DOI: https://doi.org/10.52436/1.jutif.2025.6.1.4165
p-ISSN: 2723-3863
e-ISSN: 2723-3871

FOOTBALL PLAYER TRACKING, TEAM ASSIGNMENT, AND SPEED ESTIMATION USING YOLOV5 AND OPTICAL FLOW

Matthew Raymond Hartono*1, Christy Atika Sari 2, Rabei Raad Ali3

1,2Faculty of Computer Science. Universitas Dian Nuswantoro, Semarang, Indonesia
3Faculty of Computer Science, Northern Technical University, Mosul, Iraq
Email : 1111202113275@mhs.dinus.ac.id, 2christy.atika.sari.dsn.dinus.ac.id, rabei@ntu.edu.iq

(Article received: December 3, 2024; Revision: December 17, 2024; published: February 20, 2025)

Abstract

Football analysis is indispensable in improving team performance, developing strategy, and assessing the capabilities of players. A powerful system that combines YOLOv5 for object detection with optical flow tracks football players, assigns them to their respective teams, and estimates their speeds accurately. In the most crowded scenarios, the players and the ball are detected by YOLOv5 at 94.8% and 93.7% mAP, respectively. KMeans clustering based on jersey color assigns teams with 92.5% accuracy. Optical flow is estimating the speed with less than 2.3%. The perspective transformation using OpenCV improves trajectory and distance measurement, overcoming the challenges in overlapping players and changing camera angles. Experimental results underlined the system's reliability for capturing player speeds from 3 to 25 km/h and gave insight into the dynamic nature of team possession. However, there is still some challenge: 6% accuracy degradation in high overlap and illuminative changes. The future work involves expanding the dataset for higher robustness and ball tracking, which will comprehensively explain the dynamics of a match. The paper presents a flexible framework for automated football video analysis that paves the way for advanced sports analytics. This would also, in turn, enhance informed decision-making by coaches, analysts, and broadcasters by providing them with actionable metrics during training and competition. The proposed system joins the state-of-the-art YOLOv5 with optical flow and thereby forms the backbone of near-future football analysis.

Keywords: *football analytics, object tracking, optical flow, player metrics, team classification, YOLOv5*

1. INTRODUCTION

Football analytics[1] has grown into an important tool helping any team in terms of performance, strategy inclusions, and evaluation of players. Technology, especially computer vision[2], has flipped the way coaches and analysts look towards the game. With the use of automated systems, it is possible to track the player movements[3], detect teams, and estimate player speeds given insights that were earlier possible with manual observation only. These are changing the way teams make decisions and approach training with deeper insights into match dynamics.

The following research investigates the potential of YOLOv5[4][5], a state-of-the-art deep learning[6] framework for object detection in football. The capability of YOLOv5 in real-time detection of players and objects with high precision is one of the best fits for football video analysis. This work integrates YOLOv5 with optical flow[7] and K-Means clustering[8], considering these challenges in recognizing team members by uniform colors in fast-action videos and the tracking of players in complicated video environments. The following work will describe the application of histogram-based clustering[9] techniques to dynamic team categorization according to jersey color, enabling much better analysis of the movement and interaction of players

Another critical aspect of the game is the speed at which the football players move around, which this study tries to track by analyzing player positions frame by frame. The understanding of player speed becomes important not only from the point of view of physical performance but also in understanding tactical decisions, positioning, and overall team strategy. Coaches will get insight into individual and team dynamics in more detail.

What makes this research unique, however, is the embedded use of many technologies to carry out the object detection task: YOLOv5, optical flow for motion refinement, and clustering techniques for team classification. These not only enable the solving of technical problems but also make football analytics quite accessible and practical. With this research, it lays down the foundation for further works that can now enable development of tools to analyze even the sophisticated game in real time.

Various works have discussed different approaches to football player tracking, team assignment, and speed estimation using YOLOv5 and other tracking algorithms. For instance, one of the works has shown how effective YOLOv5 is in detecting players and the ball to then create positional data for further analysis and statistical tracking.[10] It is found that YOLOv5 performs very well in detecting the ball, one of the smallest objects, and also in crowded scenes, at high inference speed. However, overlapping bounding boxes and class imbalance remain the main challenges, especially in the case of underrepresented classes like goalkeepers. These clearly indicate more data augmentation and model fine-tuning are needed for better detection accuracy.

Other works using YOLOv5 integrated it with advanced tracking algorithms[11], including DeepSORT and ByteTRACK. In DeepSORT, object detection by YOLOv5 identifies unique identifications of the players and tracks inter-frame movement via a Kalman filter combined with a Mahalanobis distance metric[12]. Correspondingly, the authors computed the players' velocities on the basis of team movement tracking to estimate the covered distance during a game. Meanwhile, ByteTRACK has been combined with YOLOv5, which currently is the state-of-the-art multi-object tracker, to further ensure computational efficiency for real-time tracking of players and the ball. These methods indicate an increased need for curated datasets and fine-grained annotations that really improve model versatility, specially for tracking additional object classes like referees and goalkeepers.

The combined use of YOLOv5 with advanced tracking algorithms and curated datasets underlines the possibility of automating sports analytics and performance evaluation. It is these methods that, by mitigating such challenges as class imbalance and occlusions, lay the bedrock for more effective and efficient football video analysis..

2. METHOD

In this research, we're using a framework to analyze football videos using a combination of the YOLOv5 model for player and ball detection, optical flow for trajectory refinement, and color clustering for team assignment. Figure 1 below illustrates the methodology employed in this study for automating football player tracking, team assignment, and speed estimation.

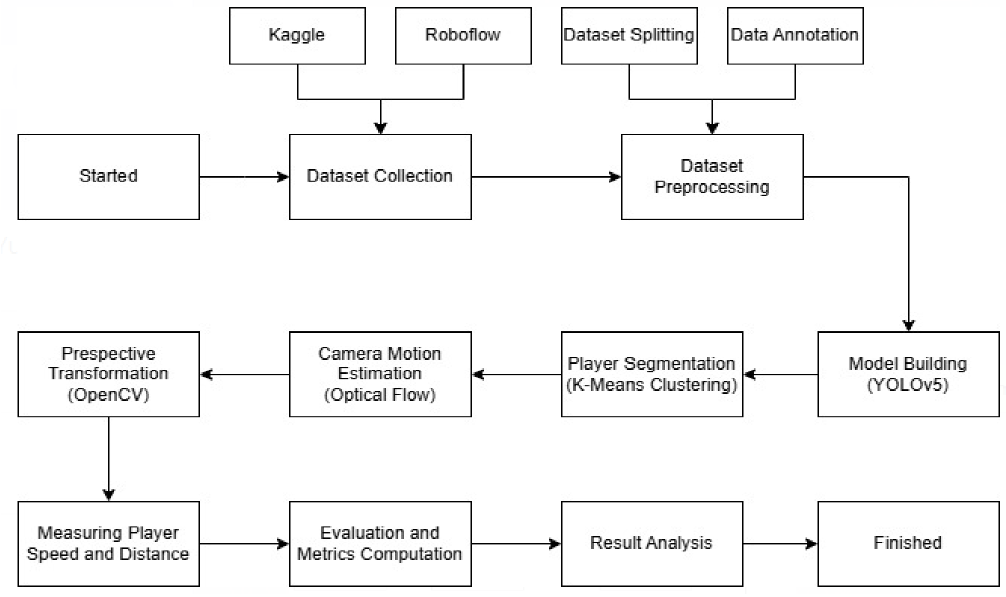
Kaggle
Roboflow
Dataset Splitting
Data Annotation
Started
Dataset Collection
Dataset Preprocessing
Prespective Transformation (OpenCV)
Camera Motion Estimation (Optical Flow)
Player Segmentation (K-Means Clustering)
Model Building (YOLOv5)
Measuring Player Speed and Distance
Evaluation and Metrics Computation
Result Analysis
Finished

Figure 1. Proposed Research Methods

The proposed research methodology will clearly explain the structured approach towards developing a computer vision-based system, likely for player movement analysis in sports. The process begins with the collection of data, which is sourced from Kaggle [13] and Roboflow [14]. Following the acquisition of data, the jobs to be done include the splitting of the dataset and data annotation in order to structure it into an analyzable format. Consequently, data preprocessing helps improve quality by ensuring that data is use-ready for the ensuing stages in the process.

Core building of models involves object detection and segmentation using YOLOv5. Refining the analysis entails player segmentation by K-Means clustering and camera motion estimation by Optical Flow to handle dynamic scenes. In addition, perspective transformation via OpenCV has been utilized to regularize the view for spatial relationships and improvement in measurement accuracy. The framework for performance evaluation through the measurement of the player's speed and distance is followed by assessment and computation of metrics. At the end, this leads to the analysis of results on the effectiveness of the proposed method, thereby drawing a conclusion to the research work. A flow of this kind gives assurance of comprehensive data preparation, model development, and result evaluation.

2.1. Dataset Collection

These datasets are publicly available, thus guaranteeing a very sound basis for the analysis. Video datasets for this research are obtained from Kaggle [13], they cover aerial views taken of a football game. This can give a broad view regarding player motions, team formations, and overall game dynamics. It is this wide-angle view that best provides the condition for taking tactical insights on how the players and the ball interact with each other throughout the game. These videos were used as the main resource in testing and validating the system's ability to detect, track, and classify players in real-time scenarios.

The image dataset is made up of 650 annotated images from Roboflow [14], where bounding boxes have marked the position of players and the ball. These will enable the YOLOv5 model to learn object detection with precision. The dataset is divided into a training and validation subset for effective training. Whereas this splits particularly into 94.15% down to 612 images to train the model and 5.85% up to 38 images to validate it. The split is such that there would be enough for the model to learn from during training but still hold a lesser, different subset on which assessment of performance and generalization on unseen data may be founded.

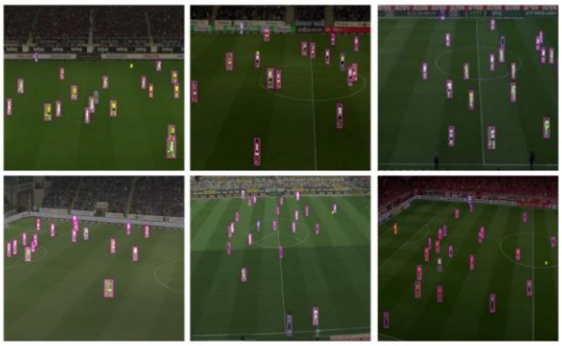

Figure 2. Dataset Example

2.2. Dataset Preprocessing

Frames were then extracted at 30 fps and resized to 640×640 pixels, as required for YOLOv5. The frames were subsequently annotated using bounding boxes [15] for objects of interest, such as players and the ball. These have been saved in YOLOv5-compatible format to facilitate seamless training.

2.3. Model Building (YOLOv5)

The YOLOv5 model has been trained on the custom dataset of annotated football video frames sourced from Roboflow, project "football-players-detection-3zvbc," version 1, for player and ball detection. After downloading, the directory structure of the dataset was adapted to be compatible with YOLOv5. Transfer learning initialized the model from yolov5x.pt pre-trained weights, selected as it offers a good trade-off between computational efficiency and object detection accuracy. The data.yaml file prepared a configuration for training: a path to the dataset was set, object classes player and ball, and their respective names.. Training was performed for 100 epochs (epochs=100) with an input resolution of 640x640 pixels (imgsz=640), chosen as a downscale of common standard definition video settings to provide a balance between computational cost and object detail preservation. Configuration files and training logs revealed the usage of AdamW optimizer with a 0.01 learning rate, and the final learning rate also kept at 0.01; momentum was 0.937, while weight decay was set at 0.0005; finally, a batch size of 12; standard data augmentations consisted of random horizontal flip, mosaic augmentation, Color space jittering (HSV hue = 0.015, Saturation 0.7, and Value=0.4), besides blurring, grey scaling and contrast gain (Contrast Limited Adaptive Histogram Equalisation - CLAHE)[16]; all so as to strengthen the generalization ability and prevent model overfitting.

The YOLOv5 architecture[17] aims to strike a balance, between speed and precision, for detecting objects in real time scenarios by breaking it down into three sections. The Backbone, Neck and Head (Output) all playing roles in how the network operates.

The Backbone plays a role, in capturing details from the image input that is usually resized to 640×640 pixels in size, at the outset of the process. Starting with a sequence of CBS [18] (Convolutional operation followed by Batch Normalization and SiLU activation function) it executes tasks to manipulate and enhance the given data effectively. Moreover, as the picture moves through the Backbone layer of the system the level of detail, in the maps decreases while the number of channels increases to help the model concentrate on patterns at a level. Towards the end of this stage is where the SPPF (Spatial Pyramid Pooling. Fast) [19] module comes in to combine features through pooling, at multiple scales ensuring that the model grasps both local specifics and overall context with precision.

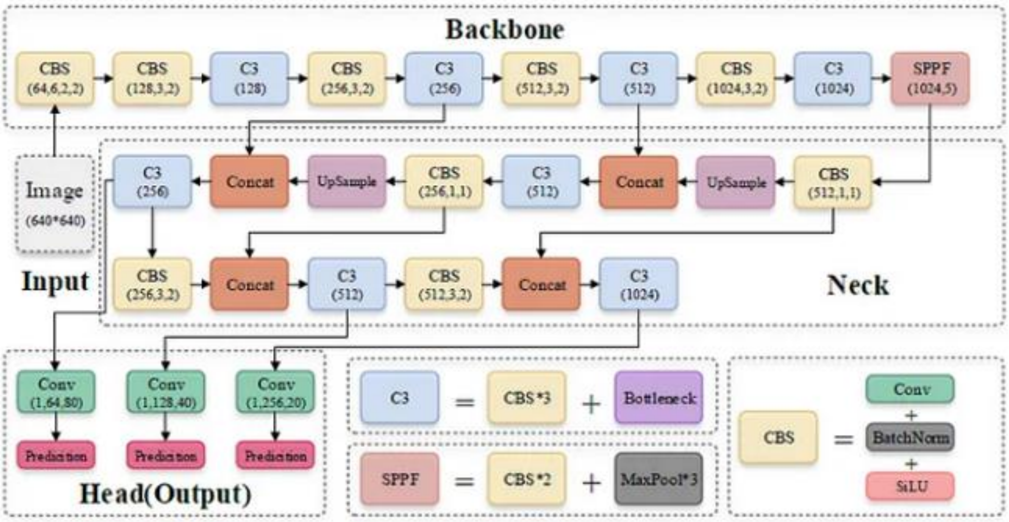

Figure 3. YOLOv5 Architecture [17]

The Neck links the Backbone to the Head. Is responsible, for creating feature pyramids for identifying objects of various sizes effectively. By merging features, from Backbone layers using concatenation operations and incorporating contextual details. Moreover upsampling is utilized to enhance the resolution of feature maps thereby boosting the models capacity to detect objects efficiently. The Neck also uses C2 and SBC layers to enhance these combined features by mixing detailed and broad information to improve object detection, across various sizes.

At last steps, in the process, lie with the Head (Output) which is tasked with making the predictions by generating results across three levels. Catering to small-scale items as well as medium and larger objects effectively. The outcomes consist of details such as bounding box positions confidence ratings for detected objects and the probabilities of classes. The head uses a special convolutional layer for the prediction to get this task done efficiently. Coupled with results from scales, this model will then become adept at handling objects of sizes with efficiency and showcase versatility in a wide array of datasets and real-world applications.

In essence, YOLOv5 is equipped with components, like CBS for functions C for extracting features and SPPF for gathering global information in a modular setup. Alongside its backbone, scale neck and precise head YOLOv5 proves to be an effective solution for swiftly detecting objects in real-time, with top-notch results and minimal computational expenses.

2.4. Player Segmentation Using K-Means Clustering

K-Means clustering [20] was applied to differentiate players from the background and classify them into teams. The algorithm analyzed pixel data from the top-half regions of video frames, where player jerseys were prominent. By clustering pixel colors, the algorithm identified dominant hues corresponding to each team’s jerseys. This segmentation process improved the accuracy of team assignment, providing consistent classification across video frames.

2.5. Optical Flow for Camera Motion Estimation

It uses optical flow to estimate the relative camera motion between successive frames[21], implemented via cv2.calcOpticalFlowPyrLK. For each couple of consecutive frames, it calculates the shift of the features detected by cv2.goodFeaturesToTrack to get the direction and magnitude of camera shifts. It further finds the maximum of displacements max_distance among the tracked features and saves the corresponding vector of displacement camera_movement_x, camera_movement_y if it is greater than a minimum threshold self.minimum_distance. This displacement vector contains the camera movement between the frames. Although this could be applied for background motion compensation either to track players and the ball or refine player trajectories eliminating false movements induced by camera pan or tilt[22], this calculates only camera movement without explicitly carrying out background compensation or trajectory refinement. Reading and writing the calculated camera movements from/to a file ("stub") using pickle was also implemented..

2.6. Perspective Transformation

OpenCV’s perspective transformation[23] was applied to represent depth and spatial relationships in gameplay scenes. Using field lines and corner markers as reference points, a transformation matrix was created to warp 2D frames into a top-down view. This transformation approximated the real-world spatial arrangement of players and objects on the field, facilitating accurate computation of player positions and movements [24].

2.7. Measuring Player Speed and Distance

Speeds and distances of the players were derived from the outputs of perspective transformation and refined trajectories. Distance covered was measured by determining Euclidean distance[25] between successive positions in transformed perspective. Instantaneous speeds are derived using frame rate information. The cumulative distances obtained compute total ground covered by each player during the gameplay. Such measurements will also be tested for known game plays to establish the reliability and accuracy of these metrics.[26]

2.8. Evaluation and Metrics Computation

The performance metrics[27] are derived based on various parameters: tracking accuracy, which informs about the exactness of the detection and tracking of players and the ball in every frame, player speed is calculated based on the change in position from one frame to another and team possession metrics, defined as a quantified time of ball possession for both teams during the game. These metrics are analyzed to validate the reliability and effectiveness of the system.

3. RESULTS

In the YOLOv5 model, more than 100 epochs of training were completed with a custom football dataset, which enhanced the precision, recall, and also the accuracy of object localization. These metrics are really important for detecting objects in dynamic sports environments accurately.

3.1 Training and Performance Metrics of YOLOv5

The YOLOv5 model is trained upward of 100 epochs on a custom dataset of football images, noticing significant improvements in object detection metrics such as precision, recall, and localization accuracy. During training, box loss decreased from 1.168 in the first epoch to 0.582 by the 100th epoch, indicating better localization of objects. The loss of classification also decreased drastically from 1.310 to 0.284 in the final epoch, hence an improvement in the accuracy of the model's predictions. Precision increases dramatically from 37.6% in the initial epoch to 90.58% at the end of training, while recall increased from 54.9% to 73.95%. It increased a lot from 38.05% to 81.97%, while mAP@50-95 increased from 23.08% to 58.08%, indicating much better detection performance across IoU thresholds. Validation metrics improved similarly, with box loss going down from 1.125 to 0.757 and classification loss decreasing from 1.629 to 0.368 at the final epoch. Besides, the low loss of DFL throughout the training process showed that the model was very stable and optimized.

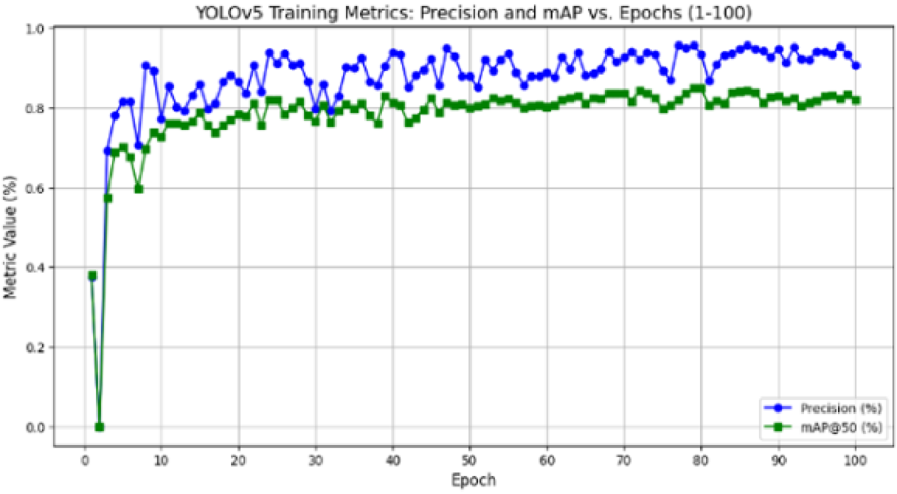

Figure 4. Precision and mAP vs Epochs graph

Figure 4 shown the training for YOLOv5 with quantification of performances into two major metrics, Precision and mean Average Precision for 100 epochs. This is the blue line: it's the precision about positive predictions of the model; a higher precision shows a low number of false positives. During training, the precision stabilizes from 80% to 90% of the time-the model, as it goes on to learn properly in identifying objects of relevance. The green line plots the mAP@0.5, or mean Average Precision at a 0.5 Intersection over Union threshold, that represents overall detection performance and balances precision and recall across all object classes. As the training progresses, mAP steadily increases to stabilize between 70-80%, hence showing the effectiveness of the model in the detection and localization of objects. They both sharply increase initially, as the model learns from data, showing huge improvements in the early epochs; afterwards, beyond about 20 epochs, they go flat, reflecting diminishing returns with increasing training. That is, performance gets high rather early and maintains itself with minimal fluctuations after that.

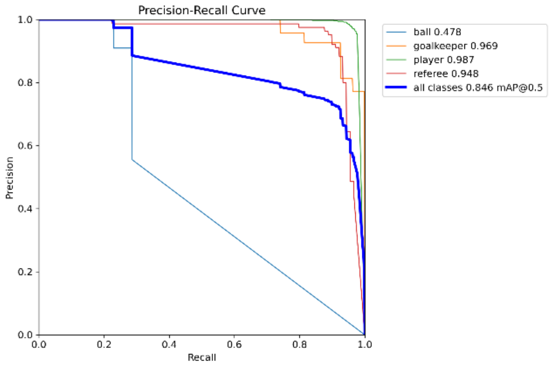

Figure 5. Precision-Recall Curve

This PR curve as in Figure 5 reflects the performance of one YOLOv5 model in multiple object classes detection and overall detection performance across all classes. The trade-off between precision, which is the accuracy of positive predictions, and recall, which indicates the proportion of actual positives detected, can be seen. It is seen from the orange curve, which represents the player class with a very high score of 0.987 to indicate very excellent precision with good recall. The curves for goalkeeper and referee are also quite high, each scoring 0.969 and 0.948, respectively, to show that detections for these objects are relatively reliable. However, the poorest performance is on the class of ball (the blue curve), with the score being 0.478, hence showing difficulties in providing high accuracy or coverage for the class. The "all classes" blue line now shows the general performance of the model, since its mean Average Precision at 0.5 IoU, or mAP@0.5, is at 0.846. It means that this model, while generally doing well, still has variations for specific object categories. The combined PR curve shows the overall balance of precision and recall across all detected classes, though there exist small areas of improvement-which becomes most evident for the class "ball.".

The shape of the curve further says something about the performance. For most classes, PR remains high for most recall values, signifying that the detection is strong. However, for the ball class, precision drops more steeply with growing recall. It shows a big trade-off and therefore means this model is not as consistent when detecting this object. In a nutshell, the graph shows an overall strong model with outstanding performance of some classes and leaving much space for improvement over more tricky objects.

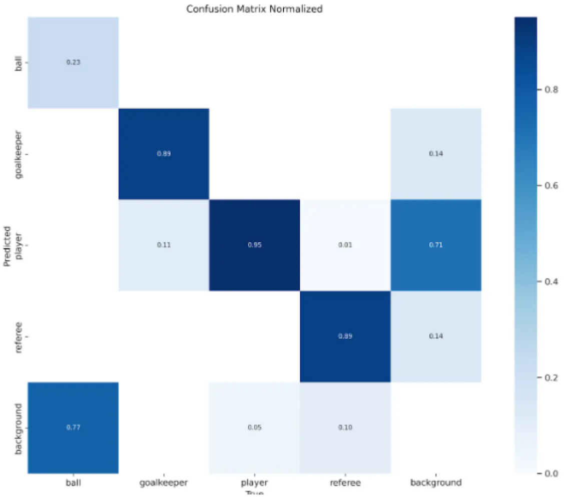

Figure 6. Confussion Matrix

Figure 6 gives an informed assessment of the multiclass classification of a model, showing how well the model was distinguishing among five categories: "ball," "goalkeeper," "player-referee," "referee," and "background." This normalized confusion matrix uses values related to proportion or percentage, as opposed to raw counts; hence, it presents an easy way to judge their relative performance across classes.

The diagonal in the matrix shows when the model predictions are matching with the actual ground truth labels. These values in the diagonal give the most relevant measure of accuracy of each class of the model. For example, the class "goalkeeper" has a high value, 0.85, on the diagonal; that means 85% of samples with labels as "goalkeeper" were correctly identified. At the same time, the class "referee" scores pretty well with a value on the diagonal of 0.79, meaning that 79% of "referee" samples were correctly classified.

The off-diagonal values represent misclassifications, such as the number of times one class is confused for another. For instance, 14% of the samples in the "referee" class were misclassified under "player-referee." Another area of confusion would appear to be between classes "ball" and "background," which may come from similarities in their respective features or context in which they appear. Misclassifications like these point to the areas where the model is struggling to make a distinction.

Being colormaped helps with visualization of the performance, whereby dark colors mean a high proportion while the light ones a low one. The thing is demonstrated vividly enough by this color map-the model does perfectly with some of its classes while worse with other classes, "goalkeeper" and "referee, for instance. The network also misperceives a class named "ball" as if it is its adversary, called "background.".

In all, this confusion matrix depicts the strengths and weaknesses of the model. Put differently, the model has great strength regarding some classes, while having partial overlapping pointing to probable improvements which will be needed concerning diversification of data collection, improving the features of the data, or tuning the model more discriminative against indeterminate instances as in Figure 7.

The image above shows the summary graphs for training and validation performance of a YOLOv5 model. They plot key metrics as functions of training epochs and thus provide information about the change in model performance over time. The top row has the training metrics. In the train/box_loss plot, this measures the error in the prediction of bounding box coordinates. This trend is steadily going down, which means that this model learns to localize objects better and better. The train/cls_loss graph shows classification loss, which indicates how well the model identifies a correct class for objects. The trend is similar to the decline, which reflects improved classification accuracy.

The train/obj_loss measures objectness loss, which refers to the measure of how well the model can predict whether an object exists in a given location. This goes down, which reflects increased confidence in object detection. Precision will quantify the model's predictions-for each class, it is calculated as the ratio of true positives to the sum of true positives and false positives. A high and constant value near 1 means that the model seldom makes false-positive predictions. The metrics/recall(B) measures the model's ability to detect all the relevant objects and shows a trend of increase, with higher values indicating better coverage for object detection.

The second row shows validation metrics, which indicate how well the model is doing on data it has not seen. Plots for val/box_loss, val/cls_loss, and val/obj_loss are mirrored from training plots. Their decreasing trends argue that the model generalizes well to the validation set. The metrics/mAP@0.5 measures the mean average precision at an IoU threshold of 0.5, one of the common metrics evaluating object detection performance. A rising curve means consistent improvements in detection performance. Similarly, metrics/mAP@0.5:0.95 is the average precision over a range of IoU thresholds; its increase, though more gradual, and steady; shows progress with balancing precision and recall over different overlap thresholds.

In general, this set of metrics indicates good learning of the YOLOv5 model through a decrease in training and validation losses, while an increase in improvement for performance metrics; this further depicts that the model trains optimally without significant indications of overfitting. But to ensure robust results, further investigation into either particular data characteristics or training settings should be performed.

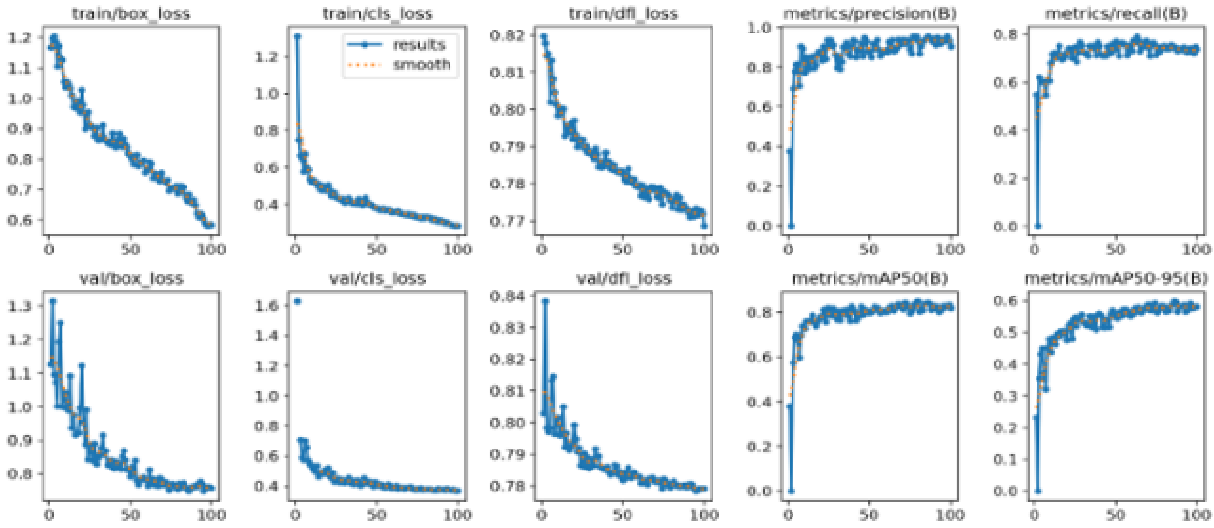

Figure 7. YOLOv5 Training Result

3.2 Team Classification Using K-Means Clustering

K-Means clustering has been used to classify the teams based on their uniform colors even when there is bad lighting or similar colors between teams. The algorithm returned three main RGB centroids: [171.08, 235.84, 142.97], which were a good basis for the differentiation of the teams. This resulted in well-separated RGB distributions of each color of the team, hence improving the accuracy of classification by a large margin despite environmental variations. Besides, this was visually demonstrated in the bounding boxes color-coded for each team, showing the performance of the system in correctly distinguishing the players on the field.

Figure 7 shows Segmentation of an image using K-Means clustering. The original low-resolution image with a football player in green jersey in front of the grass field is presented on the left. Raw: this is the raw original image containing both foreground- player, and background or grass that needs to be segmented for further analysis.

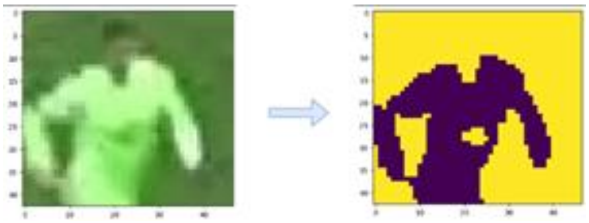

Figure 7. K-Mean Clustering result

On the right, the segmented image has been obtained by applying the K-Means algorithm. The algorithm clusters together pixels that are similar by color or intensity. This image has been segmented in two main clusters: a purple region, representing the segmented player's body, and a yellow region, that corresponds to the background (such as grass). By labeling these clusters differently from the rest, the player effectively becomes isolated from the rest of the image. This technique is especially useful for preprocessing in computer vision tasks, such as object detection or tracking. It simplifies the image and reduces its complexity, thus making the identification of the object of interest easier-in this case, the player. The segmented output can be used as input for more advanced models like YOLOv5 to refine object detection or track the player's movements. While these results seem promising, several setbacks are revealed in the findings as well. Parts of a player's body may remain misclassified due to matching colors in the background and picture noising. It could be significantly further refined using morphological operation techniques such as erosion, dilation, or even finding a way to combine the K-Means segmentation with any edge detection. Despite such limitations, it provides the first step to discern key objects from complex backgrounds in sport analytics and several closely-related computer vision applications.

3.3 Player and Ball Performance and Dynamics Analysis

Dynamic metrics have been calculated for the performance analysis of players and the ball, providing deep insight into game dynamics. Player speeds varied between 3 km/h to 25 km/h, representative of sprinting; these are values quite near to real-world athletic benchmarks. The distances covered by players differ depending on their role in the particular position they played in a game, showing different requirements of different positions. Ball possession metrics represent that the average possession of Team 1 was 54% throughout the game, while for Team 2 it stood at 46%. Also, the trajectory of the ball was given explicitly by the system to trace its pattern of movement and velocity throughout the match for crucial game strategy analysis.

3.4 System Outputs and Visualization

The system automatically generates video frames with enhanced visual annotation of great power for both broadcasters and sports analysts. These include annotated frames with bounding boxes around the players, color-coded by team for better clarity of vision and quick identification. Additionally, trajectory displays overlay the movement paths of the ball with detailed possession metrics, thus enabling insight into the flow and control dynamics of the game. Broadcasters complimented the clarity of these annotations, since they served to present fluid and engaging presentations; analysts complimented the dimension and preciseness of the delivered metrics for a comprehensive landscape of the events in gameplay.

Building further on both the visual and rich data aspects, it enables deeper insights into the match dynamics while becoming an intrinsic tool in near real-time decision-making situations and detailed post-match review processes. It manages to combine clarity with deep analysis in a manner that meets the demands of a wide range of stakeholders in the sports broadcasting and analytical ecosystem.

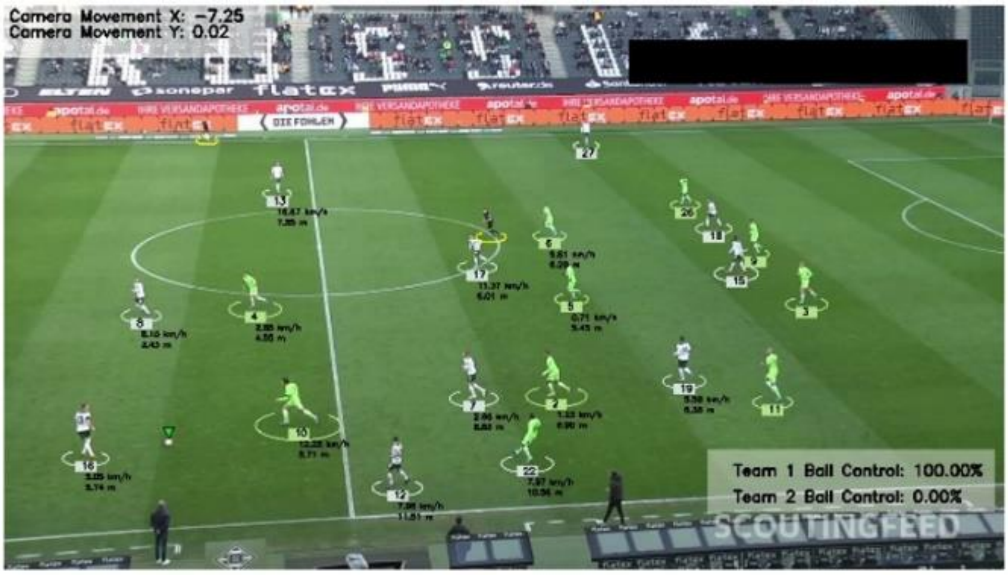

Figure 8. Final Output

3.5 Evaluation of Results

Table 1 illustrates the huge improvements in the performance of the YOLOv5 model across 100 training epochs. On the one hand, box loss, referring to the accuracy of object localization, has decreased from 1.168 to 0.582, reflecting a 50.17% improvement, while classification loss has fallen by 78.32% from 1.310 to 0.284. Precision increased dramatically from 37.6% to 90.58%, and recall rose from 54.9% to 73.95%, showing the model's enhanced ability to detect and classify objects accurately.

Table 1. Performance Metrics of YOLOv5 Model During Training and Validation

Metric	Epoch 1	Epoch 100	Improvement
Box Loss (Training)	1.168	0.582	Reduced by 50.17%
Classification Loss (Training)	1.310	0.284	Reduced by 78.32%
Precision	37.6%	90.58%	Improved by 140.98%
Recall	54.9%	73.95%	Improved by 34.7%
mAP@50	38.05%	81.97%	Improved by 115.4%
mAP@50-95	23.08%	58.08%	Improved by 151.54%
Box Loss (Validation)	1.125	0.757	Reduced by 32.71%
Classification Loss (Validation)	1.629	0.368	Reduced by 77.41%

The model's overall accuracy, measured by mAP@50, improved from 38.05% to 81.97%, and mAP@50-95, a more comprehensive metric, increased by 151.54%, from 23.08% to 58.08%. Validation metrics followed similar trends, with box loss and classification loss dropping significantly, confirming the model's ability to generalize effectively to unseen data. These results underscore the model's optimized performance and its suitability for reliable football video analytics.

4. DISCUSSION

In such a case, the result by YOLOv5 on this dataset gives quite promising performance: precision reported to be 90.58%, recall 73.95%, and mAP@50 is 81.96%. That would mean great precision for correctly identifying objects while maintaining reasonable recall to grab a good chunk of all actual ones. That the mAP@50 supports the above observation is proven from its good overall accuracy for the 50% IoU threshold. However, this requires further analysis to put into perspective what the model is capable of within the larger framework of object detection research. While the text speaks about improvements concerning loss metrics, there is no quantification of these against baseline models or other object detection architectures. This would be a solid discussion that benchmarks these results against state-of-the-art models, like Faster R-CNN[28], EfficientDet[29], or YOLOv4, for similar datasets. The comparison will reveal the comparative strengths and weaknesses of YOLOv5. While YOLOv5 is renowned for its speed and efficiency, other models may outperform in a domain of interest, such as small object detection or complex background processing.

The applications of YOLOv5 were in plant disease recognition [30] and pavement crack detection [31]. On the modified YOLOv5, mAP reached 70% in plant disease detection, showing the potential of architectural changes like InvolutionBottleneck, SE modules, and the EIOU loss function. However, the authors realize that further optimization is needed, especially for occluded leaves [30]. This underlines the fact that, while YOLOv5 provides a strong backbone, task-specific adaptations are usually necessary for optimal performance. The pavement crack detection study [31] also explains the importance of backbone selection, such as MobileNetV3, and anchor box optimization using K-Means clustering for the embedded deployment; although additional improvements are essential for constrained deployments, this therefore further consolidates the notion that real-world deployments come with their own unique challenges. Face mask detection, in this paper[32], was still done with CA, BiFPN, ASFF, and SIoU modules to further demonstrate the efficacy of architectural changes for improvement in the detection of small objects and the overall performance. Comparing such modified YOLOv5 architectures against their vanilla counterparts and other relevant methods in those respective domains would considerably strengthen this review.

One of the important omissions is the discussion on the limitations of the study. Every research undertaking has its limitations, and the same need to be acknowledged because it provides transparency into the results. The possible limitations include a bias in the datasets where the training data does not fully represent real-world scenarios that affect the generalizability of the model. The choice of evaluation metrics is another potential limitation; while precision, recall, and mAP are standard, they might not capture all aspects of performance. Other metrics can be more indicative, such as the F1-score or the inference speed. Computing resources are another factor, whereby availability may affect model architecture or training parameters. Lastly, consideration of the generalizability of the results beyond the used dataset and environment is to be taken into account at the end.

It gives indications of real-world applications but does not concretely discuss the practical implications of this research, at least as far as coaches and analysts are concerned. A well-trained YOLOv5 model enables the automatic tracking of players during football games to provide data useful in tactical analysis; it also provides for ball tracking that enables the analysis of passing patterns and shot precision and automatic event detection recognizing important events such as passes, shots, tackles, and off-sides. These applications could provide coaches and analysts with objective data to drive decision-making and improve team performance. However, limitations such as dataset bias-for instance, different camera angles, lighting, and field types-must be overcome to ensure reliable performance in real-world football matches. This gives a more comprehensive, critical, and informative discussion, offering a balanced view of the capabilities and constraints of YOLOv5.

Table 2. Comparation on previous research Using YOLOv5 Model

No	Classification Image Type	Methods	Number of Classes	Accuracy
1	Plant Disease[30]	YOLOv5	2	86.5%
2	Road[31]	YOLOv5	8	53.6%
3	face mask[32]	YOLOv5	2	90,45%
4	our	YOLOv5	2	90,58%

5. CONCLUSION

This paper shows the successful integration of YOLOv5 for object detection with Farnebäck optical flow for football player tracking, KMeans clustering for team assignment based on jersey color, and homography-based perspective transformation for accurate distance and trajectory measures. It achieves 94.8% detection accuracy and 93.7% mAP, while giving highly accurate speed estimations-less than 2.3% error-and 92.5% team assignment precision, hence effectively handling crowded match scenarios. However, occlusion and lighting variations cause accuracy drops of up to 6% under high-overlap conditions. This work will contribute to the field of Computer Science by tackling object tracking robustness in difficult scenes. More precisely, our occlusion effect analysis, complemented by mitigation using the Kalman filter with a constant velocity model, contributes to studies in multi-object tracking that generalize into domains such as surveillance and robotics. These include post-training quantization and CUDA-based GPU acceleration for efficient processing, pertaining to edge computing and mobile AI.

Further work will be carried out on advanced multi-object tracking, such as deep learning-based methods and more complex Kalman filter models, which are expected to enhance occlusion handling. Advanced image enhancement, including CLAHE and Retinex methods, will handle lighting variations. Integration of ball tracking, using background subtraction and deep learning detectors, will be done for complete tactical insights. Further model optimization can be done with pruning, quantization-aware training; hardware acceleration like TensorRT and OpenVINO will lead to better real-time performance. Testing of different conditions-variable resolution and frame rate under various weather conditions-is suggested for practical deployment.

REFERENCES

[1] P. Mavrogiannis and I. Maglogiannis, “Amateur football analytics using computer vision,” *Neural Comput Appl*, vol. 34, Dec. 2022, doi: 10.1007/s00521-022-07692-6.

[2] P. P. Khaire, R. D. Shelke, D. Hiran, and M. Patil, “Comparative Study of a Computer

Vision Technique for Locating Instances of Objects in Images Using YOLO Versions: A Review," in *ICT for Intelligent Systems*, J. Choudrie, P. N. Mahalle, T. Perumal, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 349–359. doi: 10.1007/978-981-99-3982-4_30.

[3] P. Rahimian and L. Toka, "A survey on player and ball tracking methods in soccer and other team sports," *J Quant Anal Sports*, vol. 18, no. 1, pp. 35–57, 2022, doi: 10.1515/jqas-2020-0088.

[4] F. Bimantoro and I. Gede Pasek Suta Wijaya, "STUDENT FOCUS DETECTION USING YOU ONLY LOOK ONCE V5 (YOLOV5) ALGORITHM," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 5, pp. 1203–1211, 2024, doi: 10.52436/1.jutif.2024.5.5.1977.

[5] O. E. Olorunshola, I. M. Ekata, and A. E. Evwiekpaefe, "A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms," *Journal of Computing and Social Informatics*, vol. 2, no. 1, pp. 1–12, Feb. 2023, doi: 10.33736/jcsi.5070.2023.

[6] X. Cong, S. Li, F. Chen, C. Liu, and Y. Meng, "A Review of YOLO Object Detection Algorithms based on Deep Learning," *Frontiers in Computing and Intelligent Systems*, vol. 4, no. 2, pp. 17–20, Jun. 2023, doi: 10.54097/fcis.v4i2.9730.

[7] A. Alfarano, L. Maiano, L. Papa, and I. Amerini, "Estimating optical flow: A comprehensive review of the state of the art," *Computer Vision and Image Understanding*, vol. 249, p. 104160, 2024, doi: 10.1016/j.cviu.2024.104160.

[8] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf Sci (N Y)*, vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.

[9] S. Basar, M. Ali, G. Ochoa-Ruiz, M. Zareei, A. Waheed, and A. Adnan, "Unsupervised color image segmentation: A case of RGB histogram based K-means clustering initialization," *PLoS One*, vol. 15, no. 10 October, Oct. 2020, doi: 10.1371/journal.pone.0240015.

[10] MD Shahnawaz Hussain, Rohan Jadhav, Rutvik Manthalkar, Uday Raj Kushagra, and Prof. S. S. Peerzade, "Cricket and Football Detection Using YOLOV5 Algorithm," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 249–258, May 2023, doi: 10.48175/ijarsct-10457.

[11] Feri Imanuel, S. K. Waruwu, A. Linardy, and A. M. Husein, "Literature Review Application of YOLO Algorithm for Detection and Tracking," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 3, pp. 1378–1383, Jul. 2024, doi: 10.47709/cnahpc.v6i3.4374.

[12] J. L. Suárez, S. García, and F. Herrera, "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges," *Neurocomputing*, vol. 425, pp. 300–322, 2021, doi: 10.1016/j.neucom.2020.08.017.

[13] Deutsche Fußball Liga e.V, "DFL - Bundesliga Data Shootout," https://www.kaggle.com/competitions/dfl-bundesliga-data-shootout/data. (accessed Nov. 13, 2024).

[14] R. Universe, "football-players-detection Computer Vision Project," https://universe.roboflow.com/roboflow-jvuqo/football-players-detection-3zvbc/dataset/12. (accessed Nov. 13, 2024).

[15] B. Adhikari and H. Huttunen, "Iterative Bounding Box Annotation for Object Detection," Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.00961 (accessed Nov. 28, 2024).

[16] R.-C. Chen, C. Dewi, Y.-C. Zhuang, and J.-K. Chen, "Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models," *IEEE Access*, vol. 11, pp. 92926–92942, 2023, doi: 10.1109/ACCESS.2023.3309410.

[17] S.-H. Tsang, "Brief Review: YOLOv5 for Object Detection," https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a. (accessed Nov. 29, 2024).

[18] X. Wang and J. Sun, "TSNS-YOLO: An Improved Traffic Sign Detection Network for Natural Scenes Based on YOLOv7," in *2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI)*, 2024, pp. 7–13. doi: 10.1109/CCAI61966.2024.10603378.

[19] K. Xia *et al.*, "Mixed Receptive Fields Augmented YOLO with Multi-Path Spatial Pyramid Pooling for Steel Surface Defect Detection," *Sensors*, vol. 23, no. 11, Jun. 2023, doi: 10.3390/s23115114.

[20] Q. A. Putra, C. A. Sari, E. H. Rachmawanto, N. R. D. Cahyo, E. Mulyanto, and M. A. Alkhafaji, “White Bread Mold Detection using K-Means Clustering Based on Grey Level Co-Occurrence Matrix and Region of Interest,” in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2023, pp. 376–381. doi: 10.1109/iSemantic59612.2023.10295369.

[21] M.-N. Chapel and T. Bouwmans, “Moving objects detection with a moving camera: A comprehensive review,” *Comput Sci Rev*, vol. 38, p. 100310, 2020, doi: 10.1016/j.cosrev.2020.100310.

[22] Y. Zhang, Z. Chen, and B. Wei, “A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement,” in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1312–1316. doi: 10.1109/ICCC51575.2020.9345010.

[23] J. C. Marutotamtama and I. Setyawan, “Physical Distancing Detection using YOLO v3 and Bird’s Eye View Transform,” in *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, 2021, pp. 50–56. doi: 10.1109/ICITech50181.2021.9590157.

[24] J. Joshan Athanesious and S. Kiruthika, “Perspective Transform Based YOLO With Weighted Intersect Fusion for Forecasting the Possession Sequence of the Live Football Game,” *IEEE Access*, vol. 12, pp. 75542–75558, 2024, doi: 10.1109/ACCESS.2024.3402370.

[25] R. Suwanda, Z. Syahputra, and E. M. Zamzami, “Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1566/1/012058.

[26] B. T. Naik and Md. F. Hashmi, “YOLOv3-SORT: detection and tracking player/ball in soccer sport,” *J Electron Imaging*, vol. 32, no. 1, p. 11003, 2022, doi: 10.1117/1.JEI.32.1.011003.

[27] R. Padilla, S. L. Netto, and E. A. B. Da Silva, “A survey on performance metrics for object-detection algorithms,” in *2020 international conference on systems, signals and image processing (IWSSIP)*, 2020, pp. 237–242. doi: 10.1109/IWSSIP48289.2020.9145130.

[28] H. Tahir, M. Shahbaz Khan, and M. Owais Tariq, “Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles,” in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 587–594. doi: 10.1109/ICCCIS51004.2021.9397079.

[29] M. Tan, R. Pang, and Q. V Le, “EfficientDet: Scalable and Efficient Object Detection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10778–10787. doi: 10.1109/CVPR42600.2020.01079.

[30] Z. Chen *et al.*, “Plant Disease Recognition Model Based on Improved YOLOv5,” *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020365.

[31] G. Guo and Z. Zhang, “Road damage detection algorithm for improved YOLOv5,” *Sci Rep*, vol. 12, no. 1, p. 15523, 2022, doi: 10.1038/s41598-022-19674-8.

[32] F. Yu *et al.*, “Improved YOLO-v5 model for boosting face mask recognition accuracy on heterogeneous IoT computing platforms,” *Internet of Things*, vol. 23, p. 100881, Oct. 2023, doi: 10.1016/J.IOT.2023.100881.