DETECTING, TRACKING AND PREDICTING THE PATH OF AN OBJECT

Adnan Shah[1], Anvitha S V[2], Arun Karthik K[3], Nethra T N[4], Rashmi R Deshpande[5]

[1] Student, Electronics and Communication Dept., AMCEC, Karnataka, India
[2] Student, Electronics and Communication Dept., AMCEC, Karnataka, India
[3] Student, Electronics and Communication Dept., AMCEC, Karnataka, India
[4] Student, Electronics and Communication Dept., AMCEC, Karnataka, India
[5] Associate. Professor, Electronics and Communication Dept., AMCEC, Karnataka, India

ABSTRACT

This can be achieved using various algorithms such as polynomial regression, neural networks, contour detection. Object detection involves identifying the presence and location of an object in an image or video frame. This task can be accomplished using various techniques such as deep learning-based approaches, feature-based approaches, and template matching. Object tracking, on the other hand, involves following the detected object across successive frames in a video sequence. Tracking can be achieved using methods such as particle filters, and correlation-based tracking. Finally, predicting the path of an object involves estimating its future position and movement based on its past trajectory.

Keyword: - *DTR- Detection, Tracking, Prediction.*

---

1. INTRODUCTION

Detection, tracking, and prediction of the path of an object is an important process used in various fields, including surveillance, robotics, and autonomous vehicles. The task of tracking, a crucial component of many computer vision systems, can be naturally specified as an online learning problem [1], [2]. The process involves using sensors or cameras to detect the presence of an object, followed by tracking its movement in real-time. The information obtained during tracking is then used to predict the future path of the object, which can be used to anticipate its behaviour and take necessary actions accordingly. The accuracy of detection, tracking, and prediction is highly dependent on the type of sensors used, the environment in which the object is moving, and the algorithms used to process the sensor data.

The development of advanced machine learning algorithms and computer vision techniques has greatly improved the accuracy of these processes, making them an essential component of modern-day applications that require object tracking and prediction.

The objectives of detection, tracking, and prediction of the path of an object are to accurately detect and track the movement of an object in real-time and to predict its future path based on the information gathered during tracking. These objectives are important because they enable us to anticipate the behavior of an object and take necessary actions accordingly. Pre-trained tracking is a different problem for which the performance not only depends on the frames of videos but also on the training data, an issue we try to avoid here. We also do not consider offline tracking, which allows for global optimization of the path, scanning forwards and backwards through the frames of a video [3][4].

For example, in surveillance applications, detecting and tracking the movement of a suspicious object can help prevent potential security threats. In robotics and autonomous vehicles, accurate object tracking and prediction can ensure safe navigation and collision avoidance. Overall, the main objective of these processes is to provide reliable and accurate information about the movement of objects in real-world scenarios, which can be used to improve safety and efficiency in various applications.

The motive behind detection, tracking, and prediction of the path of an object is to enhance the accuracy and efficiency of various applications that require object tracking and prediction. By accurately detecting, tracking, and predicting the movement of objects, we can improve safety and security in surveillance applications [6], prevent collisions in robotics and autonomous vehicles, and optimize the efficiency of logistics and transportation systems. Additionally, these processes can be used to gather valuable data about the behavior and movement of objects, which can be used to inform decision-making in various industries [7].

Ultimately, the motive behind these processes is to improve our ability to understand and interact with the world around us, making our lives easier, safer, and more efficient.6

2. METHEDOLOGY

The Fig -1 shown below explains the complete flow of the methodology along with the objectives

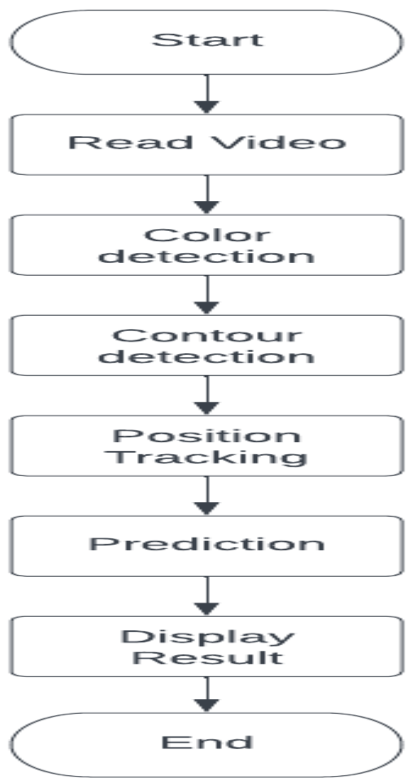

Fig -1: Methodology

2.1 MODULE EXPLANATION

Detection involves using various OpenCV functions to identify objects or events of interest within a given environment. This can be accomplished using various sensors, such as cameras or lidar, and involves analyzing the data received from these sensors to recognize and locate the desired objects or events. OpenCV provides various functions for object detection, such as Haar cascades, which can detect faces, eyes, and other objects, and HOG (Histogram of Oriented Gradients), which can detect pedestrians and vehicles.

Once objects or events have been detected, tracking is used to monitor their movement over time. OpenCV provides various algorithms for object tracking, such as the MeanShift and CamShift algorithms, which can track the position of the detected objects in subsequent frames of the video.

Finally, prediction can be accomplished using various techniques such as polynomial regression, which is a mathematical algorithm that can predict the future states of the detected objects or events based on their current and past positions and velocities.

OpenCV provides a built-in Kalman filtering function, which can be used for object tracking and prediction. Overall, OpenCV provides a wide range of functions and algorithms that can be used to implement detection, tracking, and prediction methodologies, making it an essential tool for developing intelligent systems that can operate effectively in complex environments.

The goal of this program is to track the shooting range of the basketball in the shooting video provided by the data set, predict whether the basketball will successfully fall into the basket according to its initial moving direction, and successfully display the prediction result.

3. implementation

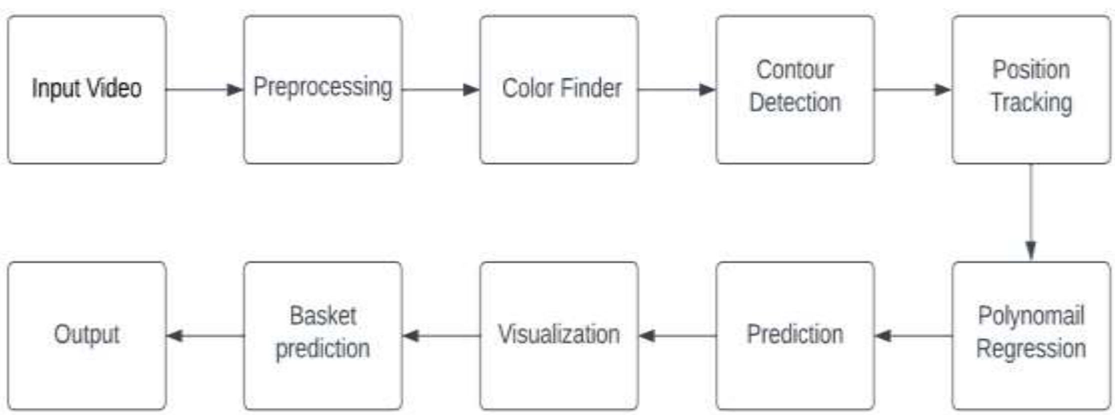

fig -2: Block diagram

3.1 Algorithm

1. Input Video - The program takes a video file as input for processing.
2. ColorFinder: The ColorFinder module is used to find the color range of the ball that needs to be tracked. It takes the input image and the desired HSV range as input, and returns the image and mask of the object with the desired color range

3. Contour Detection: The findContours method from the cvzone library is used to detect contours in the masked image. The minimum area parameter is set to 500 to filter out small contours

4. Position Tracking: The program tracks the position of the ball by appending the center coordinates of the largest contour to two separate lists: posListX and posListY.
5. Polynomial Regression - The program performs a polynomial regression on the list of ball center coordinates to obtain a second-degree polynomial that models the ball's trajectory. It uses the NumPy library's polyfit () function to perform the regression.

6. Prediction: If the number of position data points is less than 10, the program predicts the next position of the ball using the quadratic equation. If the predicted position falls within a certain range (between 330 and 430 pixels on the x-axis), the program predicts that the ball will go into the basket.

7. Visualization - The program visualizes the ball's trajectory by drawing circles and lines on the input video frames. It also draws a predicted trajectory on a separate image, which is shown alongside the input video.

8. Basket Prediction - The program predicts whether the ball will go into the basket by checking whether the predicted trajectory intersects with a predefined region of interest. It uses a boolean variable to store the prediction and displays the result on the output image.

4. CONCLUSIONS

In conclusion, detecting, tracking, and predicting the path of an object is a critical task in computer vision and machine learning that has numerous applications across various fields. The ability to detect and track objects in real-time provides a higher level of situational awareness, reduces human intervention, and improves efficiency. Object detection and tracking are essential in applications like surveillance, security, autonomous vehicles, robotics, sports analysis, medical imaging, augmented reality, traffic management, and wildlife monitoring. With the rapid advancements in computer vision and machine learning, the detection, tracking, and prediction of the path of an object will continue to evolve and become more powerful, enabling new and innovative applications in the future.

5. REFERENCES

[1] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[2] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: A review,” Neurocomput., vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple object tracking using k-shortest paths optimization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 9, pp. 1806–1819, Sept. 2011.

[4] A. R. Zamir, A. Dehghan, and M. Shah, “GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs,” in Proc. 12th ECCV, Florence, Italy, 2012

[3] S. Agren, Object Tracking Methods and Their Areas of Application: A Meta-Analysis, MSc Thesis, Umea Universitet, Sweden, 2017.

[4] M. A. Pinheiro, Multi Object Tracking in Video Sequences, MSc Thesis, University of Porto, Portugal, 2010.

[5] G. Basso, T.G. Amorim, A.V. Brito and T.P. Nascimento, "Kalman filter with dynamical setting of optimal process noise," IEEE Access, vol. 5, pp. 8385-8393, 2017.

[6] G. Du and P. Zhang, "A Markerless human robot Interface using particle filter and Kalman filter for dual robots," IEEE Transactions on Industrial Electronics, vol. 62, pp. 2257-2264m 2015.

[7] L. Martino, J. Read, V. Elvira and F. Louzada, "Cooperative parallel particle filters for on-line model selection and applications to urban mobility," Digital Signal Processing, vol. 60, pp. 172-185, 2017.

[8] K. Saho and M. Masugi, "Performance analysis of alpha-beta-gamma tracking filters using position and velocity measurements," EURASIP Journal on Advances in Signal Processing, vol. 35, 2015.

[9] S.S. Blackman and R. Popoli, Design and Analysis of Modern Tracking Systems, Boston: Artech House, USA, 1999.

[10] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: the KITTI dataset," International Journal of Robotics Research, vol. 32, pp. 1231-1237, 2013.

[11] G. Welch and G. Bishop, An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, Chapel Hill, USA, 2006.

[12] M. Laaraiedh, Implementation of Kalman Filter with Python Language, University of Rennes: IETR Labs, Rennes, France, 2012.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82, pp. 34-45, 1960.

[14] P. A. Jones, Techniques in Kalman Filtering for Autonomous Vehicle Navigation, MSc Thesis, Virginia Polytechnic Institute and State University, Blacksburg Virginia, USA, 2015.

[15] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," In Proc. IEEE International Conference on Information Fusion, Cologne, Germany, 2008, pp.