OELP PROJECT REPORT

VIDEO ANALYSIS

by:
Jasir K 111901025

Deon Saji 111901022

Faculty Mentor:
Dr. Vivek Chaturvedi

Computer Science and Engineering

Indian Institute of Technology, Palakkad

1.Introduction

Video content analysis is one new technology which uses computer vision and in general different deep learning models. It is used for automatically identifying different events in the videos , monitor video streams in real Time and then extract useful information . This is mainly used in the field of sports,transport,security etc. For security purposes, cameras can be installed in different places and then detect different anomalies. In the sports field, coaches could find the weak point of their opponent and moreover find the technical mistakes made by their own player.

2.Aim of the Project

In this project, we will devise a fast learning algorithm which will be able to learn from analyzing video frames to recognize the player's behavior and techniques and then use that information to give suggestions to improve his/her skills and give suggestions for the areas that the player needs to work upon. These results will aid athletes in identifying and correcting their frequent errors as well as modifying their technique if necessary to achieve better results.

3.Summary of works till mid-sem

The learnings done include hand tracking,pose estimation etc using mediapipe,movenet framework and analyzing different algorithms such as R-CNN,YOLO and its different versions for object detection. Research papers, manuals, presentations were read to understand the previous work on this field and different factors affecting gameplay. Along with that we went through some coach manuals to understand what the rules are and different kinds of shots and its execution and different performance factors such as technique, tactics, physical, psychological and lifestyle. Using YOLOv5s we were able to use transfer learning techniques and train our annotated dataset for object detection of different shots and objects.

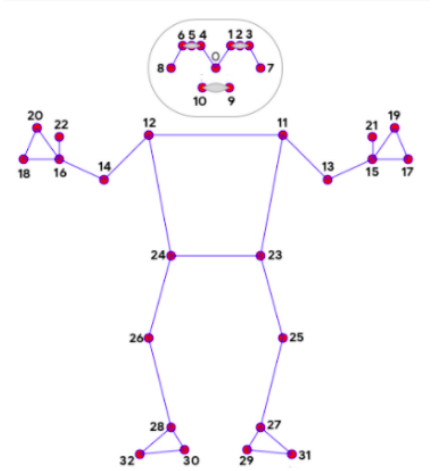
4. Pose detection of players using movenet and posenet

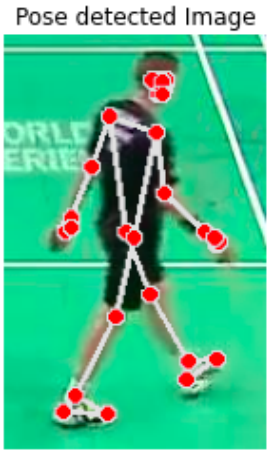
Main aspect we looked at in this project was to look at the pose of each player. Pose detection of players is used in this project for detecting forehand and backhand shots and to detect the foot movements of the player. Which would be helpful in understanding whether the player moves forward or backward, understanding the foot movements and to know whether he/she gets into the right position, and the pose during the shot and different techniques used by the player.
One of the existing libraries we tried was movenet. Movenet is an ultra fast and accurate model that detects 17 key points of a body. This model is offered on TF Hub with 2 variants known as lightning and thunder. We used the thunder mode which is intended for higher accuracy. Even though it is a highly accurate model , the accuracy of landmarks detected for the far person was poor and there were only 17 key points detected. It can detect at a higher fps of 50 but the problem was with accuracy.

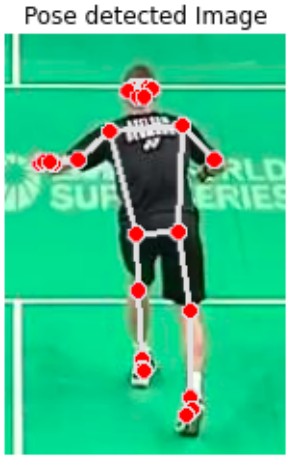
Similarly we tried Posenet, using tensorflow and 17 key points were detected. In the media pipe we can detect 32 key points, in addition to the 17 COCO keypoints and provide key points for the face, hands and feet. So we moved on to mediapipe for pose detection.

5. Pose detection of players using mediapipe

Another framework we tried was mediapipe which detects 32 different landmarks of the human body and returns their coordinates when an image is given as input. One difficulty we faced was that the mediapipe module detects only one player even if there are multiple ones in the same image and therefore shots and movements of only one player were getting recorded. To solve this we initially tried to use sliding window technique where a small window was moved through the image and the portion of the image that was enclosed in this rectangular window was then passed to the mediapipe module. But this method performed very poorly because it took a lot of time in completing even 5-6 frames.

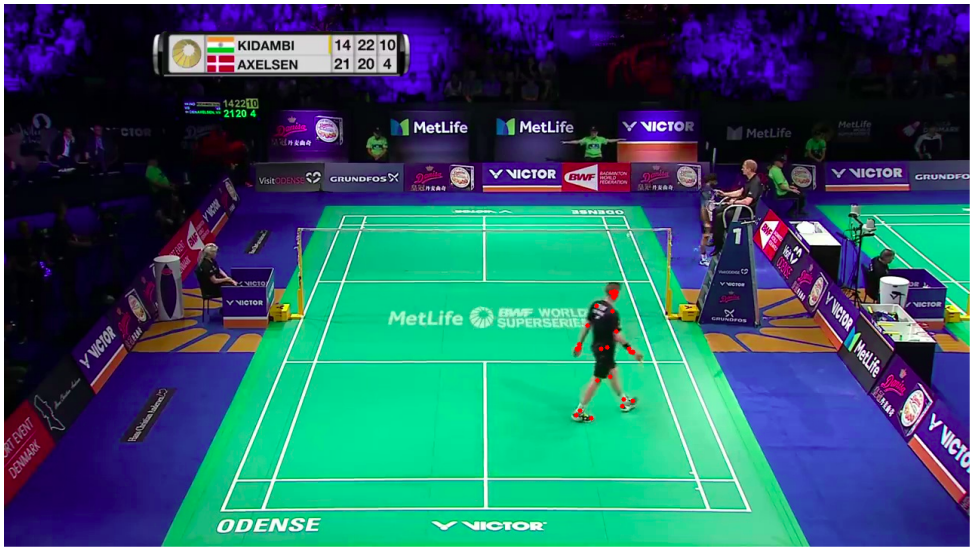






The next method we tried implemented was on the basis of how to improve upon the sliding window technique and the idea was to pass only a few sections of a frame instead of passing all the window of images which was done in the sliding window method. There are multiple players in each frame and the corresponding bounding box coordinates of each player were already stored. Hence the plan was to crop those bounding boxes out from each frame and pass them through the module. The main point here is that the pose coordinates now detected are with respect to the cropped image and not the actual image. Therefore rescaling of the coordinates is important. This was done by extracting the original frame size (let it be w2,h2) ,cropped image size (w1,h1) and let (xmin,ymin) be top left corner coordinates of the original frame, then pose coordinates with respect to original frame was obtained by the formula,

xnew = (x*w1 + xmin*w2)/w2, ynew = (y*h1 + ymin*h2)/h2 where x,y represents coordinates with respect to the cropped image.
Below is the output after rescaling the coordinates



6. Detecting different players

We used a python class for storing all the necessary details associated with players such as the bounding box's coordinates, different landmarks using mediapipe, different shots played by that player, his position in the court, etc. In order to detect different players, we used the net's y-coordinate as a reference line. With the help of the reference line if the player's bounding box is above the reference line then the player belongs to PLAYER_1 otherwise PLAYER_2.

The pose estimation with mediapipe was applied on the video, and with the assumption that the head coordinates would be contained in the player's detected bounding_box, we were able to assign the pose to that player.

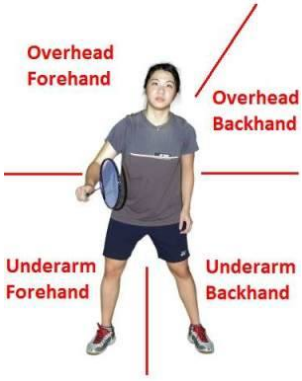
In total 5 shots were detected by the model trained using dataset 2. The shots include Drop shots, Smash shots, Net shots, Over defensive clear and serve shots. Now, with the help of a trained model of dataset2, frequency of the shots is being calculated and stored in a dictionary with python script and running it for various input source videos. With the help of the frequency comparison of various shots gives an idea about the overall gameplay which is played in the source video.

Specifically, the service shots were under-annotated in the initial training, so at a later point we retrained our model using a new set of data.

We took the shots with great confidence in each frame and the detection of shots were made according to frames, therefore we will be counting the same shots multiple times, so at the end we will be taking ratios of different shots frequency to come to a conclusion. This information can be used to compare the game plays and analyze the same.

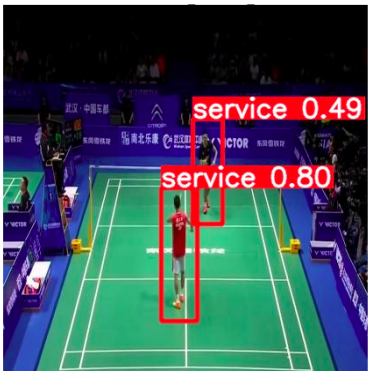
7. Forehand vs Backhand classification

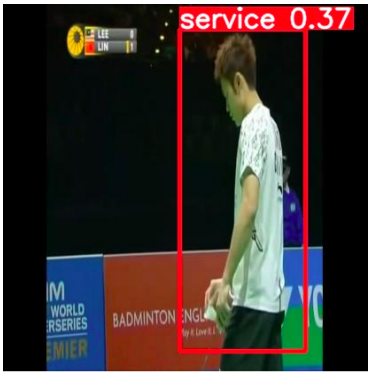
Out of the 32 landmarks that were detected, landmark 0 and 16 were considered because they denote nose and right hand wrist. Here the logic is that if any shot is being played then the right hand is to the right of the center of the body. Taking nose coordinates as center of the body, then the shot is forehand if the wrist is to the right side of the body else backhand. Using this count of forehand and backhand shots are taken and all types of shots (Smash,Drop,Net) are classified as forehand or backhand using the same logic. This logic may not always match and different angles and shots need to be analyzed to get the exact answer. Statistics of forehand and backhand shots played by each player is important to understand how each player responds to their opponent's game and to check whether they are technically playing the right shot.



8. Training serve shot

For more analysis, we found it good to analyze serve shots. Dataset was collected online and was annotated using roboflow. This software helped us to draw bounding boxes for each image and to get the corresponding coordinates so that these images could be trained for serve shot detection. 342 images were annotated and train-test split was done according to 85%-10%-5% ratio. These were then trained using yolov5.



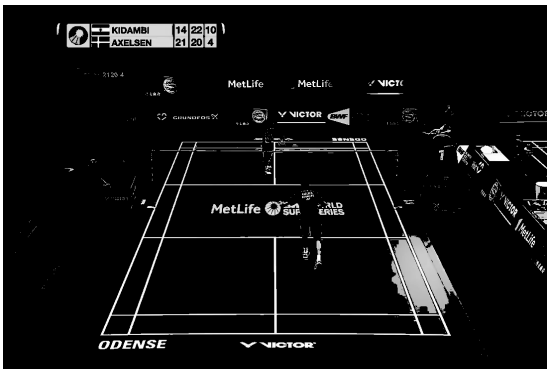


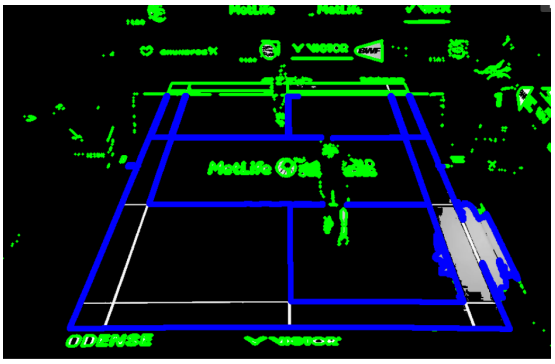
From the above detection, one thing to notice is that even though only one player does the actual serve shot but because both the players have a similar stance in one frame, the receiving player is also getting detected as a serve shot. In some cases, their confidence is low which is a good result.

Frames with a side view and a single person were ineffective for analysis. In order to remove those frames from analysis, we applied the logic of area of the bounding box, i.e. the bounding box of the player is much larger in the frames with side views than in the other frames. As a result, we fine-tuned the parameters and were able to eliminate the frames with side views.

9. Court annotation and detection

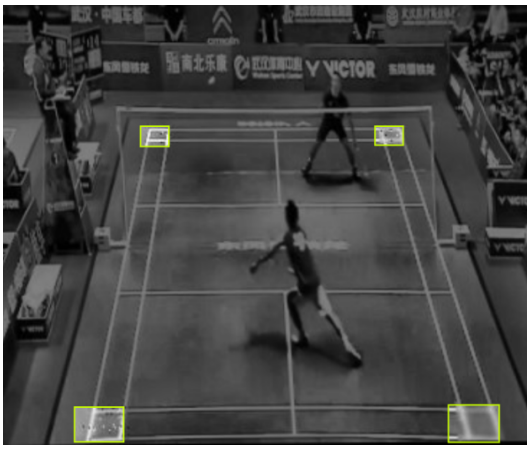
To extract the corners of the court from a video frame, first we thought of extracting the white pixels and with the information of intersection of lines we would be able to detect the corners of the court. At first we converted the image into grayscale and removed the area with white pixels at a larger area, after that a canny edge detection algorithm was applied on it. But the background noise in different frames was too high.

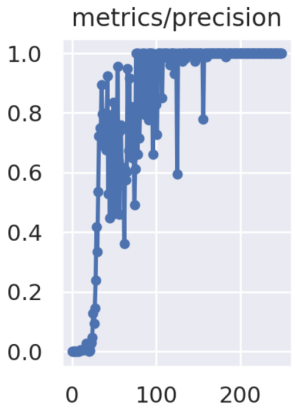


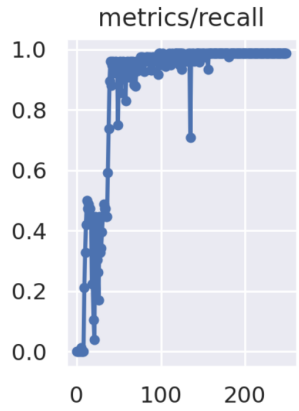


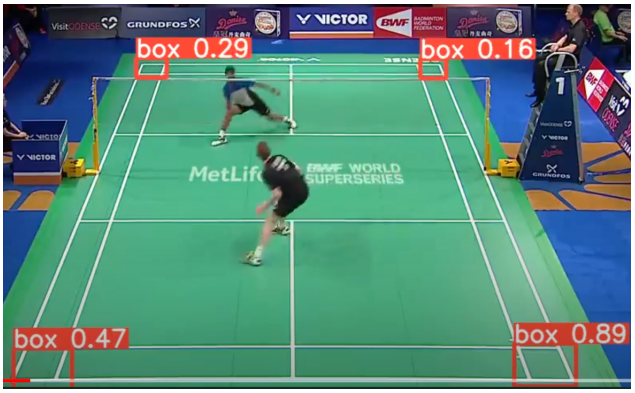
Then we tried by converting the image to grayscale and applying the corner harris algorithm on the resultant image. Then with the help of the github repository ("opencv_wrapper"), we tried to assign a threshold and find external contours in the processed image. But still the edge points detected were too high due to the background noise.

Furthermore, to improve the logic of detecting the corners we thought of annotating and training the images for the corner box in the court. So with the help of that we get the coordinates of the corners of the court considering the extreme top,left, right and bottom coordinates. For this we annotated 151 images along with applying data augmentation using stressing the image, changing the hue, rotating some images for + or - 15 degrees. Coordinates of court were further used for detecting the exact position of the player in the court, whether he was standing at pivot position or not and for further analysis.









In order to get only the court lines with tracking of player, shuttle and racket, it was decided to court detection in video frames for segmentation. So for this part, we helped **another team member** of this project for making of the dataset, in which a total of 2383 images were manually annotated with the help of polygon feature in Roboflow. Then it was trained by that **team member** using the UNET model for segmentation. And using the external contours feature and model that was trained, we were able to to mark the boundary of the court and got the coordinates of the corners.

10.Conclusion

First part of the project was to train different models to detect the players, equipment and shots played by them. This was the most basic step because the bounding box coordinates that were obtained in this step were used throughout the project.

Next step was to count the number of different shots played by each player. This includes the information of shots played one after the other which is important for doing the best action/shot depending on the opponent's move. After this pose detection was done using different frameworks and this is extremely useful to distinguish players and then analyze shots/movements of each player. Further analysis using pose includes detecting forehand and backhand shots. We worked on court detection and segmentation in order to get the coordinates of the corners of the court so that we could detect the court and pivot position.

11. Scope and Future work

These methods can be further implemented for different games and further implemented as a product to give the necessary output if a given input video source is given. Incorporating different IOT devices or special cameras for different angles can improve the accuracy and analysis. We can expand this to a product which can help athletes to coach and train themselves.

12. Bibliography

[1]https://www.yumpu.com/en/document/read/53240814/13-a-comparative-study-of-visual-reaction-time-in-badminton-players-

[2]https://www.researchgate.net/publication/266868606_A_Research_on_Visual_Analysis_of_Badminton_for_Skill_Learning

[3]https://www.researchgate.net/publication/320168613_A_study_of_attention_and_imagery_capacities_in_badminton_players

[4] Mediapipe Documentation, hand tracking and other modules.
https://mediapipe.dev/

[5] Mini Project CV
https://www.youtube.com/watch?v=01sAkU_NvOY

[6] Plotting results,
https://wandb.ai/site

[7] RoboFlow for Annotations and labeling,
https://roboflow.com/

[8] Yolov5 - Custom Data
https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

[9] Movenet Model

https://www.tensorflow.org/hub/tutorials/movenet#:~:text=MoveNet%20is%20an%20ultra%20fast,applications%20that%20require%20high%20accuracy.
[10] Harris corner detection
https://automaticaddison.com/detect-the-corners-of-objects-using-harris-corner-detector/
[11] Contour coordinates
https://www.geeksforgeeks.org/find-co-ordinates-of-contours-using-opencv-python/
[11] Contour coordinates
https://pypi.org/project/opencv-wrapper/
[12] Mediapipe pose code
https://google.github.io/mediapipe/solutions/pose.html