Improving YOLO11s Model for Real-Time Foosball Detection

Niusha Parsa
Politecnico di Torino
Turin, Italy

Sina Hamedani
Politecnico di Torino
Turin, Italy

Abstract

This paper explores augmentation strategies and training techniques to improve the YOLO11s model for detecting figures and balls on a foosball table. Experiments include grayscale, binary augmentations, HSV adjustments, and perspective transformations. The model achieved notable improvements in detection, but challenges persist due to diverse perspectives, varying lighting, and the complex interplay of different colors across figures, the ball, and the table background. Future work proposes domain adaptation, advanced architectures, and expanded datasets to further enhance detection robustness and address these challenges effectively.

1 Introduction

Detecting objects in foosball tables presents several challenges, including overlapping objects, dynamic and uneven lighting conditions, and significant variations in perspective. The complexity is further heightened by the similar colors of the figures, ball, and unseen and different color of table background, which often confuse object detection models not tailored for domain-specific applications. To address these challenges, this study focuses on enhancing the YOLO11s model through customized augmentation strategies and training techniques aimed at improving robustness and generalization.

To develop an effective solution, we utilized a publicly available foosball table dataset from Roboflow [1], consisting of annotated images of figures and balls under various conditions. This dataset contains multiple perspectives, occlusions, and lighting scenarios that simulate real-world gameplay. Our evaluation metrics included mAP@50 (mean average precision at 50% IoU), mAP@50-95 (mean average precision at multiple IoU thresholds), and FPS (frames per second) to measure detection precision and inference speed. mAP@50 and mAP@50-95 measure the model's ability to correctly detect and localize objects, while FPS reflects how quickly the model processes frames, a critical factor for real-time applications.

The project began with an exploratory phase where we evaluated the performance of various YOLO11 model [2] versions, including YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x. These models, pretrained on the COCO dataset [3] using Ultralytics' implementation [4], were compared based on their performance using the metrics mentioned above. Among them, YOLO11s was selected as the optimal model due to its balanced tradeoff between high detection accuracy and efficient inference speed, making it suitable for real-time object detection on a foosball table.

After selecting YOLO11s, our efforts were directed toward enhancing its generalization and robustness on a custom target dataset collected at the LINKS Foundation. This dataset was created by filming multiple matches on the foosball table from various angles and perspectives to better reflect real-world scenarios. The frames extracted from these recordings included diverse lighting conditions, occlusions, and complex visual variations. The most critical challenge was the similarity in color between the figures and the ball and even the perimeter of the playground. The colors of the figures and the ball in the target data almost overlapped or closely resembled each other, particularly in perspectives where the figures were rotated, and only the head of the figure was visible, making it easily mistaken for the ball. This problem was further compounded by the stark differences in the color schemes between the target dataset and the training dataset, since even the color of the background on the foosball tables differed significantly, adding another layer of complexity to the model's ability to generalize.

Another major challenge arose from the difference in perspectives between the original dataset and the target dataset: the original dataset included limited and fixed perspectives, while the target dataset captured matches from many different and diverse angles, posing a significant domain adaptation challenge. The issue of color-based confusion, when combined with variations in perspectives, severely impacted detection accuracy, particularly under angles that presented figures in orientations unfamiliar to the model.

To address these issues, we employed augmentation strategies tailored to mitigate the domain gap and improve generalization. These included grayscale and binary transformations to reduce color dependency, encouraging the model to focus on shape-based features rather than color. HSV adjustments were applied to reduce sensitivity to color variations, and random perspective transformations were introduced to simulate diverse viewpoints and camera angles. These augmentations were designed to equip YOLO11s with the ability to better adapt to the varied visual characteristics of our target dataset.

This work has broader implications for sustainable development in fields such as autonomous systems, industrial automation, and smart sports monitoring. Improvements in object detection can enhance real-time systems' efficiency, reduce errors, and lead to better resource utilization in applications such as autonomous vehicles, robotics, and surveillance systems. As foosball detection shares key similarities with these domains —especially in dynamic object tracking— the advances made here can be extended to improve performance in real-world environments.

This paper provides comprehensive insights into the experiments, comparing various augmentation techniques and their impact on detection performance. Although our results demonstrate improvements, challenges such as detecting overlapping objects and handling extreme perspective distortions remain. Future work will explore domain adaptation techniques, advanced architectures, and an expanded dataset that captures more diverse scenarios to further improve real-world detection performance and robustness.

2 Related Work

YOLO models [5] have transformed real-time object detection with their single-shot architecture, which simultaneously predicts bounding boxes and class probabilities. This efficient design makes YOLO ideal for applications like sports analysis, autonomous driving, and video surveillance. YOLO11 [2], developed by Ultralytics, builds on this legacy with anchor-free detection and enhanced feature pyramid networks, offering improved computational performance for domain-specific tasks like foosball detection.

For this project, we utilized pretrained YOLO11 models available through Ultralytics' implementation [4], pretrained on the COCO dataset [3]. We compared YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x using metrics such as mAP@50, mAP@50-95, and inference speed (FPS) to determine the most suitable model for real-time foosball detection.

YOLO11n achieved the highest detection precision due to its ability to detect small objects effectively, but its heavy architecture resulted in slower inference speeds, making it impractical for real-time gameplay.

YOLO11s provided the best trade-off between accuracy and speed. While slightly less precise than YOLO11n, it was significantly faster and well-suited to the simpler foosball dataset, meeting real-time detection requirements.

YOLO11m and YOLO11l offered incremental improvements in precision but came at the cost of reduced speed. YOLO11m was slower than YOLO11s without a substantial accuracy gain. Same, YOLO11l's performance was higher than smaller models, it did not offer a significant improvement over YOLO11s in terms of precision to justify its use in real-time foosball detection, where speed is critical." making them less favorable for this application.

YOLO11x achieved the highest mAP but at the cost of significantly lower FPS due to its increased computational complexity, limiting its effectiveness in detecting small objects like the foosball ball and figures.

To address the challenges posed by color similarities and varying perspectives in the target dataset, we adopted augmentation strategies. Studies like Shorten and Khoshgoftaar [6] emphasized the effectiveness of augmentations such as HSV shifts and grayscale transformations in improving model robustness by simulating diverse conditions. Grayscale augmentation specifically helps reduce the model's dependency on color cues, making it beneficial in scenarios where objects share similar colors or lighting varies significantly. D. C. Ciresan et al. [7] demonstrated that grayscale preprocessing enhances generalization by preventing over-reliance on color features, which is crucial given the overlapping colors between the ball, figures, and table perimeter. These studies inspired our use of augmentation-based solutions to tackle challenges related to color confusion, perspective shifts, and environmental variability in foosball detection.

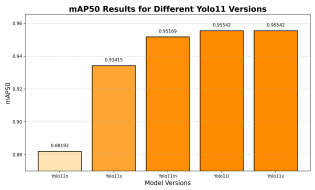
3 Method

3.1 Problem Statement

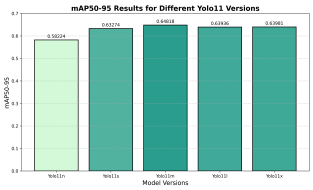
Detecting and distinguishing foosball figures and balls poses significant challenges due to overlapping objects, color similarities, and variations in perspective. The figures, ball and table perimeter often exhibit similar colors, which can cause confusion during detection—particularly when the figures are partially visible or rotated, making their heads resemble the ball. Furthermore, the color of the foosball table's background differs substantially from the backgrounds in the training data, creating additional visual discrepancies. Another major challenge arises from the limited perspective coverage in the training dataset, which features fixed and narrow angles, while the target dataset captures matches from a variety of dynamic and wide viewpoints. These challenges highlight the need for domain adaptation techniques and specialized augmentations to improve model robustness and generalization.

3.2 Model Selection

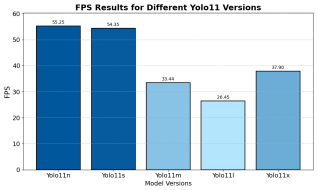
We evaluated pretrained YOLO11 models [2] provided by Ultralytics [4], including YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x, all pretrained on the COCO dataset [3]. Using the Roboflow foosball dataset [1], we fine-tuned each model for 5 epochs and compared their performance on key metrics: mAP@50, mAP@50-95, and inference speed (FPS). The results indicated that YOLO11n achieved the highest inference speed (FPS) but at the cost of lower detection accuracy, making it unsuitable for our project. In contrast, YOLO11s provided an optimal balance between detection precision and speed, performing well on the relatively simple foosball dataset while achieving high FPS. Thus, YOLO11s was selected as the primary model for this project.



(a) mAP50 results on each version of Yolo11s

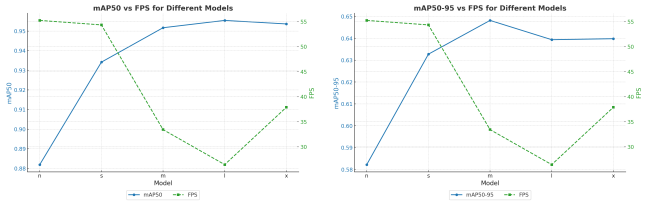


(b) mAP50-95 results on each version of Yolo11s



(c) FPS results on each version of Yolo11s

Figure 1: Evaluation results on different versions of Yolo11.



(a) mAP50 vs FPS (b) mAP50-95 vs FPS

Figure 2: Evaluation results on different versions of Yolo11

3.3 Approach

- **Offline Augmentations:** Grayscale Transformation: 30% of the images were converted to grayscale, reducing dependency on color and encouraging shape-based detection.
- **Binary Transformation:** 30% of the images underwent binary thresholding to highlight contours and edges, improving shape recognition.
- **Online Augmentations:** HSV Adjustments: Random hue, saturation, and brightness shifts simulated different lighting conditions.
- **Random Perspective Transformations:** Applied rotations, scaling, translations, and shearing to simulate varying camera angles and perspectives.
- **Combined Augmentation Pipeline:** Offline and online augmentations were integrated into the training pipeline to expose the model to a wide range of visual conditions, addressing challenges related to color confusion and domain adaptation.

3.4 Technical Tools

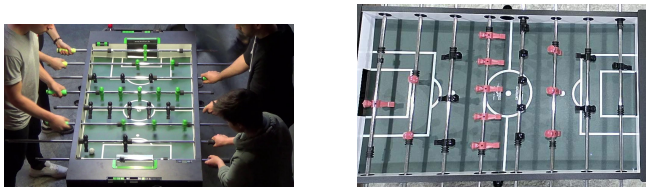
- **Model:** YOLO11s with pretrained weights from COCO [3].
- **Framework:** PyTorch with Ultralytics' YOLO11 implementation [4].

4 Experiment

4.1 Dataset Description

The dataset used in this project was collected and annotated using Roboflow [1], consisting of 5,111 images with two classes: Ball (class ID 0) and Figures (class ID 1). The annotations include diverse foosball images representing various gameplay situations. The key characteristics of the dataset are as follows:

- **Classes:** Ball and Figures
- **Diversity:** Captures varying lighting conditions, occlusions, and perspectives to reflect real-world challenges.



(a) Roboflow Sample 1 (b) Roboflow Sample 2

Figure 3: Samples from Roboflow dataset.

Additionally, to address domain adaptation challenges, a custom target dataset was created at the LINKS Foundation by filming matches on a foosball table. The target dataset introduced diverse perspectives, distinct color schemes, and lighting conditions significantly different from those in the training dataset.

(a) Target Data Sample 1 (b) Target Data Sample 2

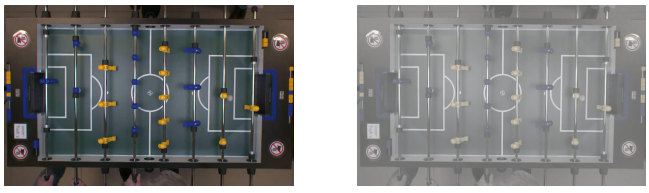
Figure 4: Samples from target dataset (captured from Links Foundation foosball table).

4.2 Experimental Configurations

To systematically evaluate and improve YOLO11s for real-time foosball detection, the following experiments were conducted:

(1) **Baseline Training:**
- Trained YOLO11s on the original dataset without any augmentations to establish a baseline for comparison.
- **Epochs:** 10

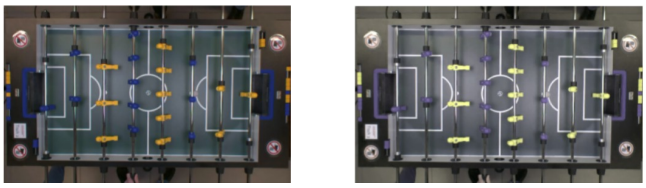
(2) **Offline Augmentation in Brightness, Saturation, and Contrast:**
- Pre-processed the training images with adjusted brightness, saturation, and contrast levels to simulate lighting variability and enhance robustness.
- **Epochs:** 10



(a) Example of original dataset. (b) Same image, augmented with the explained setting in this phase.

Figure 5: Visual example of offline augmentation in brightness, saturation, and contrast.

(3) **YOLO's Built-in HSV Augmentations (Color Jitter):**
- Applied random hue, saturation, and brightness shifts using YOLO's built-in online augmentations during training.
- **Epochs:** 10



(a) Example of original dataset. **(b) Same image, augmented with the explained setting in this phase.**

Figure 6: Visual example of Yolo built-in HSV augmentation (Color Jitter).

(4) **More Intense HSV Shifts:**
- Increased the intensity of the hue, saturation, and brightness shifts to simulate extreme lighting variations during gameplay.
- **Epochs:** 20

(a) Example of original dataset. **(b) Same image, augmented with the explained setting in this phase.**

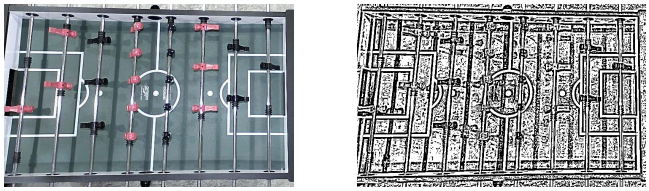
Figure 7: Visual example of more intense shift in HSV augmentation.

(5) **Offline 30% Grayscale Augmentation:**
- Converted 30% of the training images to grayscale offline to reduce the model's dependency on color features and encourage shape-based detection.
- **Epochs:** 25

(a) Example of original dataset. **(b) Same image, augmented with the explained setting in this phase.**

Figure 8: Visual example of offline 30% grayscale augmentation.

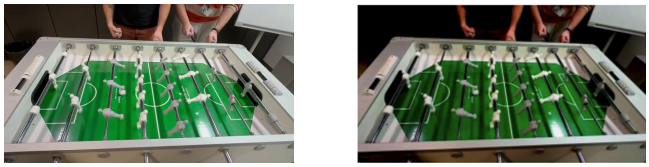
(6) **Offline 30% Partial Binary Augmentation:**
- Applied binary thresholding to 30% of the images to enhance edge and contour-based detection.
- **Note:** No training was conducted with this augmentation due to the intensity of the changes in the image and objects structure. Changes made the details of the images almost unrecognizable.



(a) Example of original dataset. **(b) Same image, augmented with the explained setting in this phase.**

Figure 9: Visual example of offline 30% partial binary augmentation.

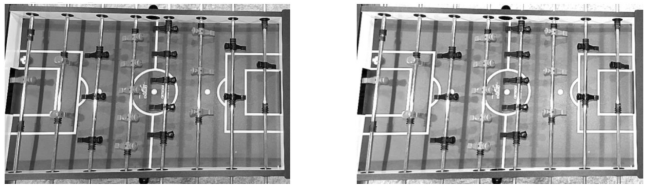
(7) **Adjust Target Test Images:**
- During the testing phase, brightness and contrast adjustments were applied to the target dataset to improve adaptation to varying lighting conditions.



(a) Example of original target test dataset. **(b) Same image, adjusted with the explained setting in this phase.**

Figure 10: Visual example of adjusted target test images.

(8) **Combined Augmentations:**
- Integrated multiple augmentations, including a strong HSV shift, random perspective transformations, and 30% offline grayscale augmentation, into the training process for comprehensive coverage of lighting, color, and perspective variations.
- **Epochs:** 40



(a) Example of used dataset.

(b) Same image, with new augmentation setting in this phase.

Figure 11: Visual example of combined augmentation (strong shift in HSV + random perspective + offline 30% grayscale augmentation.)

(9) **Extended Training of YOLO11s Without Augmentation:**
- YOLO11s was trained for 100 epochs on the original dataset without any augmentations to determine whether prolonged training could improve its performance on the foosball detection task.
- **Epochs:** 100

4.3 Results and Evaluation

Since the target dataset (LINKS Foosball Table) did not have ground-truth labels, it was not possible to present quantitative performance metrics such as mAP or precision in a numerical table. Instead, the evaluation of each experiment on the target dataset is presented qualitatively, highlighting key observations during testing.

(1) **Baseline Training (YOLO11s on Original Dataset without Augmentation, 10 epochs)** The model performed excellently on the training and validation datasets but performed poorly during testing on the target dataset. It failed to detect all the foosball figures consistently and frequently misidentified the ball due to color similarities between the figures and the ball in the target data. In many cases, it falsely detected multiple balls because the heads of the figures were misinterpreted as balls. Additionally, it occasionally considered parts of the table edge or even the player's hands as figures, further contributing to detection errors.

(2) **Offline Augmentation in Brightness, Saturation, and Contrast (10 epochs)** The model failed to train due to the excessive intensity of the augmented training data. The significant changes in brightness, saturation, and contrast made it difficult for the model to recognize objects, leading to failure during the training process.

(3) **YOLO's Built-in HSV Augmentations (Color Jitter, 10 epochs)** The model performed well during training and validation, but its performance on the target dataset remained poor. Similar issues were observed as in the baseline training, with incorrect detections of the ball and figures due to color confusion and perspective differences.

(4) **More Intense HSV Shifts (20 epochs)** With increased intensity of hue, saturation, and brightness shifts, the model performed well during training and validation. However, there was no noticeable improvement in detection performance on the target dataset.

(5) **Offline 30% Grayscale Augmentation (25 epochs)** The model achieved good performance during training and validation, but no significant improvements in detection on the target dataset were observed. While the grayscale augmentation was expected to help reduce color dependency, the model still struggled with the domain gap between the training and target data.

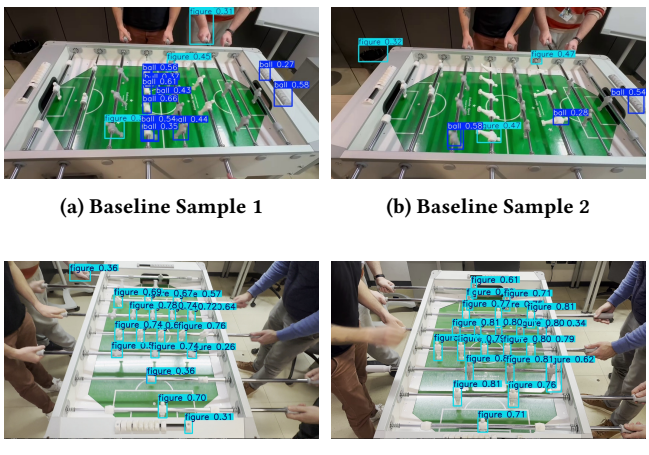
(6) **Offline 30% Partial Binary Augmentation** No training was conducted with this augmentation due to the drastic changes it introduced to the structure and appearance of the images. The binary thresholding made the image details and object contours almost unrecognizable, making it unsuitable for further experiments.

(7) **Adjust Target Test Images (Applied during testing)** During testing, adjustments to brightness and contrast were applied to the target dataset. This technique slightly improved overall detection but did not result in a significant breakthrough.

(8) **Combined Augmentations (40 epochs)** The model integrated multiple augmentations, including strong HSV shifts, random perspective transformations, and 30% grayscale augmentation. It performed well on the training and validation datasets and showed the best detection results on the adjusted target dataset compared to other experiments. While some false detections persisted, this approach demonstrated the highest robustness in adapting to diverse perspectives, lighting, and color conditions.

(9) **Extended Training of YOLO11s Without Augmentation (100 epochs)** The model performed well on the training and validation datasets but performed poorly during testing on the target dataset. Interestingly, its performance on the target data was even worse than that of YOLO11s trained for only 10 epochs, suggesting that simply increasing the number of epochs without augmentations did not address the domain adaptation challenges.

4.3.1 Sample of Detection Results.



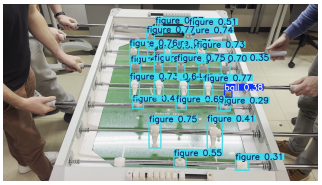
(a) Baseline Sample 1

(b) Baseline Sample 2

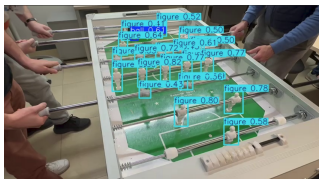
(c) Baseline Sample 3

(d) Baseline Sample 4

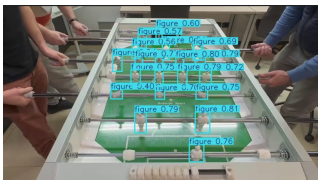
Figure 12: Detection results for 10 epochs trained YOLO11s on the original dataset without augmentations.



(a) Best Performance Sample 1

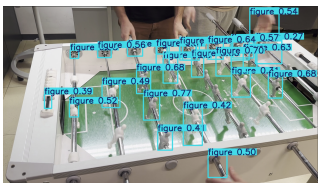


(b) Best Performance Sample 2

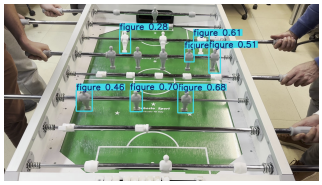


(c) Best Performance Sample 3

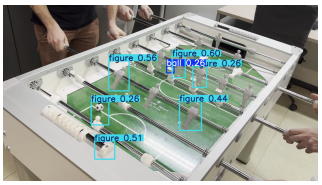
Figure 13: Qualitative examples showing the best performance achieved, resulted from combined augmentation (strong shift in HSV + random perspective + offline 30% grayscale augmentation).



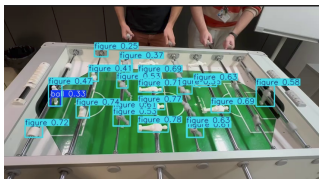
(a) Poor Performance e.g. 1



(b) Poor Performance e.g. 2



(c) Poor Performance e.g. 3



(d) Poor Performance e.g. 4

Figure 14: Qualitative examples showing poor performance, highlighting misdetections and incorrect classifications due to domain gaps and color similarity issues, resulted from combined augmentation (strong shift in HSV + random perspective + offline 30% grayscale augmentation).

5 Conclusion

The qualitative evaluation of our experiments demonstrates that combining multiple augmentation strategies —strong HSV shifts, random perspective transformations, and grayscale augmentations— was the most effective approach to enhancing the YOLO11s model's generalization on the target dataset. These augmentations addressed major challenges, including color similarity, lighting variation, and perspective shifts, by encouraging the model to focus more on shape-based features rather than relying solely on color. Specifically, grayscale and color-based augmentations, such as HSV shifts and color jitter, significantly reduced the model's dependency on color cues, helping it distinguish between the ball, figures, and table perimeter even when their colors overlapped.

While the baseline training and models without augmentations performed well on the original training and validation data, they struggled on the target dataset due to the significant domain gap. Extended training for 100 epochs without augmentation did not improve performance, highlighting that longer training alone cannot bridge the gap effectively. Instead, augmentations that promote shape-based learning, particularly geometric transforms like random perspective shifts, proved critical in helping the model ignore color-based confusion and detect objects more robustly under varying conditions.

However, the improvement was not uniform across all scenarios. While stronger augmentations improved detection in many cases, there were still instances where the model struggled—particularly in scenarios with extreme perspective changes, unfamiliar angles, or color configurations that were drastically different from those seen during training. Overlapping objects or bounding boxes in complex gameplay settings further contributed to detection challenges.

Future optimizations will focus on expanding the training dataset to include more diverse perspectives and scenarios, along with further tuning augmentation strategies to address edge cases. Additionally, exploring advanced shape-based methods and domain-specific adaptations may help bridge the remaining performance gaps, ensuring more robust detection in complex real-world environments.

6 Future Work & Other Approaches

Future efforts will focus on:

- Future optimizations will focus on expanding the training dataset to include more diverse perspectives and scenarios, addressing edge cases where the current model struggles. One of the key improvements will involve collecting more annotated and labeled data from the target domain, specifically from the LINKS Foosball Table, to better capture the variety of gameplay scenarios and environmental conditions. This expansion of the dataset will help reduce the domain gap and improve generalization on unseen data.
- Further refinement of augmentation techniques is essential for enhancing model robustness. Adjusting the intensity and frequency of augmentations such as HSV shifts, perspective transformations, and grayscale augmentations will be explored to ensure a more balanced and effective learning process. Tailoring augmentations to specific failure cases observed during evaluation will also be prioritized to reduce detection errors.
- To address the limitations observed in using a single model for simultaneous ball and figure detection, a dual-model approach will be explored. This method involves training two separate models: one dedicated solely to detecting figures and the other to detecting balls. During inference, both models will be run simultaneously, and their predictions will be combined using post-processing techniques to ensure more

accurate and robust detection. This strategy is expected to reduce confusion between figures and balls, particularly in cases where overlapping colors or ambiguous perspectives cause false detections.

- Additionally, we will consider switching to larger YOLO variants, such as YOLO11m or YOLO11l, or even exploring different architectures like YOLOv8 or Transformer-based object detectors. These models offer enhanced feature extraction capabilities and may prove beneficial in addressing scenarios with complex occlusions or overlapping objects.
- Advanced techniques such as domain adaptation or style transfer will be investigated to bridge the remaining domain gaps. Techniques like Mosaic/MixUp augmentation could further enhance model robustness by creating diverse training samples. Temporal processing or post-processing techniques could be employed to refine predictions, especially in cases where the model encounters ambiguous objects or overlapping bounding boxes.
- To improve detection precision, bounding box constraints will be considered, particularly for perspectives where we know that foosball figures or balls cannot overlap. These constraints could help reduce false positives and improve localization accuracy by leveraging prior knowledge about the foosball table's structure and gameplay rules.

By combining these strategies, future work aims to achieve more robust and accurate detection across diverse foosball scenarios while addressing the challenges of color-based confusion, overlapping objects, and complex perspectives.

Acknowledgments

This research was supported by the LINKS Foundation.

References

[1] Roboflow. Foosball dataset. https://universe.roboflow.com/tim-arnold-wqtri/foosball-4lfco/dataset/5. Accessed: February 2025.

[2] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *European conference on computer vision*, pages 740–755, 2014.

[4] Ultralytics. Ultralytics yolo implementation. https://github.com/ultralytics/ultralytics. Accessed: February 2025.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 779–788, 2016.

[6] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[7] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3642–3649, 2012.