Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 15, No. 2, 2024

YOLOv10-based Model for Player and Football Detection

Erzhizhi Hu

Faculty of Information Science and Engineering, Ocean University of China, Qingdao, 266100, China
eh2020@hw.ac.uk

Abstract: This study presents an advanced YOLOv10n-based method for the automatic detection of football players and balls directly from match videos. We enhance the YOLOv10 architecture with several significant improvements, including additional detection heads, the integration of C2f_faster and C3_faster modules for enhanced processing speed and accuracy, and the inclusion of BotNet modules with self-attention mechanisms for managing complex visual scenes. Further, we incorporate GhostConv modules to reduce computational overhead while maintaining effective feature extraction. These architectural modifications ensure robust detection capabilities in real-time sports environments, addressing challenges such as high-speed movements, frequent occlusions, and variable lighting conditions typical of both indoor and outdoor stadiums. Validation on internet-sourced images from football matches demonstrates the practicality and effectiveness of our model.

Keywords: YOLOv10; Target detection; Football detection; Player detection; Deep learning.

1. Introduction

The automatic detection of football players and the football during dynamic and fast-paced matches present considerable challenges for computer vision systems. These challenges are primarily due to the high-speed movements of players, frequent occlusions, and variable lighting conditions typical of outdoor and indoor stadiums. Efficient and accurate detection is crucial for applications ranging from tactical game analysis to enhanced fan experiences through augmented reality.

Recent advancements in object detection technologies have led to significant improvements in real-time video analysis. Among these, the YOLO (You Only Look Once) [1] series stands out due to its balance of speed and accuracy, making it particularly suitable for scenarios requiring immediate processing. The latest iteration, YOLOv10 [2], introduces optimized architectural enhancements and superior feature extraction capabilities, which are pivotal for handling the complexities of sports environments like football.

In this paper, we explore the application of YOLOv10 to the specific task of detecting football players and the ball within a game. We adapt YOLOv10 to better recognize the small, fast-moving ball and distinguish between players in close contact—a frequent occurrence in football. The objective of this research is to leverage the real-time processing power and improved detection accuracy of YOLOv10 to provide a robust solution for sports analytics.

Through comprehensive experiments and detailed analyses, we aim to demonstrate that the modified YOLOv10 model can significantly outperform traditional methods in terms of accuracy and speed, thereby providing a substantial foundation for future developments in sports technology and real-time object tracking.

2. Related works

The automated detection of football players and the ball in football games has been a significant focus in sports analytics and computer vision. Researchers have explored various methods to improve the accuracy of player detection, transitioning from basic image processing techniques to advanced machine learning models. This section reviews key developments in these technologies, emphasizing their evolution and the challenges they address, setting the stage for the advancements proposed in this study.

In 2010, Sławomir Maćkowiak and his team developed a system employing Histogram of Oriented Gradients (HOG) descriptors and Support Vector Machine (SVM) classification to detect football players in broadcast videos [3]. This method incorporated playfield detection and player tracking, showing promising results in dynamic scenes, though it faced challenges with high occlusion and rapid camera movements.

By 2018, the field had advanced significantly with Cem Direkoglu and colleagues developing a method for player detection in field sports, using a unique feature extraction technique from binary edge images and solving a diffusion equation for shape information [4]. This technique achieved an accuracy of 93% in player detection across various field sports datasets, marking significant improvements in handling variations in player appearances and environmental conditions.

In 2020, Jacek Komorowski and colleagues introduced the "FootAndBall" system [5], a sophisticated detector based on a fully convolutional architecture that identifies football players and the ball in high-resolution videos. Leveraging a Feature Pyramid Network design, this system processed video at 37 frames per second on low-end hardware, offering robust detection in complex, dynamic scenes and significantly outperforming other methods with fewer parameters, demonstrating a practical approach to real-time sports analytics. In the same year, Anthony Cioppa and his team proposed an innovative method combining fisheye and thermal cameras for real-time player detection on football fields [6]. Utilizing a student-teacher distillation method to educate the network with multimodal data, this method adapted well to various lighting and weather conditions, significantly enhancing automated sports analytics and monitoring systems.

By 2022, Wang Tianyi et al had developed a deep

convolutional neural network-based algorithm [7] for detecting football players in videos, which was significantly more parameter-efficient compared to YOLO and capable of processing an entire image in one pass and reached an accuracy of 91.5% on the ISSIA-CNR dataset. Their approach, which integrates feature maps at various scales, achieved a substantial improvement in detection accuracy, significantly enhancing the model's effectiveness across diverse match footage.

Last year, Kadir Diwan and his team introduced a sophisticated system using a YOLOv5-based deep learning model for detecting and tracking football players and the ball [8]. Optimized for real-time analysis, their system processes videos of any size and length, effectively managing the dynamic and complex environment of football matches. By integrating YOLOv5 with the Deep SORT tracking algorithm, they achieved high accuracy and efficiency, comparable to human annotation, which greatly enhances real-time sports analytics and strategic game analysis.

In this work, we enhance the YOLOv10 model by focusing on modifications to its detection head, specifically to boost its capabilities in real-time tracking of football players and the ball. By incorporating a tailored self-attention mechanism into the detection head, our model adapts to the dynamic and complex environments typical of football matches, maintaining high accuracy despite rapid movements and frequent occlusions. This improvement significantly refines the model's performance, making it more effective for real-time sports analytics and robust in tracking fast-moving objects.

3. Methodology

In this section, we will present our dataset and our enhanced model.

3.1. Dataset

The "Football Player Detection" dataset is used in this study, this dataset comprises an extensive collection of images derived from a variety of football games videos. These images depict diverse situations, including close-up views, wide-angle perspectives, various lighting environments, and different concentrations of players on the field. It contains 10.3k images as the training set, 927 images as the validation set and 520 images as the test set. The label in this dataset using annotation files in YOLOv8 format, which is also appliable in YOLOv10.

3.2. Base model and enhanced model

In this study, we utilize YOLOv10 as the base model, which is known for its flexible architecture and enhanced activation functions like Mish and Swish, which improve training dynamics compared to traditional ReLU. YOLOv10n's modular design allows for easy customization and integration of new features, enhancing its performance in complex object detection tasks.

3.2.1. YOLOv10

YOLOv10 is an advanced deep learning-based object detection model that leverages an enhanced network architecture and sophisticated training techniques to significantly improve detection capabilities. The introduction of a novel dual label assignment mechanism in YOLOv10 allows the model to receive multiple positive samples from each annotated object during the training phase. This innovation ensures that the model remains efficient during inference while handling multiple object identities effectively.

The architecture of YOLOv10 includes a robust backbone for feature extraction, typically utilizing modified versions of CSPNet [9]. This backbone is crucial for capturing complex features from input images. Further refining the model's capability, the neck utilizes mechanisms such as the Feature Pyramid Network (FPN) [10] to amalgamate features across different scales, thus enhancing the detection of objects of various sizes. The detection head then applies a series of convolutional layers directly to the feature maps, predicting bounding boxes and class probabilities. This design minimizes the reliance on traditional post-processing steps like Non-Maximum Suppression (NMS) [11], favoring instead the use of advanced loss functions and direct network output training techniques to boost detection accuracy, making YOLOv10 highly effective for real-time object detection applications.

3.2.2. Enhanced-YOLOv10n

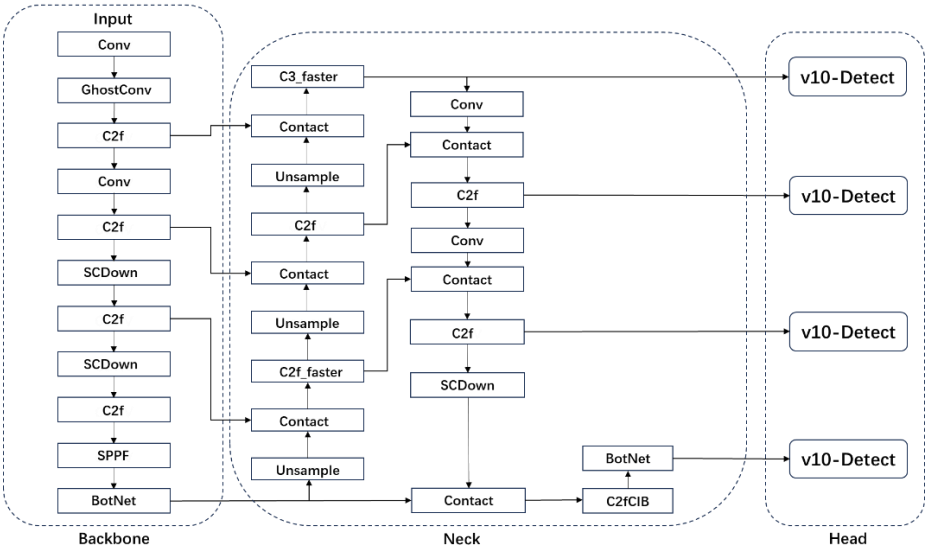


Figure 1. Architecture of our model

As Figure 1 shows, we first enhanced the YOLOv10n model by adding a new detection head into its head architecture. This modification incorporates additional upsampling and concatenation operations that facilitate a

more seamless integration of features from various depths of the network. By directly linking each detection head to distinct feature layers, the model optimizes its ability to capture and process diverse spatial details across different scales. These architectural enhancements not only enable YOLOv10n to perform more accurate and faster object detections in complex scenes but also significantly boost its efficacy in real-time player and ball tracking applications, providing a substantial improvement over the original YOLOv10n structure.

Then we applied the FasterNet [12] architecture enhancements to the YOLOv10n model by substituting several C2f modules with C2f_faster and C3_faster modules within the detection head structure. These modifications leverage the partial convolution technique, which targets a subset of input channels for convolution, thereby reducing the computational load and memory access demands. Specifically, the computational complexity of the modified layers can be estimated using the formula:

$$FLOPs = H \times W \times \left(\frac{D}{n}\right) \times (K \times K) \times C \quad (1)$$

where H and W are the height and width of the feature map, D/n represents the subset of input channels processed, K is the kernel size, and C is the number of output channels, with n indicating the reduction factor in channel usage. This reduction in FLOPs directly contributes to decreased latency and energy consumption. The memory access reduction facilitated by these modules is given by:

$$Memory\ Access = \left(H \times W \times \left(\frac{D}{n}\right)\right) + \left((K \times K) \times \left(\frac{D}{n}\right) \times C\right) \quad (2)$$

these calculations show how the C2f_faster and C3_faster modules decrease both the computational and memory overhead, enhancing the model's efficiency without compromising detection accuracy. The C2f_faster modules specifically enhance the model by streamlining the processing of features through selective convolution on reduced input channels. This helps in maintaining high processing speeds while ensuring that essential features are not overlooked. Additionally, the introduction of C3_faster enriches the model's capabilities by handling even more complex feature interactions, thereby fine-tuning the model's capacity to discern subtle distinctions in object features. These enhancements not only make the YOLOv10n model quicker but also bolster its robustness in detecting objects across diverse scales and conditions.

We also enhanced the YOLOv10 architecture by integrating the BotNet [13] module into its backbone and head configurations. Specifically, we replaced the Partial Self-Attention Module (PSA) module in the final layer of the YOLOv10 backbone with a BotNet block, and added an additional BotNet layer at the end of the detection head. This adaptation leverages the self-attention mechanism inherent in BotNet, represented by the formula:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where Q, K, and V are the query, key, and value matrices derived from the input feature maps, and d_k is the dimensionality of the keys. This mechanism focuses more comprehensively on relevant features across the input space, enabling more refined feature extraction and object recognition. The integration of BotNet into the backbone enhances the global contextual awareness of the network, while its addition to the head ensures that the final feature representations are highly discriminative. The modifications enhance the model's overall precision and efficiency, especially in complex object detection scenarios where understanding contextual relationships is crucial for performance.

Additionally, we implemented an enhancement to the YOLOv10 backbone by replacing the traditional convolutional module in the second layer with the innovative GhostConv [14] module. This modification leverages the efficiency of GhostConv, which reduces the computational load through an ingenious mechanism of generating "ghost" feature maps. These ghost feature maps are derived using a formula:.

$$Ghost\ feature\ maps = Intrinsic\ maps \times Linear\ operations \quad (4)$$

where a small number of intrinsic maps are transformed by cheap linear operations to produce additional feature maps, effectively reducing the number of direct computations required. This allows the GhostConv module to deliver similar or enhanced performance compared to traditional convolutions but with fewer parameters and lower computational demands. By integrating the GhostConv module into the YOLOv10 architecture, the model becomes better suited for real-time operations in resource-limited environments. This enhancement retains critical feature extraction capabilities and dramatically reduces both energy consumption and computational load, thus boosting the model's adaptability across diverse deployment platforms.

3.2.3 Model evaluation

We use the mean Average Precision at 50% IoU threshold (mAP50) to evaluate the results of our model, a standard metric in object detection that measures accuracy by averaging the precision achieved across all recall levels for each class at an IoU threshold of 0.5. The mAP50 is calculated by first computing the Average Precision (AP) for each class specifically at an IoU threshold of 0.5, and then averaging these AP scores across all classes. AP at this threshold is derived from the area under the precision-recall curve, calculated as follows:

$$AP = \frac{1}{11}\sum_{r \in \{0, 0.1, \ldots 1.0\}} \max_{\tilde{r}:\tilde{r}>r} p(\tilde{r}) \quad (5)$$

where r represents specific recall levels, $\tilde{r}$ represents the recall levels where the precision is calculated, $p(\tilde{r})$ is the precision at recall $\tilde{r}$. The IoU metric evaluates the overlap between predicted and actual bounding boxes, enhancing the relevance of the precision-recall relationship:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (6)$$

The mAP, a mean of these AP values, offers a comprehensive measure:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (7)$$

where N is the number of classes? This approach, integrating AP, mAP, and IoU, robustly evaluates the precision and accuracy of our model's ability to detect and precisely localize objects across various scenarios and object categories.

4. Results and discussion

In this study, we used the experimental hardware environment consisted of a remote server equipped with 16 vCPU Intel(R) Xeon(R) Platinum 8474C and an RTX 4090D with 24GB of memory. The algorithm development was based on the YOLOv10 framework and CUDA libraries, with visualization implemented using OpenCV. The entire detection system was developed by using the Python programming language, supported by the Ubuntu 22.4 operating system.

4.1. Model performance

Table 1 summarizes the evaluation results of our model on Football Player Detection dataset. Our model gives the highest score on this dataset compared with YOLOv10n.

Table 1. Comparison of results from different models

Model	Precision (P)	Recall (R)	mAP50	mAP50-95
YOLOv10n	0.881	0.728	0.801	0.507
Our model	0.900	0.757	0.843	0.531

In our comparative evaluation, the modified model shows significant improvements over the baseline YOLOv10n. Precision increased by 2.16%, from 0.881 to 0.900, enhancing the possibility of true positive detections. Recall improved by 3.98%, from 0.728 to 0.757, indicating better overall object identification. The mAP50 saw a 5.24% increase, from 0.801 to 0.843, demonstrating improved detection accuracy at moderate IoU levels. Additionally, the mAP50-95 improved by 4.73%, from 0.507 to 0.531, showing enhanced robustness across varied detection challenges. These enhancements confirm the effectiveness of our model in complex detection scenarios, supporting its application in real-time object tracking systems.

Building upon these initial results, the substantial performance gains can be attributed to strategic architectural enhancements implemented in the modified model. The integration of four detection heads allows for effective handling of multi-scale detection, crucial for accurately recognizing objects of various sizes, significantly contributing to the improved mAP scores. The adoption of C2f_faster and C3_faster modules optimizes the processing of features, which enhances the model's speed and accuracy, as reflected in the increased precision and recall. Furthermore, the inclusion of BotNet modules with self-attention mechanisms in the model's architecture significantly boosts its ability to capture complex inter-dependencies and long-range interactions within images, enhancing detection accuracy in challenging scenarios. Additionally, the replacement of standard convolution layers with GhostConv modules reduces computational overhead while maintaining robust feature extraction capabilities, facilitating superior performance across more stringent IoU thresholds as evidenced by the improvements in mAP50-95. These comprehensive modifications not only enhance the model's efficiency but also its effectiveness in real-time object detection and tracking scenarios, underscoring the advanced capabilities of the upgraded system.

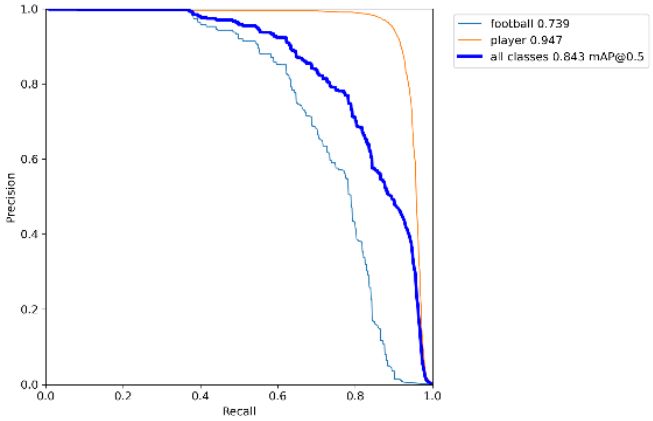


Figure 2. Precision-Recall curve of our model

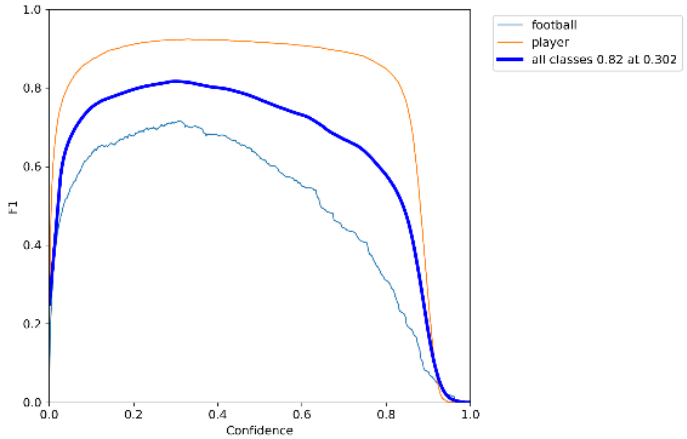
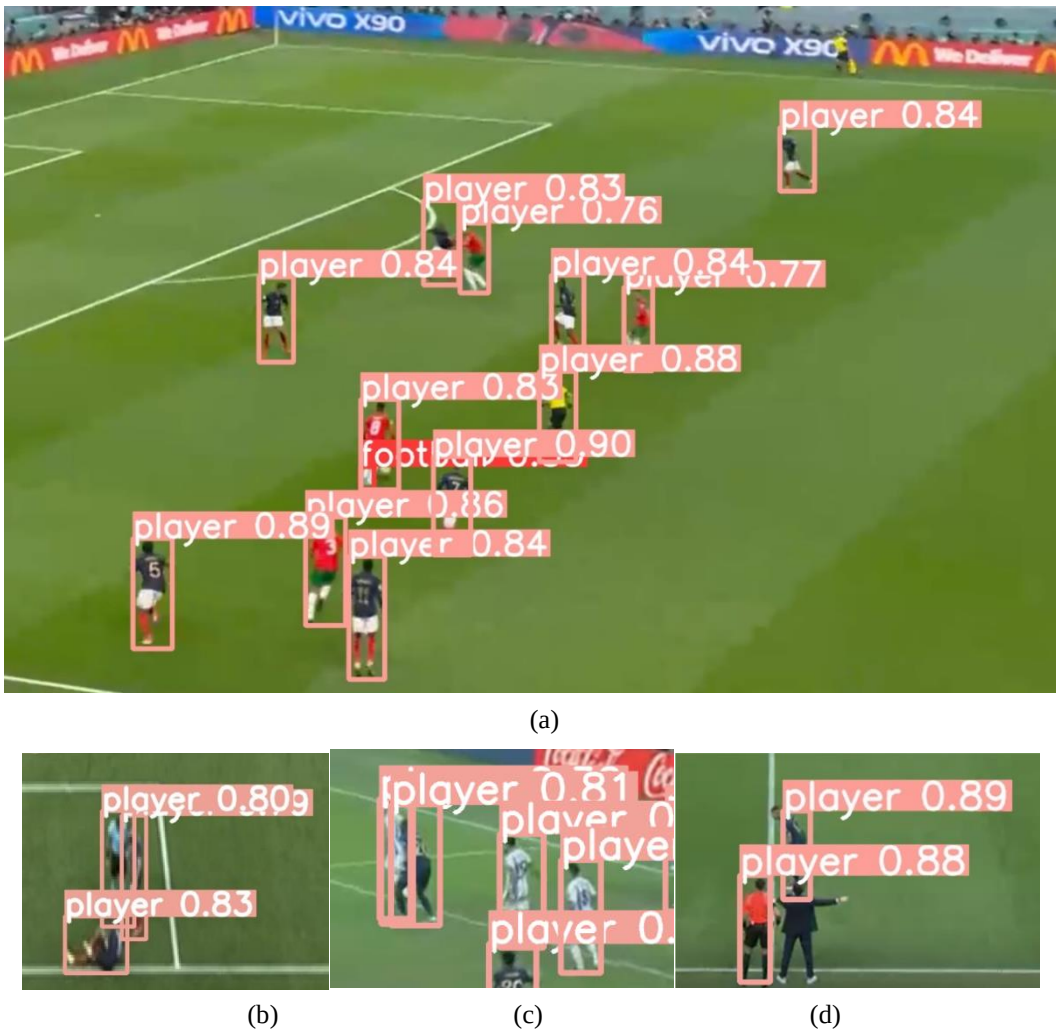


Figure 3. F1-confidence curve of our model

The analysis based on Figure 2 and Figure 3 illustrates distinct aspects of our model's performance in detecting

players and footballs. Figure 2 shows that the player detection curve maintains high precision across various recall levels with an Average Precision (AP) of 0.947, signaling strong and consistent identification capabilities. Conversely, the football detection curve depicted in the same figure indicates a lower AP of 0.739, pointing to the challenges the model faces in accurately detecting smaller and faster-moving objects like footballs. Figure 3 further investigates the model's performance across different confidence thresholds, highlighting that player detections are robust and maintain strong performance across a wide range of thresholds. However, football detections show a sharp decline after peaking early, suggesting the need for fine-tuning the model's confidence settings to better capture these quick and small objects. Overall, the optimal performance of the model, achieving an F1 score of 0.82 at a confidence threshold of 0.302, as shown in Figure 3, reveals effective and varied detection capabilities across classes.



(a)

(b) (c) (d)

Figure 4: Model performance in different situations: (a) A normal football match. (b) One player sat on the ground. (c) Two players got a close contact. (d) A none-player person appeared in the shot.

4.2. Model practical application

To further validate the practicality of our model, we tested it with images from random football matches sourced from the internet. Figure 4 illustrates the model's performance on previously unseen images. In Figure 4(a), the model successfully detects all players and the ball on the ground, each with a confidence score of no less than 0.75, outperforming traditional models. In Figure 4(b), the model adeptly identifies a player sitting on the ground with a confidence score above 0.8. Figure 4(c) shows the model's ability to discern two closely interacting players, detecting both based on visible portions of their bodies despite their overlap. In Figure 4(d), where an none-player individual appears, our model accurately refrains from misidentifying this person as a “player”, while correctly recognizing two other players with high confidence scores of 0.88 and 0.89. These results underscore the model's exceptional performance on diverse and challenging detection tasks, confirming the robustness and efficacy of our algorithm.

5. Conclusion

In conclusion, we have developed an effective YOLOv10n-based method for automatically detecting football players and balls from match videos. Our model features several enhancements including additional detection heads, C2f_faster and C3_faster modules for increased speed and precision, and BotNet modules with self-attention mechanisms for handling complex scenes. Further integration of GhostConv modules enhances computational efficiency. Validation through internet-sourced images has confirmed the model's robustness in real-world scenarios. By advancing accuracy and efficiency in sports analytics and player tracking, our model seeks to transform how sports are played, coached, and experienced, merging the dynamic nature of live sports with the precision of automated analytics. However, the detection of footballs remains a challenge. Moving forward,

we plan to address this by further refining our feature extraction techniques and optimizing our network architecture to improve performance in real-time sports environments and enhance tactical analysis and player evaluation.

Reference

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[2] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). Yolov10: Real-time end-to-end object detection. arxiv preprint arxiv:2405.14458.

[3] Maćkowiak, S., Kurc, M., Konieczny, J., & Maćkowiak, P. (2010, September). A complex system for football player detection in broadcasted video. In ICSES 2010 International Conference on Signals and Electronic Circuits (pp. 119-122). IEEE.

[4] Direkoglu, C., Sah, M., & O'Connor, N. E. (2018). Player detection in field sports. Machine Vision and Applications, 29, 187-206.

[5] Komorowski, J., Kurzejamski, G., & Sarwas, G. (2019). Footandball: Integrated player and ball detector. arxiv preprint arxiv:1912.05445.

[6] Cioppa, A., Deliege, A., Huda, N. U., Gade, R., Van Droogenbroeck, M., & Moeslund, T. B. (2020). Multimodal and multiview distillation for real-time player detection on a football field. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 880-881).

[7] Wang, T., & Li, T. (2022). Deep Learning-Based Football Player Detection in Videos. Computational Intelligence and Neuroscience, 2022(1), 3540642.

[8] Diwan, K., Bandi, R., Dicholkar, S., & Khadse, M. (2023, February). Football player and ball tracking system using deep learning. In Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1 (pp. 757-769). Singapore: Springer Nature Singapore.

[9] Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391).

[10] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[11] Neubeck, A., & Van Gool, L. (2006, August). Efficient non-maximum suppression. In 18th international conference on pattern recognition (ICPR'06) (Vol. 3, pp. 850-855). IEEE.

[12] Chen, J., Kao, S. H., He, H., Zhuo, W., Wen, S., Lee, C. H., & Chan, S. H. G. (2023). Run, don't walk: chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12021-12031).

[13] Srinivas, A., Lin, T. Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 16519-16529).

[14] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1580-1589).