1–6, iSSN: 2473-3628. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9948888

[8] K. Vats, M. Fani, D. A. Clausi, and J. Zelek, "Puck Localization and Multi-Task Event Recognition in Broadcast Hockey Videos," 2021, pp. 4567–4575. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021W/CVSports/html/Vats_Puck_Localization_and_Multi-Task_Event_Recognition_in_Broadcast_Hockey_Videos_CVPRW_2021_paper.html

[9] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[10] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.