iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72405

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

MOB-NET-SSD: An Enhanced Real Time Object Identification Approach Based on Deep Learning

Shounak Bandyopadhyay[1], Sohini Banerjee[2], Avishek Gupta[3], Shayan Ghosh[4], Souvik Paul[5], Subhadip Das[6]
[1]*Assistant Professor, Department of Electronics and Communication Engineering, AIEM, Mogra*
[2, 3]*Assistant Professor, Department of Computer Science and Engineering, AIEM, Mogra*
[4, 5]*B.Tech Student, Department of Computer Science and Engineering, AIEM, Mogra*
[6]*Assistant Professor, Department of Mechanical Engineering, AIEM, Mogra*

Abstract: ***A large, active, and complex field of computer vision dedicated to object identification and recognition is called real-time object detection. Using OpenCV (Open-source Computer Vision), a set of programming methods primarily trained towards real-time computer vision in digital photos and videos, object detection finds the semantic objects in a class. People with visual impairments are unable to recognize objects in their environment. Helping the blind overcome their challenges is the primary goal of this real-time object detection. Applications for real-time object detection include object tracking, video surveillance, people counting, pedestrian identification, self-driving automobiles, face detection, ball tracking in sports, and many more. Convolution Neural Networks, a type of deep learning technique, are used to do this. This article serves as a helpful resource for those who are visually impaired.***

I. INTRODUCTION

Object detection is a technology that allows for the detection of different things in digital photos and films. The primary applications for it are in face detection, self-driving automobiles, and other situations requiring constant object monitoring. In this project, the object detection mechanism or method used is Convolutional Neural Networks, a subclass of Deep Learning. This makes use of the Mobile-Net SSD technique, in which the multibox detector is realized through the usage of SSD, a framework, and Mobile-Net, a neural network used for picture classification and recognition. Object detection is possible when Mobile-Net and SSD are combined. The primary benefit of utilizing deep learning over machine learning is the elimination of the requirement to extract features from data.

An important factor to recognize the items in an image is the Haar-like trait. Beginning at the upper left corner of the image, they scan the entire thing, comparing each little box with the training data. This allows for the identification of even the smallest, finely detailed things in the pictures.

II. LITERATURE REVIEW

Joseph Redmon developed the YOLO method for real-time object detection and prediction with YOLO classifiers [1]. Juan Du developed optimal strategies based on the CNN model and YOLOv3 algorithm, which explain efficiency and increase dependability [2]. According to Matthew B. Blaschko, other researchers concentrated on learning and localizing objects using an organized method to provide output regression. In order to get around the drawbacks of sliding window techniques, the researchers focused on object localization and bounding box approaches in this article [3]. In computer vision, real-time object identification is a difficult problem that involves identifying and locating items of interest in live camera feeds or real-time video streams. Two well-known classifiers, Mobile-Net SSD (Single Shot Detector) and YOLOv3 (You Only Look Once), have drawn a lot of attention lately and demonstrated outstanding performance in real-time object identification [4]. YOLOv3 is a cutting-edge object identification algorithm that analyses algorithm speed and aims for high accuracy. By splitting the input picture into a grid and using each grid cell to estimate bounding boxes and class probabilities, it uses a single-shot detection technique. To capture objects at various sizes and achieve high detection accuracy, YOLOv3 employs a deep neural network architecture with numerous detection layers. It can effectively handle complicated scenarios and detect a broad variety of object classifications [5]. On the other hand, Mobile-Net SSD achieves real-time object recognition on devices with limited resources by fusing the SSD framework with the Mobile-Net architecture. Designed for mobile and embedded devices, Mobile-Net is a lightweight neural network architecture that uses depth-wise separable convolutions to minimize computational complexity without sacrificing accuracy. Objects of different sizes and aspect ratios may be detected because to the SSD framework's multi-scale feature extraction and detection capabilities [6].

Both Mobile-Net SSD and YOLOv3 have benefits in terms of accuracy, adaptability, and real-time speed. Numerous applications, such as robots, augmented reality, autonomous cars, and surveillance systems, have made extensive use of them [7]. Real-time object detection with YOLOv3 or Mobile-Net SSD in surveillance systems allows for the monitoring and identification of things of interest in live video feeds, including people, cars, or suspicious objects. This improves security and makes it possible to react quickly to possible attacks [8]. Real-time object detection is essential for autonomous cars to recognize and follow cars, people, and obstructions in their environment. Real-time object recognition capabilities from YOLOv3 and Mobile-Net SSD can help autonomous cars make wise judgments and guarantee safe navigation [9]. Real-time object detection in robotics enables robots to sense and engage with their surroundings. For jobs involving object manipulation, scene comprehension, or human-robot collaboration, YOLOv3 and Mobile-Net SSD can be utilized for object detection [10]. Real-time object detection with YOLOv3 or Mobile Net SSD allows for the tracking and identification of objects in real-world settings in augmented reality applications. This makes it possible to put and interact with virtual items in real time, creating immersive and engaging experiences [11]. Being a more current version, YOLOv4 Tiny performs better and has a number of improvements [12]. In general, YOLOv4 Tiny outperforms YOLOv3 Mobile-Net SSD in terms of accuracy while retaining its tiny model size and real-time inference capabilities. Applications with low computing resources, such embedded systems, drones, edge devices, autos, and pedestrian detection utilizing YOLOv3 and YOLOv4 self-driving cars, are especially well-suited for it [13].These articles provide valuable insights into the development and optimization of real-time object detection systems, particularly focusing on Mobile-Net-SSD and its applications in resource-constrained environments.

III. METHODOLOGY

Deep learning, an area of machine learning, which is itself a subfield of artificial intelligence (AI), uses networks that can extract knowledge from unlabeled or unstructured data. In this project, convolutional neural networks (CNN) are the method used. It makes use of Haar-cascade classifiers, which aid in object detection.

1.CNN : Although this network will be employed with one-dimensional and three-dimensional data, the convolutional neural network, or CNN for short, may also be a specific type of neural network model made for working with two-dimensional envision data.

The convolutional layer, which gives the network its name, is at the heart of a convolutional neural network. This layer carries out a process called "convolution." Similar to a conventional neural network, a convolution in a convolutional neural network may be a linear operation involving the multiplication of a set of weights with the input. An array of input files and a two-dimensional array of weights, known as a filter or kernel, are multiplied, provided that the technique was intended for two-dimensional input.

Since the filter is shorter than the input file, a scalar product may result from the sort of multiplication that was done before between an input patch that was the size of the filter and the filter. The element-wise multiplication of the filter-sized input patch by the filter, followed by its summation, yields a single value in the case of a scalar product. The operation is typically represented and referenced as the "scalar product" because it results in a value of 1.

It is deliberate to employ a filter smaller than the input because doing so enables the input array to multiply an equivalent filter (set of weights) several times at different input points. In particular, the filter is applied methodically from top to bottom, left to right, to every overlapping region or filter-sized patch in the input file.

This methodical use of a comparable filter throughout an image might be a really effective concept. If the purpose of the filter is to identify a specific type of feature in the input, then applying the filter consistently over the entire input image gives the filter an opportunity to capture that feature wherever in the image.

This capacity is commonly described as translation invariance, i.e., the complete worry about whether the characteristic is present rather than where it ought to be.Here we have proposed mobile-Net as image classifier.

A. Mobile Net

MobileNet is a convolutional neural network architecture designed for efficient mobile and embedded vision applications. It was developed by researchers at Google, with the primary aim of enabling deep learning models to run efficiently on mobile and resource-constrained devices.MobileNet architecture has some key features which is discussed below.

1) Depthwise Separable Convolution: MobileNet extensively uses depth wise separable convolutions instead of traditional convolutions. This type of convolution separates the spatial convolution (depth wise convolution) from the pointwise convolution (1x1 convolution). This reduces computational complexity and the number of parameters, making it more lightweight and efficient.

2) Width Multiplier and Resolution Multiplier: MobileNet introduces two hyperparameters, namely width multiplier and resolution multiplier, which allow trading off between model size, latency, and accuracy. The width multiplier reduces the number of channels in each layer, while the resolution multiplier reduces the input image resolution.

3) Architecture: MobileNet consists of multiple layers of depth wise separable convolutions followed by pointwise convolutions and optionally followed by batch normalization and ReLU activation. The final layers typically include global average pooling and a fully connected layer for classification.

4) Versions: There are several versions of MobileNet, such as MobileNetV1, MobileNetV2, and MobileNetV3, each introducing improvements in efficiency and performance. In our research we are using the V2 version of mobile net architecture.

Overall, MobileNet architectures are widely used in various computer vision tasks on mobile devices and embedded systems where computational resources are limited. They strike a balance between model size, computational efficiency, and accuracy, making them suitable for real-time applications on devices with restricted resources.

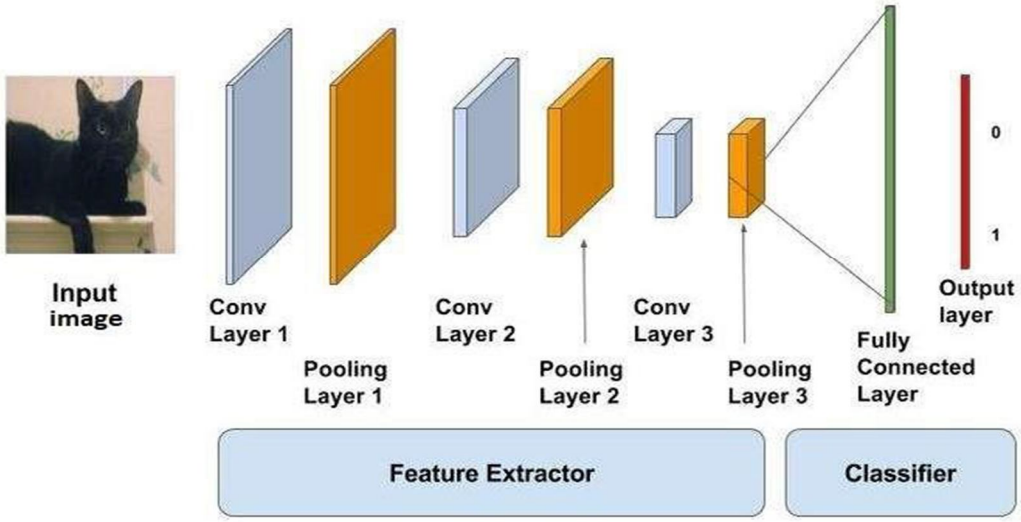


Fig 1 : Architecture of CNN based Mobile-Net model

B. SSD (Single Shot MultiBox Detector)

SSD is a popular framework for object detection tasks in computer vision. It's known for its efficiency and accuracy in detecting objects within images.

1) Base Convolutional Network: SSD typically utilizes a base convolutional network, such as VGG, Inception, or ResNet, to extract features from input images. These networks are often pre-trained on large-scale image datasets like ImageNet to capture general features.

2) Multi-scale Feature Maps: SSD incorporates feature maps at multiple scales to capture objects of different sizes. It adds convolutional layers of different sizes on top of the base network to generate feature maps at various resolutions.

3) MultiBox Priors: SSD predicts bounding boxes and class scores at each spatial location on the feature maps. Instead of predicting bounding boxes directly, SSD utilizes a set of default bounding boxes called "priors" or "anchor boxes." These priors are predefined boxes with different aspect ratios and scales, covering a wide range of possible object shapes and sizes.

4) Predictions: SSD predicts two types of information for each default box: class scores (indicating the presence of each object class) and offsets to adjust the default box to better fit the object's location. This prediction is done using additional convolutional layers applied to the feature maps.

5) Loss Function: SSD employs a combination of localization loss (such as Smooth L1 loss) and confidence loss (typically cross-entropy loss) to train the network. The localization loss penalizes the deviation between predicted bounding boxes and ground truth boxes, while the confidence loss measures the accuracy of class predictions.

6) Post-processing: After the network predicts bounding boxes and class scores, a post-processing step is applied to filter out redundant and low-confidence detections. Techniques like non-maximum suppression (NMS) are commonly used for this purpose.

SSD offers a good balance between accuracy and speed, making it suitable for real-time object detection applications, including those on mobile and embedded devices. Its efficient use of convolutional neural networks and multi-scale feature maps enables it to detect objects accurately across various sizes and aspect ratios within an image. Fig. 2 demonstrates the work flow structure of our proposed model.

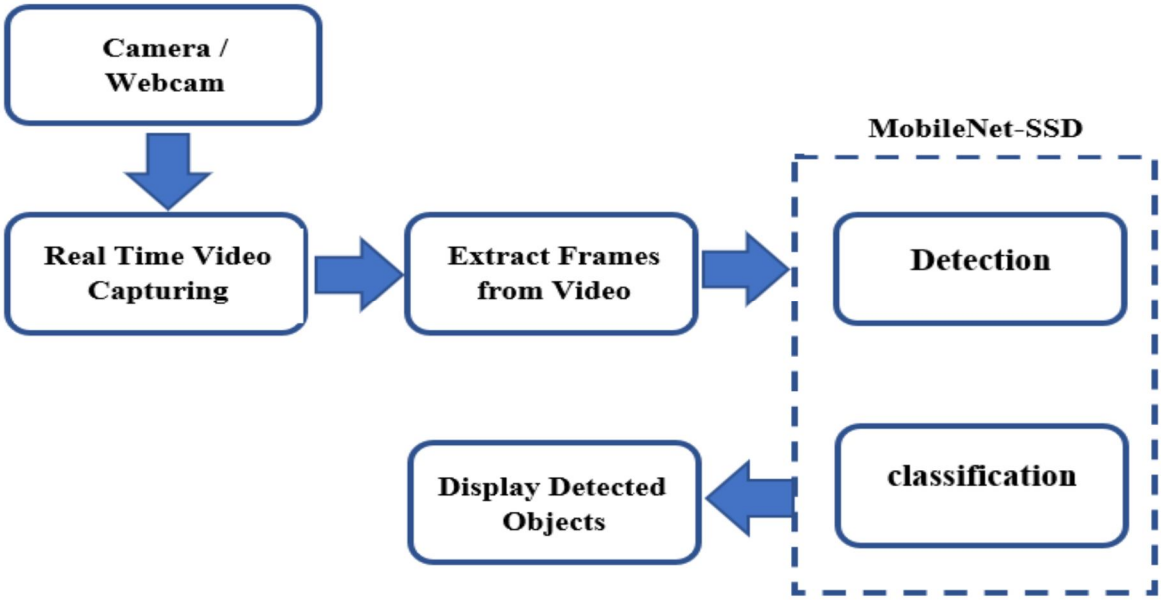


Fig. 2: Work flow structure of our proposed model.

IV. DATA SET

The collection of photos representing every object that has to be recognized makes up the info set that we employed for the thing detection. Generally, the data collection contains many photographs of each and every object. The accuracy is frequently increased if the datasets contain more photos of each object. The most crucial thing to keep in mind is that the data set's information has to be tagged. In reality, three data sets will exist. They are the validation dataset, the training dataset, and the testing dataset, respectively. Most of the labeled data, generally between 85 and 90 percent, is included in the training data set. Since our machine will be trained using this training dataset, the info set must be trained in order to acquire the model. About 5–10% of the total labeled data is included in the validation data set. This is frequently employed for purposes of validation. The testing dataset, on the other hand, is used to evaluate our machine's performance.

The authors has choose MS COCO (Microsoft Common Objects in Context) as their dataset to train, test & validate their proposed model. Large-scale object recognition, segmentation, key-point detection, and captioning are all included in the MS COCO (Microsoft Common Objects in Context) dataset. There are 328K images in the dataset.

In 2014, the MS COCO dataset's first version was made available. It has 164K photos divided into sets for testing (41K), validation (41K), and training (83K). A fresh test set of 81K photographs, comprising 40K new images plus all of the test images from prior releases, was made available in 2015.

In 2017, the training/validation split was restructured from 83K/41K to 118K/5K based on input from the community. The annotations and images are the same in the new split. A subset of the 41K photos from the 2015 test set make up the 2017 test set. The 2017 edition also includes a fresh 123K image unannotated dataset.

V. RESULTS & CALCULATION

In this research article authors are measuring loss & accuracy evaluation metrices to demonstrate the model performance during training & validation.

Loss: Loss measures how well the model is performing on the training data. It quantifies the difference between the predicted values and the actual target values.

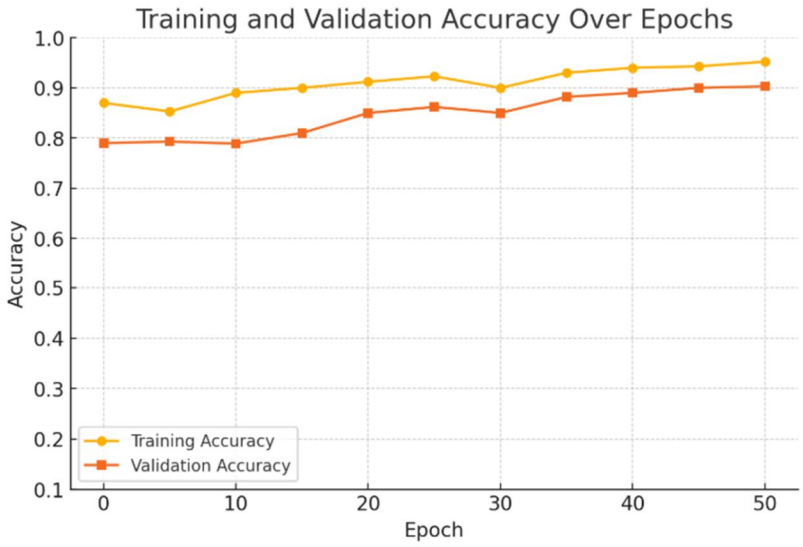
Accuracy: Accuracy is a measure of how many predictions the model got right compared to the total number of predictions made. It's usually expressed as a percentage. For classification tasks, accuracy is the number of correct predictions divided by the total number of predictions. The formula for accuracy is given below.

Accuracy = (TP + TN) / (TP + FP + TN + FN). where Here TP, FP, FN and TN represent true positives, false positives, false negatives and true negatives. We have trained the model up to 50 epochs & got up to 90.3% of validation accuracy which shows a good result in terms of deep learning based predictive model.

Table 1: MOB-NET-SSD model performance training & validation outcomes

Type	Loss	Accuracy
MOB-NET-SSD _TRAIN	0.09	0.952
MOB-NET-SSD _VAL	0.132	0.903
DIFFERENCE	-0.042	0.049

Graph 1: Training & validation loss over epochs



Graph 2: Training & validation accuracy over epochs

A. Analysis

The graph 1 represents Training and Validation Loss Over Epochs in our proposed model. Both training and validation loss decrease as epochs increase, indicating that the model is learning effectively. The training loss (yellow line) decreases smoothly.The validation loss (orange-red line) also decreases but exhibits minor fluctuations, which is common. Around epoch 40, the validation loss slightly increases before decreasing again. This could indicate slight overfitting, but the gap between training and validation loss is small, suggesting that overfitting is not severe. In general,there is no major overfitting, though minor fluctuations in validation loss suggest monitoring for potential early stopping. The model appears to be well-optimized and learning effectively**.**

Graph 2 exihibits Training and Validation Accuracy Over Epochs in the model. Both training accuracy (yellow line) and validation accuracy (red-orange line) increase as epochs progress. The validation accuracy follows the training accuracy but remains slightly lower throughout. There is a small gap between training and validation accuracy, but it is not significant, suggesting minimal overfitting. The validation accuracy remains stable and does not drop, which indicates good generalization. The model reaches a high accuracy (~90% for validation), which is a strong indication of good performance. So, from these graphs the authors can conclude that the model is likely well-trained and generalizing effectively.

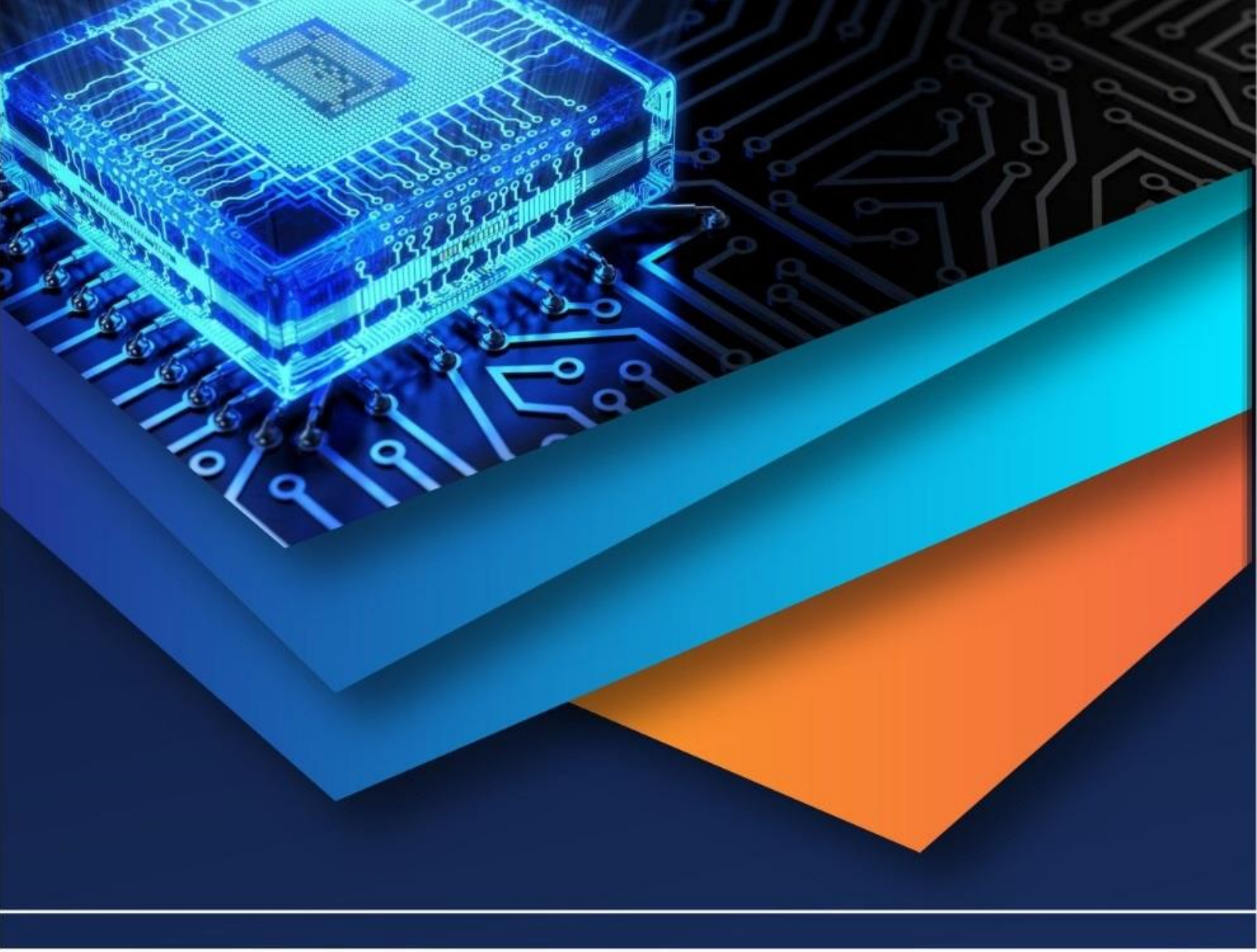
VI. CONCLUSION

In this study, we have presented MOB-NET-SSD, an enhanced real-time object identification approach that builds upon the strengths of Single Shot MultiBox Detector (SSD) and MobileNet, a lightweight deep neural network architecture. Our approach aims to provide an efficient and accurate solution for object detection in resource-constrained environments, such as mobile and embedded systems. Future work will focus on further refining the model to enhance its robustness and extend its applicability. Potential directions include incorporating advanced techniques such as attention mechanisms to improve object detection in cluttered and dynamic environments, as well as exploring the integration of MOB-NET-SSD with other sensory data to create a more holistic perception system.

In conclusion, MOB-NET-SSD represents a significant advancement in real-time object detection technology, offering a balanced trade-off between speed and accuracy. Its ability to perform efficiently on low-power devices opens up new possibilities for deploying intelligent vision systems across various domains, paving the way for smarter and more responsive applications in the future.

REFERENCES

[1] Mao, Qi-Chao, Hong-Mei Sun, Yan-Bo Liu, and Rui-Sheng Jia. "Mini-YOLOv3: real-time object detector for embedded applications." Ieee Access 7 (2019): 133529-133538.
[2] Masurekar, Omkar, Omkar Jadhav, Prateek Kulkarni, and Shubham Patil. "Real time object detection using YOLOv3." International Research Journal of Engineering and Technology (IRJET) 7, no. 03 (2020):3764-3768.
[3] Gai, Wendong, Yakun Liu, Jing Zhang, and Gang Jing. "An improved Tiny YOLOv3 for real-time object detection." Systems Science & Control Engineering 9, no. 1 (2021): 314-321.
[4] Zhang, Xiuling, Xiaopeng Dong, Qijun Wei, and Kaixuan Zhou. "Real-time object detection algorithm based on improved YOLOv3." Journal of electronic imaging 28, no. 5 (2019): 053022-053022.
[5] Srithar, S., M. Priyadharsini, F. Margret Sharmila, and R. Rajan. "Yolov3 Supervised machine learning framework for real-time object detection and localization." In Journal of Physics: Conference Series, vol. 1916, no. 1, p. 012032.IOP Publishing, 2021.
[6] Gunawan, Chichi Rizka, Nurdin Nurdin, and Fajriana Fajriana. "Design of A Real-Time Object Detection Prototype System with YOLOv3 (You Only Look Once)." International Journal of Engineering, Science and Information Technology 2, no. 3 (2022): 96-99.
[7] Gong, Hua, Hui Li, Ke Xu, and Yong Zhang. "Object detection based on improved YOLOv3-tiny." In 2019 Chinese automation congress (CAC), pp. 3240-3245. IEEE, 2019.
[8] Pang, Lei, Hui Liu, Yang Chen, and Jungang Miao. "Real-time concealed object detection from passive millimeter wave images based on the YOLOv3 algorithm." Sensors 20, no. 6 (2020): 1678.
[9] Tan, Lu, Tianran Huangfu, Liyao Wu, and Wenying Chen. "Comparison of RetinaNet, SSD, and YOLO v3 for realtime pill identification." BMC medical informatics and decision making 21 (2021): 1-11.
[10] Chen, Zhihao, Redouane Khemmar, Benoit Decoux, Amphani Atahouet, and Jean-Yves Ertaud. "Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility."In 2019 Eighth International Conference on Emerging Security Technologies (EST), pp. 1-6. IEEE, 2019.
[11] Shill, Apu, and Md Asifur Rahman. "Plant disease detection based on YOLOv3 and YOLOv4." In 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), pp. 1-6. IEEE, 2021.
[12] Nepal, Upesh, and Hossein Eslamiat. "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs." Sensors 22, no. 2 (2022): 464.
[13] Khan, Asjad M. "Vehicle and pedestrian detection using YOLOv3 and YOLOv4 for self-driving cars." PhD diss.,California State University San Marcos, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY