Video Analytics in the Field of Athlete Sports

***Abstract*—Video analytics is a rapidly growing field that has enormous potential in the world of sports. Basketball is considered an Elite Athlete sport, when powered through computer vision, advanced computer algorithms, and machine learning techniques, such an approach aids the athletes & coaches to obtain a deeper understanding of the game, identify areas for improvement, and make more informed decisions; sketched via tracked movements, shot selection and game status. This paper aims at establishing such a system that augments the play.**

Keywords— Deep Learning, Computer Vision, YOLO, YOLO-Pose, CNNs, Transfer Learning, FaceNet, Trajectory

I. INTRODUCTION

Video analytics is about to be a game-changer in the field of athlete sports especially, in basketball as it helps coaches and players to analyze player performance, develop team strategies, prepare for games and also scout potential recruits. It provides valuable insights into every aspect of the game, helping teams to improve their performance and achieve better results.

One key area where video analytics can be particularly beneficial in basketball is in player tracking. By analyzing data from multiple cameras positioned around the court, coaches can monitor the movement and positioning of individual players and gain a deeper understanding of their strengths and weaknesses. This can inform decisions around player rotations, defensive strategies, and offensive plays, and ultimately lead to better team performance. Another area where video analytics can be valuable is in the analysis of shooting technique. By analyzing footage of a player's shot, coaches can identify areas for improvement in their mechanics, such as foot placement, release timing, and follow-through. This can help players to refine their technique and increase their shooting accuracy, which can have a significant impact on their overall performance.

A full-fledged video analytics system can track the movements of players in real-time, providing coaches with detailed information about individual player positioning, speed, and agility. This can help coaches develop more effective game strategies and make more informed decisions about player substitutions and line-up changes. In developing such a system, we cling to object classification, pose estimation, face recognition, ball path tracking & prediction, and Foul detection models powered through computer vision and transfer learning

Object detection systems work by using computer vision algorithms to analyze video footage and identify specific features that correspond to the objects of interest. Once these features are identified, the system can track the objects in real-time and generate data respective to their relevant characteristics.

Pose estimation, a module that provides real-time data on the positions, angles, and subjective movements such as jumping, running, and shooting by setting up a skeletal structure/frame for the subject. This data can be used to evaluate a player's performance and identify areas for improvement. For instance, if a player consistently jumps off-balance when shooting, pose estimation data can be used to analyze their technique and provide feedback on how to improve.

Face detection pertains to the usage of algorithms that identify and verify individuals based on their facial features that undergo stages of detection, matching, extraction, and finally recognition.

Ball path tracking & prediction aids the system to develop strategies to intercept or deflect the ball, improving its overall performance, mainly shooting accuracy. By analyzing the trajectory and spin of the ball, coaches and players can identify areas for improvement and adjust their shooting techniques accordingly.

The foul detection aspect provides an upper hand for referees which paves a more accurate and objective way to identify and penalize rule violations. Referees are only human, and they can miss important details or make subjective judgments that may not always be fair or consistent.

The proposed system helps to mitigate these issues by providing a more objective and data-driven approach to identifying and penalizing rule violations. By advanced computing algorithms, we'd be able to analyze player movements and contact to identify instances of illegal physical contact between players, such as pushing, tripping, or blocking. Thereby, reducing human error and improving the accuracy of referee decisions

Transfer learning is a technique used in machine learning that involves taking a pre-trained model and using it as a starting point for a new task or problem. The goal of transfer learning is to leverage the knowledge and insights gained from the pre-trained model to improve the performance and efficiency of the new model.

The concept of transfer learning can be especially useful in situations where there is a limited amount of training data available for the new model, as the pre-trained model can provide a strong foundation of knowledge to build upon. Transfer learning paves the way to address challenges in computer vision, such as object detection, semantic segmentation, and pose estimation.

By leveraging the insights gained from pre-trained models, developers can deploy the trained weights into a lightweight system and also be able to achieve robust

computer vision set targets. The robustness of the system relies heavily on the quality and relevance of the collected data and the effectiveness of the data transformation process which later undergo training through deep learning algorithms.

A. *Extant Makes*

There exist varied Object classification & detection models, such that they are so intertwined and branched into several specialties & some are also core units and are built on top over.

1) Sliding window object detection: A traditional method that involves scanning an image with a fixed-sized window and classifying each window as an object or background. It is computationally expensive and does not scale well to large datasets

2) Convolutional Neural Networks (CNNs): Deep neural networks that have been widely used in image recognition and classification tasks. They work by convolving images with learned filters to extract features, which are then fed into fully connected layers for classification.

3) Region-based Convolutional Neural Networks (R-CNN): Two-stage object detection approach that first generates region proposals and then extracts features using a CNN for each proposal. These features are then used to classify and refine the proposals.

4) Fast R-CNN: an improvement R-CNN that integrates the region proposal and feature extraction steps. It uses a single CNN to extract features from the entire image and then applies a region of interest pooling layer to extract features for each proposal.

5) Faster R-CNN: An extension of Fast R-CNN that replaces the selective search algorithm used for region proposal with a Region Proposal Network (RPN) that shares convolutional layers with the detection network. This makes the process faster and more accurate.

6) YOLO (You Only Look Once): A one-stage object detection technique that predicts bounding boxes and class probabilities directly from the image in a single forward pass of the network. It uses a single convolutional network to predict multiple bounding boxes and class probabilities for each box considered to be a more reliable, fast, and resilient algorithm.

II. YOLO - Object Classification Model

All the headings in the main YOLO (You Only Look Once) A.K.A. State-of-the-art algorithm is a prominent object detection model that can detect multiple objects within an image in real-time. It is a convolutional neural network-based architecture that processes the entire image in one go, unlike other object detection models that perform detection at multiple scales.

YOLO divides the input image into a grid of cells and predicts the probability of objects and their corresponding bounding boxes for each cell. By doing this, it avoids redundant object proposals and is much faster than traditional region-based object detection methods.

The YOLO architecture consists of a series of convolutional layers that extract features from the input image. These features are then passed through a set of fully connected layers to predict the bounding boxes and class probabilities. YOLO predicts bounding boxes as offset values from each grid cell, rather than absolute coordinates. This means that the bounding boxes predicted by YOLO are relative to the cell, and not to the entire image.

YOLO predicts multiple bounding boxes for each grid cell, but it uses non-maximum suppression to remove redundant detections. Non-maximum suppression is a technique that selects the bounding box with the highest probability and suppresses the overlapping bounding boxes with lower probabilities. This helps YOLO eliminate false positives and improve detection accuracy.

A. *Trained Model*

YoloV7, a C3AE: Channel-Channel-Attention-Enhanced CNN for Accurate Multi-scale Object Detection model architecture used for multi-scale object detection in images. C3AE introduces a novel channel-channel attention mechanism that captures spatial dependencies between feature channels and enhances feature representations at multiple scales, there are significant improvements in accuracy and speed compared to existing methods.

Integrated with Attention-Based Multi-Modal Fusion for Multi-Label Image Classification where an image can be associated with more than one label or category. That combines multiple modalities, such as visual features and textual descriptions, using attention mechanisms to give more weight to the most informative modality for each label. The attention mechanism allows the model to focus on the most relevant parts of each modality to predict the label.

The process of data collection and the quality of data fed plays a major role in withholding the stability of the model. Our dataset bags up entities of different lighting conditions, different angles, different age categories, different working environments, and different perspectives. Later, the subjects in the dataset are labelled and split respectively for training. The training session undergoes about 100 epochs with a batch size of 32 through YOLOv7 architecture.

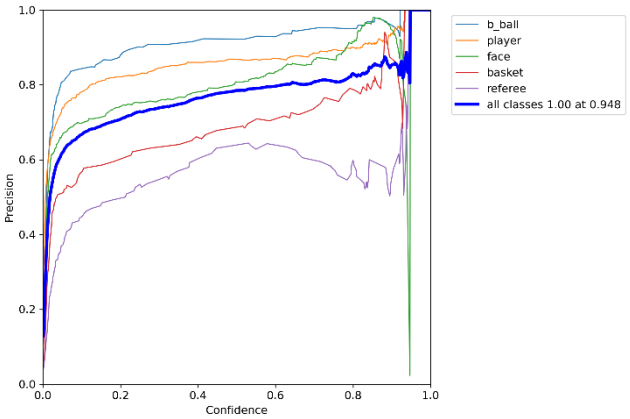

Fig. 1. Precision vs Confidence graph for the trained model

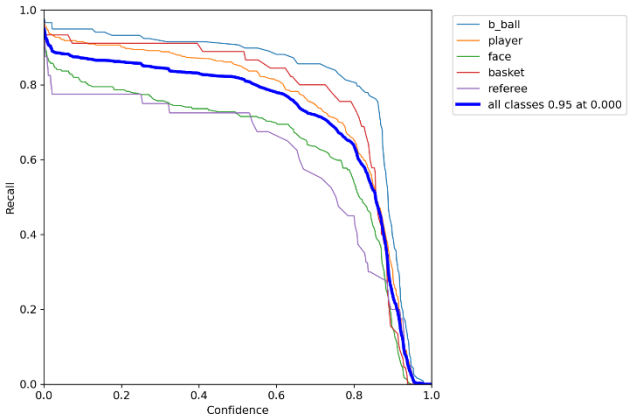

Fig. 2. Recall vs Confidence graph for the trained model

Fig. 3. The trained classification model out

III. Pose Estimation

MediaPipe and YOLO-Pose are two popular deep learning-based frameworks for pose estimation, which involves detecting and tracking the human body and its movements from the frames.

MediaPipe being an open-source framework offers a suite of pre-built solutions for various computer vision tasks, including pose estimation. It uses a combination of convolutional neural networks (CNNs) and geometric models to estimate the pose of a person in real-time. One of the main advantages of MediaPipe is its accuracy, which is achieved through the use of multiple CNNs trained on large datasets. MediaPipe also provides a user-friendly interface and can be easily integrated into existing applications.

YOLO-Pose is a framework that combines the YOLO object detection framework with a pose estimation model built on top of YOLOv5 architecture. YOLO-Pose uses a single neural network to perform both object detection and pose estimation simultaneously, which makes it faster and more efficient than MediaPipe. YOLO-Pose is faster and requires less memory than MediaPipe, making it more suitable for real-time applications with limited computing resources.

A. YOLO-Pose Model

The working of YOLO-Pose is staged by the following when a model is fed through a pre-trained weight:

1) Pre-processing: In this stage, the input frame is preprocessed to prepare it for object detection and pose estimation. This involves resizing the image to a fixed size, normalizing the pixel values, and converting the color space if needed.

2) Object Detection: YOLO-Pose uses the YOLOv5 object detection model to detect the human body in the input image. The model is a fully convolutional neural network that takes the input image and predicts a set of bounding boxes around objects in the image along with their associated class probabilities.

3) Pose Estimation: Once the human body is detected, YOLO-Pose uses a single neural network to estimate the pose of the person. The pose estimation model is based on the OpenPose architecture, which uses a multi-stage CNN to estimate the 2D coordinates of body joints. The network takes the cropped image of the detected person as input and produces a set of heatmaps that indicate the likelihood of each joint being present at each location in the image. These heatmaps are then used to estimate the 2D coordinates of each joint using a decoding algorithm.

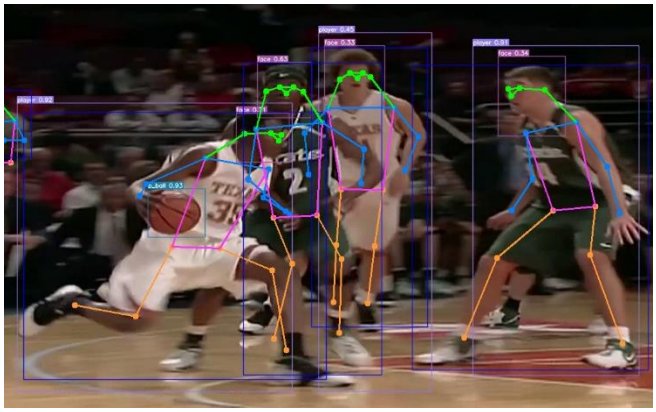

Fig. 4. Rendered through YOLO-pose weight

IV. Face Recognition

Face recognition is a complex process that involves various technical challenges, such as variations in lighting conditions, facial expressions, occlusions, and pose variations. Therefore, the accuracy of face recognition systems depends on the quality of the images, the robustness of the algorithms used, and the size and diversity of the database used for training and testing the system.

A. Flow

1) Face detection: The first step in face recognition is to detect the presence of a face in an image or video frame. This is done using various techniques such as Viola-Jones, Histogram of Oriented Gradients (HOG), and Multi-Task Convolutional Neural Networks. The output of this step is a bounding box that encloses the face. This can be achieved through our trained YOLO model

2) Face alignment: Once the face is detected, it needs to be aligned to a standard position and size to enable accurate feature extraction. This involves normalizing the face's orientation, scale, and position relative to the camera. The alignment can be done using various techniques, such as 2D and 3D face alignment methods.

3) Feature extraction: Unique features are extracted from the aligned face at this stage, which is used to identify and compare faces. Various techniques such as Local Binary Patterns, Scale-Invariant Feature Transform, and Convolutional Neural Networks can be used to extract these features.

4) Face matching: Once the features are extracted, they are compared with the features of known faces in a database to determine the identity of the individual. And our model is achieved through Euclidean_L2 metrics.

5) Recognition: The final step in face recognition involves making a decision based on the similarity score obtained from the matching algorithm. If the similarity score is above a certain threshold, the face is considered a match, and the individual's identity is verified. Otherwise, the face is considered a non-match, and the individual's identity is rejected.

B. FaceNet Architechure

FaceNet is a deep learning model for face recognition that uses a neural network architecture called the Siamese network to learn a mapping from face images to a high-dimensional feature space where distances correspond to a measure of face similarity. The architecture of the FaceNet model bags up the following three main components:

1) Feature Extraction: The CNN in FaceNet is designed to extract high-level features from face images. It takes an input image and generates a high-dimensional feature vector that captures the facial characteristics of the input image. The CNN consists of multiple convolutional layers followed by max-pooling layers that reduce the spatial dimensions of the feature maps.

2) Triplet Loss: The triplet loss function is used to learn a mapping from the input images to the feature space in such a way that the distance between the feature vectors of the same person is minimized, while the distance between the feature vectors of different persons is maximized. The triplet loss function takes three images: an anchor image, a positive image (the same person as the anchor), and a negative image (a different person from the anchor). The goal is to minimize the distance between the anchor and positive images while maximizing the distance between the anchor and negative images. This ensures that the feature space is discriminative and can accurately differentiate between different faces.

3) Face Recognition: The final component of the FaceNet model is a linear SVM classifier that is trained on the feature vectors generated by the CNN. During testing, the SVM classifier takes the feature vector of an input image and predicts the identity of the person in the image by comparing it to the feature vectors of known individuals in a database. The identity of the person with the closest feature vector is returned as the predicted identity.

Our system opts Facenet512, an advanced facial recognition model that has been developed as an extension of the original FaceNet model. The primary difference between the two models is the dimensionality of the feature vectors they generate. While the original FaceNet model generates 128-dimensional feature vectors, Facenet512 generates 512-dimensional feature vectors. This increased dimensionality allows Facenet512 to capture more fine-grained details of facial features, resulting in improved performance in face recognition tasks. The differences in their architectures affect the dimensionality of the feature vectors they generate and the number of neural network parameters they have.

Facenet has around 22 million parameters, while Facenet512 has around 94 million parameters, allowing it to learn more complex representations of facial features. Despite the differences, both models are highly accurate and robust and have been trained on large-scale datasets of face images.

V. Ball Trajectory

The ball trajectory is an essential factor in video analytics in basketball as it provides crucial information on player performance, team strategy, and game outcomes. By analyzing ball trajectories, coaches and players can identify areas for improvement, develop effective game plans, and make better decisions during the game. It also helps to track the movement of the ball and predict its path, providing valuable insights into player and team performance.

A. Ball Path Tracking

The ball path tracking algorithm works by analyzing video footage of a basketball game, detecting the position of the ball in each frame, and using this information to estimate the trajectory of the ball over time. This requires the use of sophisticated computer vision techniques, such as object detection, object tracking, and motion estimation. This involves analyzing the pixels in the frame to identify regions that are likely to contain the ball.

Once the ball's detected, object tracking algorithms are used to follow the ball's movement over time, even as it moves in and out of view or is occluded by other objects. The set plot points of the ball are enqueued in a fixed queue, such that it minimizes the visual complexity. Later used to plot them in the frames.

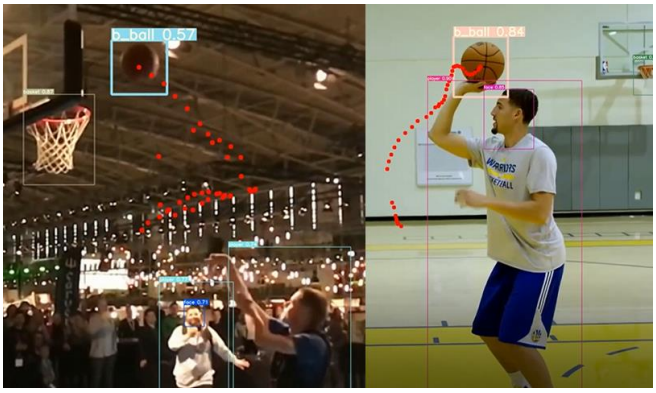

Fig. 5. Tracking the ball path

B. Projectile Motion

Once the ball's trajectory is estimated, it can be used to generate a variety of statistics and visualizations, such as the speed and direction of the ball, the location of the ball at key moments in the game, and the path of the ball as it moves around the court. This involves analysing the movement of the ball in each frame and using this information to predict where the ball will be in the next frame. By combining the information from multiple frames, the ball's trajectory can be estimated with high accuracy.

Ball trajectory prediction, also known as projectile motion involving physics-based models, is an important technique in basketball video analytics that involves predicting the path of the ball as it moves through the air. Projectile motion is a type of motion that occurs when an object is projected into the air and moves under the influence of gravity. The motion of the ball can be modelled using equations that take into account the initial velocity of the ball,

the angle at which it was thrown, and the force of gravity. By combining these equations, the trajectory of the ball can be predicted with high accuracy.

$$Trajectory\ Equation = x \tan\theta - \frac{gx^2}{2u^2 cos^2\theta} \quad (1)$$

To predict the trajectory of the ball, the algorithm first identifies the position of the ball in each frame of the video footage. It then uses this information to estimate the initial velocity of the ball and the angle at which it was thrown. Using these values, the algorithm can then apply the equations of projectile motion to predict the trajectory of the ball. The 3-Dimensional path tracking loads up an enormous level of complexity; Thereby, this system is limited to only a 2-dimensional perspective of the frame.

VI. Foul Tracks

Fouls are infractions committed by players that result in a penalty, such as a free throw or loss of possession. Detecting fouls is an important aspect of basketball video analytics, as it can provide valuable insights into the performance of individual players and teams, and help referees make more accurate calls during games. Fouls can have a significant impact on the outcome of a game, and detecting them accurately is essential for maintaining fairness and impartiality.

The process of detecting fouls in basketball video analytics typically involves the use of computer vision techniques such as object detection and tracking, to identify instances of fouls in the video footage. This involves training a neural network on a large dataset of annotated video footage, where instances of fouls are been labelled by human experts. The neural network can then learn to recognize patterns in the video footage that are indicative of fouls.

A. *Typical BasketBall Fouls*

Basketball, being a contact sport; fouls are a common occurrence. Fouls in basketball are violations of the rules of the game that are committed by players. These violations result in a penalty, such as a free throw or loss of possession.

The following are some of the most common types of fouls in basketball:

1) Personal fouls: These are fouls that involve illegal contact between players. Examples of personal fouls include pushing, tripping, or holding an opposing player.

2) Technical fouls: These are fouls that are not related to physical contact but are violations of the rules of the game. Examples of technical fouls include arguing with the referee, using profanity, or intentionally delaying the game.

3) Flagrant fouls: These are fouls that involve excessive contact and are considered to be dangerous to the safety of the players. Examples of flagrant fouls include striking an opposing player or making unnecessary contact with an airborne player.

4) Offensive fouls: These are fouls that are committed by an offensive player and result in a turnover of possession. Examples of offensive fouls include charging into a defensive player or using an elbow to clear space.

5) Defensive fouls: These are fouls that are committed by a defensive player and result in a penalty, such as a free throw. Examples of defensive fouls include blocking or pushing an opposing player.

B. *Trained Model*

The modelled is extended with a newer dataset bagging a new class – foul. Several image datasets were scrapped from the web and labelled in subject to the fouls which are structured into personal, technical, flagrant, offensive, and defensive fouls in the frames. The pre-processed data is split and fed again into the YOLOv7 architecture for training. When a foul is detected in the rolling frames, the foul is classified & written in a text file with the respective time stamp of occurrence.

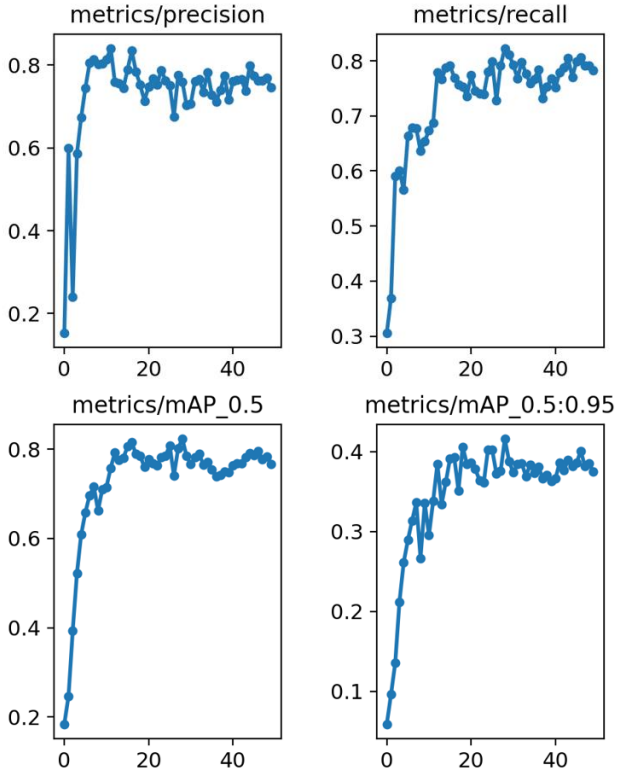

Fig. 6. Resultant out graphs for 50 epochs

VII. Conclusion

Integrating object classification, pose estimation, facial feature extraction, matching & recognition, and involving ball trajectory flow gives out an omnipotent change in the field of sports. With such an approach towards athlete sports, augments the gameplay of the players in several multitudes. The resultant accuracy scores in each case were precise enough to satisfy the industry standards.

VIII. References

[1] Florian Schroff, Dmitry Kalenichenko and James Philbin, "FaceNet: A unified embedding for face recognition and clustering", IEEE, https://doi.org/10.1109/CVPR.2015.7298682, 15 October 2015

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", IEEE, https://doi.org/10.1109/5.726791, November 1998

[3] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", ResearchGate, http://dx.doi.org/10.1145/3065386, January 2012

[4] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE, https://doi.org/10.1109/TPAMI.2016.2577031, 6 June 2016

[5] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", ResearchGate, http://dx.doi.org/10.1109/CVPR.2014.81, November 2013

[6] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", ResearchGate, http://dx.doi.org/10.1109/TPAMI.2016.2577031, June 2015

[7] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato and Lior Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", IEEE, https://doi.org/10.1109/CVPR.2014.220, June 2014

[8] Qiong Cao, Li Shen and Weidi Xie, "VGGFace2: A Dataset for Recognising Faces across Pose and Age", ResearchGate, http://dx.doi.org/10.1109/FG.2018.00020, May 2018

[9] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", IEEE, https://doi.org/10.1109/CVPR.2016.91, 12 December 2016

[10] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement", arXiv, https://doi.org/10.48550/arXiv.1804.02767, 8 April 2018

[11] Xingkui Zhu, Shuchang Lyu, Xu Wang and Qi Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios", arXiv, https://doi.org/10.48550/arXiv.2108.11539, 26 August 2021,

[12] Chien-Yao Wang, Alexey Bochkovskiy and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", arXiv, https://doi.org/10.48550/arXiv.2207.02696, 6 July 2022

[13] Joseph Redmon and Ali Farhadi, "YOLO9000: Better, Faster, Stronger", IEEE, https://doi.org/10.1109/CVPR.2017.690, 9 November 2017

[14] Debapriya Maji, Soyeb Nagori, Manu Mathew and Deepak Poddar, "YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss", arXiv, https://doi.org/10.48550/arXiv.2204.06806, 14 April 2022