Real-Time Object Tracking in Sports and Ice Hockey Videos Using a Fine-Tuned YOLOv8 Deep Learning Model

*COS791 Semester Project

1st Reuben Jooste (u21457060)
Computer Science Dept.
University of Pretoria
Pretoria, South Africa
u21457060@tuks.co.za

2nd Arno Jooste (u21457451)
Computer Science Dept.
University of Pretoria
Pretoria, South Africa
u21457451@tuks.co.za

3rd Lukas Anthonissen (u21434345)
Computer Science Dept.
University of Pretoria
Pretoria, South Africa
u21434345@tuks.co.za

***Abstract*—In this study, we propose a real-time object tracking solution using a fine-tuned, pretrained YOLOv8 deep learning model. We detail the process of creating a custom dataset specifically for sports hockey and ice hockey videos, and describe the implementation of our model. Finally, we present the results of our approach, along with a comparison to previous work, highlighting the improvements in accuracy and efficiency.**

***Keywords*—Object Tracking, Object Detection, YOLOv8, Deep Learning**

I. Introduction

Real-time object tracking has become a significant area of research, particularly in the sports industry. By leveraging object tracking models, we can gain deeper insights into various sports. For example, critical sports analysis can be used to study athletes' behavior and identify ways to enhance their performance.

In this study, we propose a deep learning approach to achieve accurate object tracking in sports and ice hockey videos. Deep learning algorithms have the ability to learn and extract features directly from inputs [1]. Specifically, we fine-tune a pretrained YOLOv8 model, originally trained to detect a wide range of objects, including animals, humans, and vehicles, among others.

While previous studies have explored object tracking for sports and ice hockey videos [2], [3], there are limited datasets available for detecting hockey balls and pucks. This gap is due to the relatively limited focus on object tracking in field and ice hockey. To address this, we created a custom dataset by extracting frames from provided videos, and we integrated datasets with similar images for hockey ball and puck detection [4], [5]. Our primary objective is to accurately detect and track the hockey ball or puck in real-time, with our dataset containing annotations that highlight the ball/puck in each frame.

Additionally, we aim to provide an approach that enhances the performance of object tracking models, offering a foundation for future research in this field.

II. Methodology

A. Dataset Preparation

Despite having access to some datasets for ball/puck detection in field hockey/ice hockey, we still required a dataset specifically designed for our problem. Therefore, we created our own dataset by extracting all the frames from each of the provided videos. The ice hockey video has a resolution of 1920x1080 while the field hockey video has a resolution of 640x360. Both videos have an FPS of 25. The ice hockey video has a total of 280 frames while the field hockey video has 1052 frames.

We then had to manually label all the frames. Due to the limited number of frames, the model did not show good results achieving a Precision score of 10% at best. Just like tabular data, augmenting the data by applying different rotations, blurring, and contrast effects can significantly increase the dataset size and also enhance model performance. Therefore, we resized all the frames to a resolution of 640x640 and augmented the images by applying rotation between -15degrees to 15degrees. Figures 1 and 2 shows an example of applying the preprocessing techniques. Table I shows the effects of applying data augmentation to our dataset.

	Before Preprocessing	**After Preprocessing and Augmentation**
Total Images	1332	1785
Classes	1	1
Training set	932 (70%)	1560 (87%)
Validation set	267 (20%)	150 (8%)
Testing set	133 (10%)	75 (4%)
Image Resolution	1920x1080, 640x360	640x640

TABLE I: Effects of applying data augmentation

For training the YOLOv8 model, we first used two publicly available datasets: *Hockey_ball Detection* and *Hockey Puck detection*. These two dataset acted as general data for fine-tuning the model for detecting hockey balls/pucks in different videos with similar background to our input videos. After training the model for 150 epochs on each of these datasets, we used our augmented dataset as the final dataset to fine-tune the model such that it can specifically detect the objects in the two input videos.

(a) Field Hockey -10° rotation

(b) Field Hockey Normal

(c) Field Hockey +10° rotation

Fig. 1: Field Hockey Data Augmentation Example

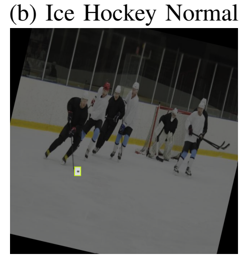
B. Object Tracking Model

The pretrained YOLOv8 model was fine-tuned to detect and ice hockey pucks and field hockey balls for two separate videos. This model is an ideal choice for solving our problem as it is considered a state-of-the-art object detection algorithm.

We tuned some the parameters to help enhance the model's detection performance. These hyperparameters are shown in Table II. The Adam optimiser was used to adjust the model weights during training. The Adam optimiser helps the model achieve greater generalisation capabilities by waking up dead neurons that are affected by activation saturation, making it an better choice than other optimisers such as the SGD optimiser [6], [7]. During training the model used the Adam optimiser to find optimal values for the learning rate (Lr) and momentum parameters. Additionally, we experimented with different values for the batch size (8, 16, 32) and number of epochs (100, 150, 200, 250). We found that a batch size of 32

(a) Ice Hockey -15° rotation

(b) Ice Hockey Normal



(c) Ice Hockey +15° rotation

Fig. 2: Ice Hockey Data Augmentation Example

and 150 epochs produced the optimal results. Finally, we used the default value of 640 as the image size. Figure 3 shows the process flow of fine-tuning the model.

TABLE II: Optimal Hyperparameter values

Optimiser	Lr	Momentum	Epochs	Batch Size	Image Size
Adam	**0.0002**	**0.9**	**150**	**32**	**640**

After fine-tuning the YOLOv8 model on the three datasets and evaluating it to ensure it is able to make accurate enough predictions we moved on to implement the object tracking algorithm. We saved the model checkpoint where it achieved the best results and loaded it in implementation. We experimented with different tracking algorithms such as the DeepSort algorithm but found that it struggles with tracking the ball/puck even though our detection model showed great results in detecting the object. Therefore, we opted to use the YOLOv8 model's built-in *track* function.

We simply read the video frame by frame and pass in the frame as an image for the fine-tuned YOLOv8 model. The model then makes the prediction of detecting the object and returns the coordinates of the bounding box. Our tracking algorithm then uses these coordinates and a **scaling factor of 1.2** to enlarge the detected region of the object. This then creates new coordinates for the bounding box. Furthermore, the algorithm then uses the enlarged bounding box and colours it using a bright orange colour. Additionally, we add the class

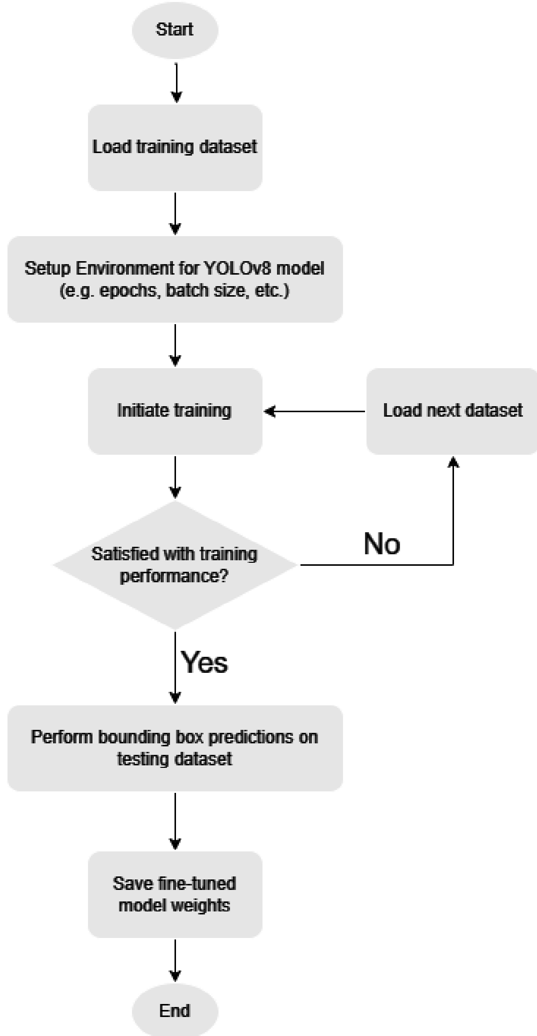


Fig. 3: Fine-tuning YOLOv8 process flow

of the detected object to further clarify that the algorithm detects a ball/puck depending on which input video is used. Figure 4 helps to illustrate our implementation process.

With each frame, the algorithm reads the annotated frame to a saved directory. Finally, after processing each frame the program reads each saved frame, resizes it to a 640x640 resolution and writes it to a video file. The output video is then saved in a separate directory for us to review.

III. Results

Experiments were conducted using a system equipped with two GPU T4 accelerators provided by Kaggle. As previously stated, the object detection model was trained and tested on three different datasets: the first contained hockey ball images, the second contained hockey puck images, and the third was a manually annotated and augmented dataset, created from extracting each frame from the two input videos provided. The performance indicators for these datasets have been summarised in Table III. The final augmented dataset outperformed all criteria, with the greatest precision, recall, and mean

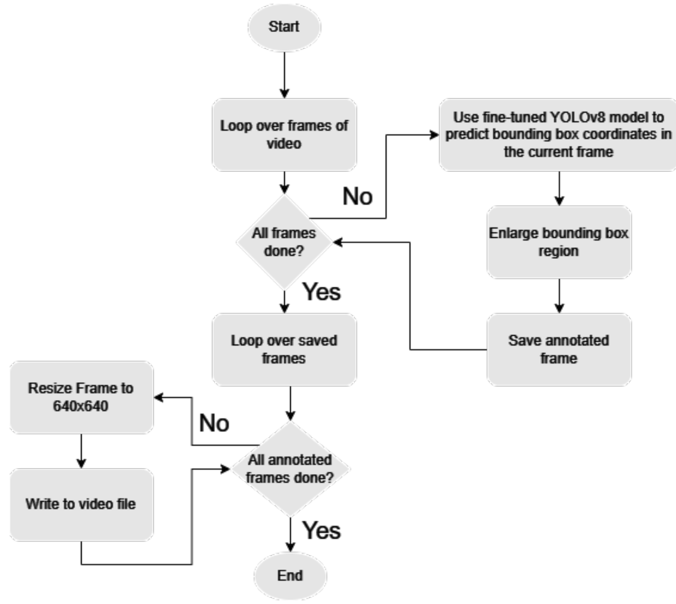


Fig. 4: Object Tracking Algorithm

Average Precision (mAP@50). The improvement can be attributed to the greater variability introduced by integrating both datasets and adding image augmentations, which most likely improved the model's capacity to generalise across multiple circumstances. This finding demonstrates the effectiveness of augmentations in increasing the robustness and accuracy of object detection models. The entire evolution of precision, recall, and mAP@50 across 150 epochs for each dataset is shown in Figures 6 to 8, where the datasets are labeled as *train* (Hockey Ball Detection), *train2* (Hockey Puck Detection), and *train3* (Augmented Dataset). These figures clearly reflect the improvement of precision, recall, and mAP@50 during the training epochs, demonstrating that the model trained on the augmented dataset consistently performed better.

TABLE III: Object Tracking Performance for the first two datasets

No. of epochs	Dataset	Class	Precision	Recall	mAP@50
150	Hockey_Ball Detection	ball	0.7543	0.6754	0.7099
	Hockey Puck Detection	puck	0.7922	0.4674	0.5001
	Final metric scores for augmented dataset		0.9466	0.6624	0.7679

If we compare our fine-tuned model's performance with that of previous work done by Patel et al. [2], [3] we find that our model outperformed both papers' approach across all criteria. The final findings show that the augmented dataset (*train3*) outperformed the individual hockey ball and puck datasets in precision and recall. The puck detection dataset (*train2*) demonstrated a relatively good precision of 79.22%, but its recall was substantially lower at 46.74%, indicating that the model struggled to detect pucks in certain images.

In contrast, the hockey ball dataset (*train*) demonstrated more balanced performance across precision and recall, though it still fell short of the augmented dataset. Based on the increased performance with each dataset we can confirm that the model was successfully fine-tuned to make accurate detections for the hockey ball and puck.

Fig. 5: Distribution of Class Loss over 150 epochs for each dataset

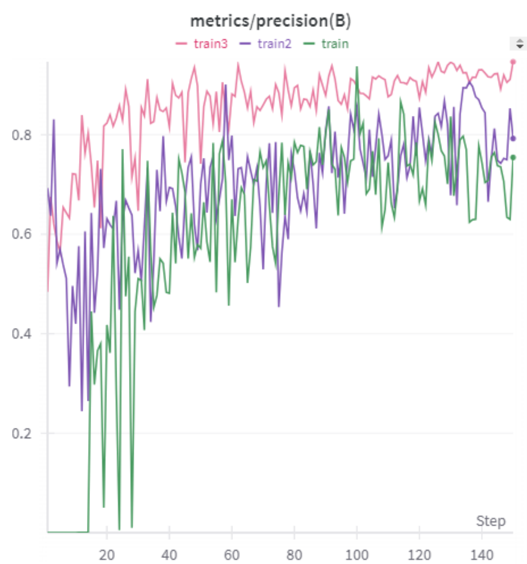


Fig. 6: Comparison of Precision between the three datasets across 150 epochs

Further insights into the training process can be seen in the loss distributions over 150 epochs, as presented in **??** and fig. 5. The box loss, which measures the inaccuracy in predicting the coordinates of bounding boxes, reduced consistently across all datasets, suggesting the model's enhanced capacity to localise objects. Similarly, the class loss, often calculated using Cross-Entropy Loss to assess the model's classification accuracy, revealed significant reductions, partic-

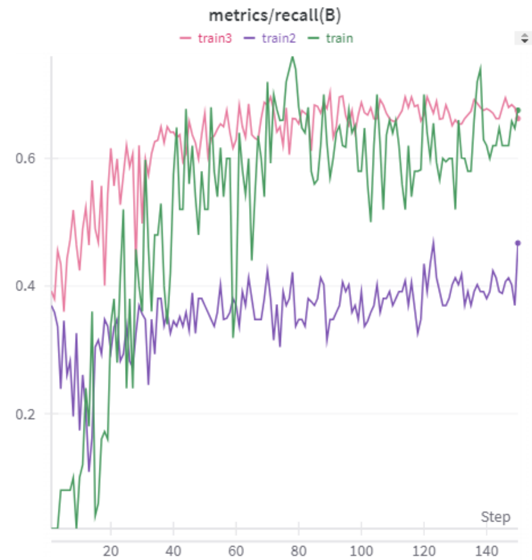


Fig. 7: Comparison of Recall between the three datasets across 150 epochs

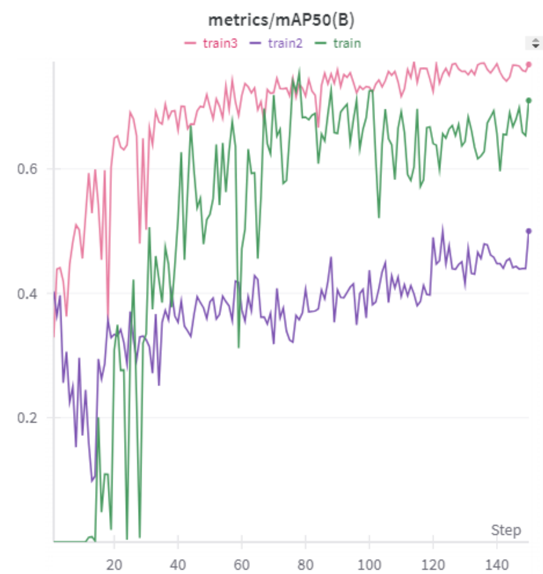


Fig. 8: Comparison of mAP between the three datasets across 150 epochs

ularly in the augmented dataset. These trends reflect the higher performance gained through dataset augmentation.

Finally, the augmented dataset not only provides a larger and more diverse training set, but it also enables the model to handle the complexities of identifying hockey balls and pucks under different situations.

IV. Conclusion

In this study, we proposed a solution to enhance the performance of real-time object tracking in field hockey and ice hockey videos. We demonstrated the effectiveness of data augmentation techniques in increasing the diversity and size of the training data, leading to improved model performance. Our results showed that the fine-tuned YOLOv8 model achieved better accuracy as new training data was incorporated. Furthermore, we compared our findings with previous work, highlighting the superiority of our approach in terms of detection accuracy and tracking reliability. Additionally, we introduced an innovative mechanism for enlarging and highlighting the detected region, improving the visual clarity of tracked objects. Our study provides a foundation for future work in object tracking for sports, and the proposed methods can be further built upon by researchers in this field.

[1] [2]

REFERENCES

[1] K. Rangasamy, M. A. As'ari, N. A. Rahmad, N. F. Ghazali, and S. Ismail, "Deep learning in sport video analysis: a review," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 1926–1933, 2020.

[2] S. Patel and D. Kamdar, "Accurate ball detection in field hockey videos using yolov8 algorithm," *Int. J. Adv. Res. Ideas Innov. Technol*, vol. 9, no. 2, 2023.

[3] S. H. Patel and D. Kamdar, "Object detection in hockey sport video via pretrained yolov3 based deep learning model." *ICTACT Journal on Image & Video Processing*, vol. 13, no. 3, 2023.

[4] project, "hockey_ball dataset," https://universe.roboflow.com/project-9c0mz/hockey_ball, apr 2024, visited on 2024-10-13. [Online]. Available: https://universe.roboflow.com/project-9c0mz/hockey_ball

[5] P. Burns, "Hockey puck detection dataset," https://universe.roboflow.com/peyton-burns/hockey-puck-detection, nov 2021, visited on 2024-10-13. [Online]. Available: https://universe.roboflow.com/peyton-burns/hockey-puck-detection

[6] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How do adam and training strategies help bnns optimization," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6936–6946. [Online]. Available: https://proceedings.mlr.press/v139/liu21t.html

[7] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, and W. E, "Towards theoretically understanding why sgd generalizes better than adam in deep learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 285–21 296. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf

[1] Github Repo: https://github.com/08Arno30/COS791-Project

[2] Annotated Dataset: https://universe.roboflow.com/university-p38pv/cos791-project/dataset/1