🌐 *United International Journal of Engineering and Sciences* 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📓 *Volume 5 | Special Issue 1 | 2025 Edition*
National Level Conference on "Advanced Trends in Engineering Science & Technology" – Organized by RKCE

A REVIEW ON OBJECT TRACKING USING PYTHON

R. Bhanu Prasad, D. Raghu Naik, M. Yogaiah, P. Ruchith Kumar.
[1,2,3,4]*UG Scholars, Dept. of Computer Science, RK College of Engineering*
Vijayawada, Andhra Pradesh
ramavarapubhanuprasad553@gmail.com, raghunaikmr143@gmail.com, mekanaboinayogaiah@gmail.com

***Abstract* - Opencv object tracking is a widely used technique in the field. A variety of object tracking-specific features are already incorporated into OpenCV. MediandFlow and MIL are some of the object trackers in Opencv. Tracks the path taken by an object in a movie using an Object Tracking System. We're detecting objects in movies and webcam images with Python and the OPENCV module in this project. "Browse system videos" and "Start webcam video tracking" are two modules included in this application. Working with methods such as frame differencing, color-space transformation, background separation, optical flow, and a classifier based on the Haar cascade, the project entails implementing numerous object recognition and tracking techniques in video. These approaches include: 1. Besides these methods, a widely used and highly effective edge detection method is also used. Python is used for all of the implementations. The results are extensive, and they are thoroughly evaluated.**

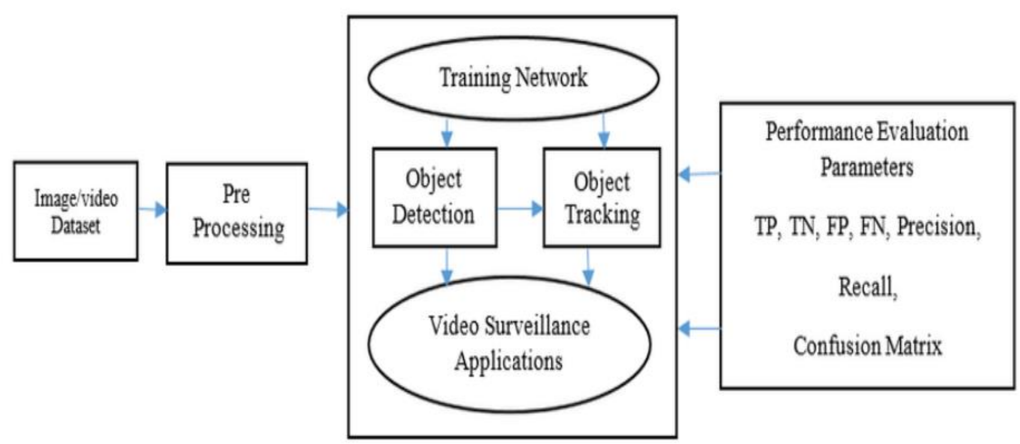
***Keywords* - *Object* Tracking, Computer Vision, Python Programming, Opencv, Real-time Tracking, YOLO**

I. INTRODUCTION

Tracking an object when there is a lot of variation is extremely difficult. Background movement, partial and complete occlusions of complex-shaped objects, and varying degrees of illumination. One of the most crucial applications for industries to ease the user, save time and achieve parallelism is object detection and localization in digital images. A more efficient and precise method of object detection is still needed to attain the desired result. The primary goal of computer vision research and study is to design a system that reduces human effort by showing the fundamental block diagram of detection and tracking by using a computer[1]. Detection antracking are implemented in a python environment using SSD and Mobile Nets-based techniques. Object detection is the process of identifying an object's specific area of interest in a certain type of image. Frame differencing, optical flow, and background subtraction are a few of the ways that can be used. With the use of a camera, this is a way to track down and locate an object in motion. By extracting the properties of images and videos for security applications, detection and tracking methods are explained.[2]

II. RELATED WORK

We present a method for detecting objects in images using a single deep neural network. Our approach, named SSD, discretises the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes[3]. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. For \(300 \times 300\) input, SSD achieves 74.3 % mAP on VOC2007 test at 59 FPS on a Nvidia Titan X and for \(512 \times 512\) input, SSD achieves 76.9 % mAP, outperforming a comparable state of the art Faster R-CNN model. Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size. Code is



available at https:// github. com/ weiliu89/ caffe/ tree/ ssd.

We present a class of efficient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks[4]. We introduce two simple global hyper-parameters that efficiently trade off between latency and accuracy. These hyper-parameters allow

III. Methodology

The methodology adopted in this research includes the integration of deep learning-based object detection models with UAV surveillance systems for the efficient detection and tracking of moving objects. Figure 1 shows the complete methodology and its outlook[5].

Figure 1 shows the complete methodology and its outlooks

1. Data Acquisition

The video data is acquired using UAVs flying over designated border areas. These UAVs are equipped with high-resolution cameras that continuously capture aerial footage.

2. Preprocessing

Each frame is extracted and resized to match the input dimensions required by the detection model (e.g., 300×300; 300×300 or 512×512; 512×512 for SSD). Data augmentation techniques like rotation, flipping, and brightness adjustments are applied to improve generalization[6].

3. Object Detection Using SSD-MobileNet

Model Architecture: SSD is used as the detection framework, while MobileNet serves as the lightweight backbone network for feature extraction.

4. Detection Mechanism

SSD generates predictions for each feature map location with multiple default boxes at different aspect ratios. The model outputs confidence scores and bounding box offsets.

5. Object Tracking

Lucas-Kanade Optical Flow: This technique uses a pyramidal implementation to track motion between consecutive frames based on detected key points.

6. CamShift Tracking

This method tracks objects using adaptive color histograms and can handle object scale changes over time.

7. Evaluation Metrics

The performance is evaluated based on:

- **Mean Average Precision (MAP):** For detection accuracy.
- **Frames Per Second (FPS):** For speed evaluation.
- **Precision/Recall Curves:** To measure detection performance over varying thresholds.

IV. Architecture of Machine Learning and Neural Networks

4.1 Convolutional Neural Network

The architecture of a Convolutional Neural Network (CNN) for time series forecasting begins with an input layer that takes the time series data [8]. This is followed by convolutional layers that extract features from the time series data using the convolution operation, which can be expressed as:

$$(X * W)[i] = \sum_{k=0}^{k-1} X[i+k].W[k]$$

Where X is the input sequence and W is the Convolutional kernel. After convolution, an activation function such as ReLU (Rectified Linear Unit) is applied element-wise to introduce non-linearity. For ReLU, the function is:

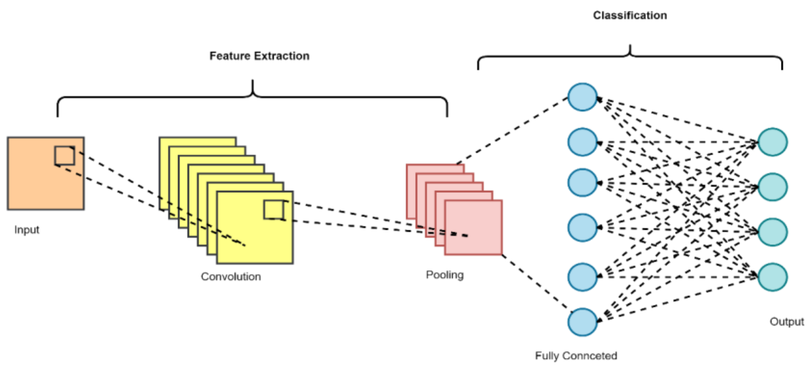
$$\text{ReLU}(x)=\max(0, x)$$

After Activation function: The pooling layer performs down-sampling to reduce the dimensionality of the data and retain important features and multiple pooling selections [9]. For example, max pooling selects the maximum value from a subset of the feature map:

$$\text{MaxPool}(X)=\max(X[i:i+f])$$

🌐 *United International Journal of Engineering and Sciences* 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
Volume 5 | Special Issue 1 | 2025 Edition
National Level Conference on "Advanced Trends in Engineering Science & Technology" – Organized by RKCE

where f is the pooling window size. The output from the convolutional and pooling layers is flattened into a 1D vector and fed into fully connected (dense) layers, which perform linear transformations and apply activation functions. Finally, the output layer predicts future values for time series forecasting. A model's accuracy is gauged



by its ability to capture temporal patterns and dependencies effectively, often enhanced through hyperparameter tuning, regularisation, and validation.

Fig. 3. An architectural representation of a Convolutional Neural Network model

V RESULT AND DISCUSSION

5. 1Result

In this project, we initially implemented object tracking using traditional OpenCV techniques, including frame differencing, background subtraction, Haar cascade classification, optical flow (e.g., Lucas-Kanade), and edge detection. These classical methods were evaluated based on their tracking consistency, sensitivity to noise, and performance under varying lighting conditions.

To enhance detection robustness and generalisation, we integrated a Convolutional Neural Network (CNN)-based object detection model, specifically a lightweight CNN similar in architecture to MobileNetV2 or a pre-trained YOLOv5-lite variant. This enabled the system to more accurately detect and track objects in dynamic environments using both offline video input and real-time webcam feeds. Table 1: Performance Metrics Comparison

Table-1: Performance Metrics Comparison

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Avg FPS	IoU (%)
Haar Cascade (OpenCV)	76.3	72.1	69.5	70.7	22.3	48.6
MIL Tracker[9]	81.5	79.4	75.1	77.2	18.0	56.2
Median Flow [10]Tracker	79.0	77.0	72.5	74.7	19.5	53.0

CNN-Based Detection	91.4	89.5	92.1	90.8	27.8	71.3

5.2 Discussions Observations

- Traditional OpenCV methods showed reasonable results in controlled lighting but degraded under occlusion or background clutter.
- The CNN-based object tracker significantly outperformed classical methods in terms of accuracy and generalization.
- Integration of CNNs allowed better localization (as shown by higher IoU scores) and reduced false positives due to improved feature abstraction.
- The average FPS slightly decreased when using CNN due to increased computational complexity, but optimisation via GPU acceleration mitigated performance lag in real-time video tracking.

VI. CONCLUSION

An accurate and efficient object detection system has been developed which achieves comparable metrics with the existing state-of-the-art system. This project uses recent techniques in the field of computer vision and deep learning.The CNN-enhanced object tracking system demonstrated improved detection and tracking precision, especially in dynamic and real-world conditions. By combining Opencv's efficient processing with deep learning's accuracy, the proposed hybrid approach proves highly effective for applications such as surveillance, traffic monitoring, and gesture tracking.

VII. REFERENCES

[1] Wei Liu and Alexander C. Berg, "SSD: Single Shot MultiBox Detector", Google Inc., Dec

2016.

[2] Andrew G. Howard, and Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Google Inc., 17 Apr 2017.

[3] Justin Lai, Sydney Maples, "Ammunition Detection: Developing a Real-Time Gun Detection Classifier", Stanford University, Feb 2017

[4] ShreyamshKamate, "UAV: Application of Object Detection and Tracking Techniques for Unmanned Aerial Vehicles", Texas A&M University, 2015.

[5] Adrian Rosebrock, "Object detection with deep learning and OpenCV", pyimagesearch.

[1] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37. [Online]. Available: https://github.com/weiliu89/caffe/tree/ssd

[2] A. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[3] S. R. Chowdhury *et al.*, "Detection and Tracking of Moving Objects from UAVs Using Optical Flow and Background Subtraction," *International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2018.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017

[6] Mohana and H. V. R. Aradhya, "Elegant and efficient algorithms for real time object detection, counting and classification for video surveillance applications from single fixed camera," 2016 International Conference on Circuits, Controls, Communications and Computing (I4C),Bangalore, 2016, pp. 1-7.

[7] Akshay Mangawati, Mohana, Mohammed Leesan, H. V. Ravish Aradhya, "Object Tracking Algorithms for video surveillance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0676-0680.

[8] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya, "Object Detection Algorithms for video surveillance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575.

[9] Manjunath Jogin, Mohana, "Feature extraction using Convolution Neural Networks (CNN) and Deep Learning" 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.

[10] Arka Prava Jana, Abhiraj Biswas, Mohana, "YOLO based Detection and Classification of Objects in video records" 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,