# Object Detection using Deep Learning Approach

Paromita Saha[1], Moloy Dhar[2]
[1,2]Guru Nanak Institute of Technology, Kolkata
[1,2]Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology
(Formally known as West Bengal University of Technology)
Kolkata, West Bengal, India.
[1]paromita2398@gmail.com, [2]moloy.dhar@gnit.ac.in

***Abstract:*** **The most often utilized strategies for current deep learning models to accomplish a multitude of activities on devices are mobile networks and binary neural networks. In this research, we propose a method for identifying an object using the pre-trained deep learning model MobileNet for Single Shot Multi-Box Detector (SSD). This technique is utilized for real-time detection as well as webcams to detect the object in a video feed. To construct the module, we use the MobileNet and SSD frameworks to provide a faster and effective deep learning-based object detection approach. Deep learning has evolved into a powerful machine learning technology that incorporates multiple layers of features or representations of data to get cutting-edge results. Deep learning has demonstrated outstanding performance in a variety of fields, including picture classification, segmentation, and object detection. Deep learning approaches have recently made significant progress in fine-grained picture categorization, which tries to differentiate subordinate-level categories. The major goal of our study is to investigate the accuracy of an object identification method called SSD, as well as the significance of a pre-trained deep learning model called MobileNet. To perform this challenge of detecting an item in an image or video, I used OpenCV libraries, Python, and NumPy. This enhances the accuracy of behavior recognition at a processing speed required for real-time detection and daily monitoring requirements indoors and outdoors.**

***Keywords- Object Recognition, Deep Learning, CNN, Single Shot Multi-Box Detector, Open Source Computer Vision, MobileNet V3. (key words)***

## I. INTRODUCTION

Object detection is now employed in a variety of fields around the world, including surveillance cameras, pedestrian displays, self-driving cars, and facial recognition. The sub-discipline of Object Detection in the Deep Learning field comprises an image such as a picture, video, or webcams [1]. Object detection from videos is an important task in video surveillance applications these days. Object identification is a technique for identifying necessary items in video streams and clustering their pixels.

Object detection is the process of extracting and classifying real-world object instances from photos or videos, such as a car, bike, TV, flowers, and persons. Because it allows for the recognition, localization, and detection of many objects within an image, an object detection approach allows you to analyse the features of an image or video [2].

Object detection is the process of locating and classifying items using rectangular bounding boxes to identify them and sort them into categories. Object detection and object categorization, as well as semantic segmentation and instance segmentation, have certain connections. Face detection, text detection, pedestrian detection, logo detection, video detection, vehicle detection, and medical image detection are all examples of object detection, which has vital uses in both scientific study and effective industrial production [3].

Object detection has come a long way since R-CNN predicted it in 2014, based on DNN. Eventually, object identification algorithms include SPP-NET, Fast-RCN, Faster RCNN, and R-FCN, which are all based on R-CNN. The efficiency of these methods was high; although, a network structure is made up of numerous elements in a complicated interaction [1]. The complex DNN model for the recognition of things can accomplish high accuracy. However, they require a massive number of computation and setup variables, which aren't really appropriate for all systems.

Conventional methods can be used to handle this challenge, however Convolutional Neural Networks (CNN) appear to be a viable option for systems. Because it is impossible to process live video with the SSD-MobileNet [4] paradigm in an embedded device, we investigated strategies that would allow us to speed up video processing times with minimal loss of accuracy in a system. The following is one of the specific contributions of my work: A evaluation of systems for a counting application in terms of accuracy, performance, and processing times. The object detection method is SSD-MobileNet. The trained data set includes of our own photos for each of the five classes, a well-known data set MS-COCO (91 classes).

As a result, a single network and faster performance are required. As a result, the Single Shot Multi-Box Detector is built on MobileNet and features additional layers such as feature extraction layers that were designed specifically for real-time object recognition, removing the need for a region proposal network and speeding up the process. SSD makes various modifications to the multi-scale features and default boxes idea to compensate for the reduction in precision. There are two parts to the SSD object detection: To detect objects, first extract feature maps and then apply convolution filters [6].

Mobilenet-v3 networks now provide a decent balance of processing speed and object detection accuracy. As a result, we chose the Mobilenet-v3 network, which is supported by many embedded platforms, as the backbone network for the development of the proposed compact object detection model, taking into consideration the restrictions of the device [8].

Section II of this paper discusses related work, Section III details the proposed technique, and Section IV depicts the conclusion.

## II. RELATED WORK

They employed RGB-depth videos, single and multiple viewpoints, and only single networks in this paper [9]. All of the datasets they utilized perform single activities, with one individual executing one task at a time. On the three types of datasets, this work gives a review of several state-of-the-art deep learning-based approaches proposed for human action recognition. However, these methods have a number of disadvantages, including the necessity to create big datasets, the fact that performance is dependent on the magnitude of the network weights, and the fact that hyper-parameter adjustment is difficult. Multiple networks from different streams are also necessary to recognise multiple human actions at the same time.

The methods employed in this project are Convolutional Neural Networks, which are used for identifying head posture, mouth motions, and face detection in this paper [10]. This research proposes a collection of techniques for online evaluation abnormal behavior monitoring based on image data. The monitoring of anomalous behavior such as turning heads and talking during the online test is done through the application of head pose estimation based on convolutional neural networks and threshold-based mouth state assessment, as well as the combination of specific decision rules. However, it detects all noticeable face motions, resulting in false reports and a low identification rate.

The researcher applied convolutional neural networks and deep neural networks, as well as a variety of dataset models, in this publication [11]. Object detection systems such as GoogleNet, AlexNet, ResNet, ResNeXt, SENet, DenseNet, and others are utilized. Animal detection, handed arms detection, human detection, and several other image object detections are also included. There are a few drawbacks to this project. Researchers can reduce the false positive rate by employing more models and evaluating the system, or by pre-processing the pictures and using videos.

Three separate pre-trained datasets were employed by the author. MobileNetv2, GoogleNet, and MobileNetv3 are the three. According to the results of this method present in this study [12], MobileNetV3 [15] is appropriate for handgun detection since it provides a perfect match between prediction speed and precision. The proposed model had a training accuracy rate of 96%. Real-time detection in live videos and photos was used in the tests. The system has an SMS capability that allows it to send an alarm message to the supervisor once a firearm is spotted. The proposed method can be utilised for a variety of purposes, including real-time identification of guns in supermarkets.

This review study [13] provides a thorough and in-depth examination of deep learning-based object detection algorithms. Backbone networks, detection designs, and loss functions are three parts of it. It also provides a thorough examination of the difficult issues. To offer a complete examination of complicated situations, the authors used Deep CNNs, Recurrent CNNs, and Support Vector Machines. More accurate detection frameworks can increase the real-time and accuracy of embedded detection applications, allowing object detection to be used in numerous applications. There is still a scarcity of research on object detection in 3D images and depth images (RGB-D), which necessitates additional attention. The current object detection algorithm is mostly intended for photos of small and medium size. In addition, the accuracy of detecting different scales of objects in high definition photographs is incredibly low.

The object detection algorithm in this paper [14] can recognise objects at up to 14 frames per second, therefore even low-quality cameras with any frame rate can yield good results. They just use a webcam with a frame rate of 6 frames per second. The SSD method demonstrated interior and outdoor input video frames through camera in our testing, but the placement of the objects differed between two consecutive frames. The video acquired by the webcam, as well as the algorithm, convert the size of a single frame to 300 x 300 pixels. The SSD can generate numerous anchor boxes for multiple categories with varying confidence levels by employing a greater proportion of default boxes, which can have a stronger effect, and distinct boxes for each position. The author is merely utilizing a webcam to detect items. Furthermore, the webcam is limited to 14 frames per second.

The author of this paper [16]employed the Pascal VOC [19] dataset in the experiments for network model training and testing in order to evaluate the detection performance of the proposed Mobilenet-SSDv2 detector. They also evaluated the new detector's object detection accuracy to that of the existing MobileNet-SSD detector. The network model's input picture format is a 512x512 RGB color image. They used a 200-epoch SGD method optimizer to train the proposed network. Based on the MobileNet-v2 backbone network, they suggested a lightweight network design with better feature extraction. However, the optimizer they utilized consumes a lot of memory, and the network's computation rate is extremely poor.

The article [17] describes a series of algorithms that use a Convolutional Neural Network to handle characteristics such as video resolution, bit rate, and extra hardware (VPU) for video processing. A comparison is made between an SSD-MobileNet model with self-trained and pre-trained training. The goal is to produce high performance metrics and fast execution time, which will allow a vehicle counting system to be implemented in a low-capacity embedded device. They could include enhancements to the automobile counting, notably in the tracking block, allowing for accurate vehicle tracking even as the number of skip frames grows.

They attempted to recognise an object that was shown in front of a webcam in this study [18]. The generated model was evaluated and trained using Google's TensorFlow Object Detection API framework. They concentrated on threading methodology to enhance fps, which resulted in a significant reduction in processing time. The detecting rate of the item decreases as the distance between the webcam and the object increases due to the 1.3mp web camera's inadequate pixel capacity. Furthermore, detecting a single object takes approximately three seconds, which is a significant constraint.

## III. PROPOSED METHODOLOGY

### 1. Object Detection:

Every object in the bounding box is classified and localised using object detection. Object recognition in computer vision is as straightforward as it sounds: it focuses on detecting, identifying, and locating things. Image categorization is used for object detection. The sliding window approach [23], which is the most basic strategy for detecting objects, is used to reduce time complications in detection techniques. To search over the objective image, a window of suitable size M x N, also known as a bounding box, is chosen in this method. These methods can be divided into two groups: region proposal-based methods and classification-based methods. Single shot detector (SSD) [5] is one of the classification-based methods used in our model.

### 2. Related Technology:

#### 2.1. Deep Neural Network:

Deep learning is a type of machine learning. It allows a computer to learn how to predict and classify information by filtering inputs through layers. Images, text, and audio can all be used to express observations. The way the human brain filters information is the source of inspiration for deep learning. When machines can perform tasks that would normally need human intelligence, artificial intelligence becomes vital. It falls under the machine learning layer, in which machines may adapt to new experience and develop abilities without the need for human intervention. Artificial neural networks, algorithms inspired by the human brain, learn from enormous volumes of data in deep learning, which falls under machine learning. Learning is enabled by the many (deep) layers of neural networks.

#### 2.2. Convolutional Neural Network:

CNNs have proven to be effective in picture recognition and categorization. They are deep learning algorithms that accept video/image input, give weights and bias to various parts of the image, and then discriminate them from one another. They try to make use of the spatial information contained within an image's pixels. CNNs are the most extensively used deep learning algorithms and the most well-known type of neural network, mostly in high-dimensional data such as photos and videos. It's a neural network (NN) architecture with multiple layers. It's a multi-layer neural network (NN) architecture with convolutional layers (or layers) followed by fully connected (FC) layers (s). Between these two levels, subsampling layers can occur.

**2.2.1. Convolutional Layer:** The outcome of linked inputs in the receptive field is determined by this layer, which is the primary building element of a convolutional neural network. Kernels are concatenated over the height and width of the data sets, calculating the matrix multiplication in between input and filter values, to get this result.

**2.2.2. Non-linearity Layer:** When represented, nonlinear functions are quite important and have a degree greater than one. The primary goal of this phase is to convert the input signal into an output signal, which will then be used as an input in the next layer. Non-linearity layers come in a variety of shapes and sizes, including sigmoid, Tanh, ReLU, and others.

**2.2.3. Pooling Layer:** The Pooling layer, much like Convolutional Layer, is responsible for shrinking the Convolved Feature's size of the image. Through dimensionality reduction, the computer power required to process the data is reduced. It's also beneficial for extracting rotatable and spatial consistent high - level features, which helps keep the model's training process flowing efficiently.

**2.2.4. Fully Connected Layer:** FC layers comprise standard deep NN layers that attempt to derive predictions from activation functions for classification and regression problems. Implementing a Fully-Connected layer is a (typically) low-cost approach of training non-linear combinations of high-level information represented by the convolutional layer's outcome.

**2.2.5. Classification\Loss Layer:** This loss layer specifies how the training prevents variation between the true and predicted labels, hence it is mostly utilized to direct the NN training phase. Different loss functions, including as Softmax, cross-entropy, and others, may be used in DCNN to fit particular tasks.

**3. Data Model:** Deep CNN contributed significantly in a number of areas, including picture recognition and classification, and as a result, they have become widely accepted benchmarks. The contemporary structure of Deep CNN, that we utilised in our project, is discussed in this section.

**3.1. MobileNet v3:** MobileNetV3 is the remastered edition of the framework that supports many popular mobile applications' visual analysis capabilities. Popular platforms like TensorFlow Lite have also adopted the approach. The improvements in computer vision and deep learning in general, as well as the limits of mobile contexts, must be carefully balanced by MobileNets. The use of Machine learning algorithms to discover the highest suitable neural network design for a given task is MobileNetV3's key contribution. This contrasts with previous incarnations of the architecture's hand-crafted design. In research paper [6], detailed most current developments to the MobileNets framework. We utilized MobileNetV3 object detection models within classification models, which decreased detection latencies by 25% for the MS-COCO dataset [22] compared to MobileNetV2 at the very same precision. MobileNet employs 3x3 depth-wise separate convolutions, which need up to 8 times less processing than ordinary convolution while achieving

just a minor drop in accuracy. Object identification, fine grain categorization, facial characteristics, and large scale-localization are some of the applications and use cases [27].

**Figure 1. Architecture of MobileNetv3.**

**3.2. Non-Maximum Suppression:** NMS (Non-Maximum Suppression) is a computer vision approach that is utilised in a variety of tasks. It's a group of methods for picking one thing (like bounding boxes) out of several other overlapping ones. The most typical criteria are some kind of frequency number and some kind of overlap metric. We used NMS to reduce a large number of observed bounding boxes just to very few. These windows are said to contain just one object, and each class is assigned a probability/score by a classifier. After the detector generates a huge number of bounding boxes, the best ones must be filtered away.

**3.3. Single Shot Multi-Box Detector:** SSD Object Detection generates a subset of features from a CNN-based deep learning model and then implements convolution filters to recognise things. The base network in our approach is MobileNet. When employing multibox, a one-shot detector like YOLO uses only one shot to detect numerous objects in an image. It has a substantially faster object detecting system with good accuracy. With object detection in SSD [25], detection rate accuracy is accomplished by utilising numerous boxes or filters of various sizes and aspect ratios. These filters are also applied to various extracted features from a network's subsequent phases. This facilitates detection at various scales. On VOC2007, a fast comparison of the speed and accuracy of various object identification models. When compared to two-shot RPN-based techniques, SSD is substantially faster. At 59 frames per second, the SSD300 delivers 74.3 percent mAP, while the SSD500 scores 76.9% mAP at 22 frames per second.

**Figure 2. Architecture of SSD Model.**

**3.4. Caffe Model:** Berkeley AI Research and community collaborators created Caffe [24] [26], a deep learning framework. Caffe was created as a quicker and more convenient object detection framework than existing systems. With a single NVIDIA K-40 GPU, Caffe can compute 60 million photos per day. Inference takes 1 millisecond each image, while learning takes 4 milliseconds per image. Caffe is indeed a deep learning platform that prioritises flexibility, performance, and simplicity. Berkeley AI Research (BAIR) and community collaborators are working on it.

## 4. Libraries we used:

**4.1. OpenCV:** The term OpenCV refers to the Open Source Computer Vision Library. One of the most widely used computer vision libraries is OpenCV. It was created by Intel and then sponsored by Willow Garage and Itseez. Under the open-source Apache 2 License, the package is cross-platform and accessible to use. OpenCV is a large open-source library for computer vision, machine learning, and image processing that currently plays a critical part in real-time operations, which are critical in today's systems. It may be used to process photos and videos in order to recognise things, faces, and even human writings. The aim of OpenCV was to create a consistent infrastructure for computer vision applications and to make machine perception more accessible. It's a huge, robust library with 2500 optimised algorithms, including a wide range of conventional and advanced computer vision and machine learning techniques.

**4.2. Numpy:** NumPY refers for "Numeric Python" or "Numerical Python." NumPy is a Python-based open source project that aims to create numerical computing quicker. It was established in 2005, based on the Numeric and Numarray libraries' earlier work. NumPy will be always 100% open source software that anyone can use [29]. NumPy is open-source and created on GitHub by the NumPy and wider scientific Python communities. NumPy is a Python module that allows you to interact with arrays. Its array class in NumPy is named ndarray, and it comes with several helper features to make operating with it a simple. In data research, when efficiency and resources are critical, arrays are widely employed. NumPy arrays, unlike lists, are stored in a single continuous location in memory, allowing programmes to acquire and modify them quickly. This is the primary reason why NumPy outperforms lists. It's also been adjusted to work with the most recent CPU configurations.

## IV. IMPLEMENTATION

**a.** ***Dataset Requirement:*** Using the OpenCV library and a deep learning pre-trained model, we attempted to recognise objects. The SSD approach was used to pre-train our model. To implement the SSD approach, we used pre-trained models from Mobilenets. On the basis of the trained model, this approach can classify labels. We used the MS-COCO dataset [22] as shown in **fig 3**, which had 91 classes. We load the input video as well as the static pictures as input and convert it to a single frame input drop by scaling each frame to a set size of 300x300 pixels. Our pre-trained models have two files: one for configuration and the other for weights. As a result, the model is a representation of how neurons are grouped in a neural network: 1- Configure and 2-Weights. The training/validation split was modified from 83K/41,000 to 118K/51,000 in 2017. The same photos and annotations are used in the new split. The 2017 testing set is a variant of the 2015 test set's 41K pictures. A fresh unlabelled dataset of 123K photos is included in the 2017 edition 7.

```
['Person', 'Bicycle', 'Car', 'Motorcycle', 'Airplane', 'Bus', 'Train',
 'Truck', 'Boat', 'Traffic Light', 'Fire Hydrant', 'Street Sign', 'Stop Sign',
  'Parking Meter', 'Bench', 'Bird', 'Cat', 'Dog', 'Horse', 'Sheep', 'Cow',
 'Elephant', 'Bear', 'Zebra', 'Giraffe', 'Hat', 'Backpack', 'Umbrella',
 'Shoe', 'Eye Glasses', 'Handbag', 'Tie', 'Suitcase', 'Frisbee', 'Skis',
 'Snowboard', 'Sports Ball', 'Kite', 'Baseball Bat', 'Baseball Glove',
 'Skateboard', 'Surfboard', 'Tennis Racket', 'Bottle', 'Plate', 'Wine Glass',
 'Cup', 'Fork', 'Knife', 'Spoon', 'Bowl', 'Banana', 'Apple', 'Sandwich',
 'Orange', 'Broccoli', 'Carrot', 'Hot Dog', 'Pizza', 'Donut', 'Cake', 'Chair',
  'Couch', 'Potted Plant', 'Bed', 'Mirror', 'Dining Table', 'Window', 'Desk',
 'Toilet', 'Door', 'Tv', 'Laptop', 'Mouse', 'Remote', 'Keyboard', 'Cell
Phone', 'Microwave', 'Oven', 'Toaster', 'Sink', 'Refrigerator', 'Blender',
 'Book', 'Clock', 'Vase', 'Scissors', 'Teddy Bear', 'Hair Drier',
 'Toothbrush', 'Hair Brush']
```

**Figure 3. MS-COCO Large Dataset.**

**b. *MobileNet-SSD-v3 Architecture:*** Google used the MobileNet network model [20] to minimize computational complexity and improve the SSD detector's real-time performance. The SSD detector's backbone network model is the MobileNet network. On a real-time basis, the MobileNet approach is used to improve the SSD algorithm and speed rating precision. To detect several objects, this method necessitates taking a single shot. For detecting purposes, the SSD is a neural network architecture design. This implies that both localization and categorization are taking place at the same time. The default box set's restricted output space is revealed by SSD. This network quickly analyses a default box for the presence of different object classes and combines the box to fit what's inside. This network also accommodates a variety of models with varying sizes of natural bonds and resolutions. We also utilised Non-Maximum Suppression to reject the majority of these bounding boxes either their confidence is low or just because they enclose the very same object like another bounding box with a high level of confidence score.

## V. RESULT ANALYSIS

The MS-COCO [21] dataset was used in our experiment to validate the detection results of the proposed pre-trained Mobilenet-SSDv3 detector. Our network model's input image format is a 512x512 RGB color filter. The proposed detector, on the other hand, can greatly improve processing speed and detection accuracy. **Figure 4 & 5** depicts our Mobilenet-SSDv3 detector's object detection results in various environments.

Our object detection method can detect objects at up to 30 frames per second. The SSD method demonstrated interior and outdoor feed video frames through camera as well as through static images in our testing, but the position of the objects differed between two consecutive frames. The video acquired by the webcam, as well as the algorithm, convert the size of a single frame to 300 x 300 pixels. The SSD can generate numerous bounding boxes for different classes with varying confidence levels by employing a higher proportion of default boxes, which can have a stronger effect, and distinct boxes for each location. Frame difference is used in this suggested single-shot multi-box detection approach.

**Figure 4. Object Detection through Static Images.**

**Figure 5. Object Detection through Webcam.**

## VI. CONCLUSION

Deep learning-based object detection has been a research hotspot in recent years. In this research, we attempted to recognise an object that was displayed in front of a webcam. MobileNet and Single Shot Multi-Box Detector were used to pre-train the generated model. Reading a frame from a web camera generates numerous problems, so a good frames per second solution is designed to reduce Input / Output concerns. Based on the Mobilenet-v3 backbone network, we suggested a lightweight network design with better feature extraction. We integrate the Mobilenet-v3 and SSD models to increase the feature map of the input image and the back-end detection network's detection accuracy. We are able to detect objects more precisely and identify them individually based on testing results, with the specific location of an object in the frame in the x, y axis. This study also includes experimental findings on various approaches for item identification and detection, as well as a comparison of the efficiency of each approach. For x86 hardware with minimal resources, this is really a huge benefit. Experiments demonstrate that the proposed Mobilenet-SSDv2 detector not only preserves the original MobileNet-SSD detector's benefit of fast processing, but also considerably enhances detection performance. This is achieved by combining two techniques: deep learning with OpenCV for object detection, and OpenCV for efficient, threaded video streaming. By using MobileNet and the SSD detector for object detection, a high precision object detection approach has been established, making it efficient to all cameras.

Our system can recognise objects in its dataset, such as cars, a motorcycle, bottles, a couch, and so on. The goal of this study is to create an autonomous system in which object and scene recognition aids the community in making the system more engaging and appealing.


## ACKNOWLEDGEMENT

I'd need to pass on my most profound appreciation Prof. Moloy Dhar for his reliable cooperation, unconstrained help, and reasonable counsel throughout a few gatherings. I additionally like the wise input and ideas from Guru Nanak Institute of Technology, which is affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), previously known as West Bengal University of Technology (WBUT).

## REFERENCES


**[1]** Younis, A., Shixin, L., Jn, S., & Hai, Z. (2020, January). Real-time object detection using pre-trained deep learning models MobileNet-SSD. In Proceedings of 2020 the 6th International Conference on Computing and Data Engineering (pp. 44-48). doi: https://doi.org/10.1145/3379247.3379264

**[2]** Vaishali, Shilpi Singh. "Real-Time Object Detection System using Caffe Model." International Research Journal of Engineering and Technology (IRJET) Volume 6 (2019). https://www.irjet.net/archives/V6/i5/IRJET-V6I5764.pdf

**[3]** Xiao, Youzi, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. "A review of object detection based on deep learning." Multimedia Tools and Applications 79, no. 33 (2020): 23729-23791. doi: https://doi.org/10.1007/s11042-020-08976-6

**[4]** Heredia, A., & Barros-Gavilanes, G. (2019, June). Video processing inside embedded devices using ssd-mobilenet to count mobility actors. In 2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI) (pp. 1-6). IEEE. doi: https://doi.org/10.1109/ColCACI.2019.8781798

**[5]** Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

**[6]** Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).

**[7]** Kanimozhi, S., Gayathri, G., & Mala, T. (2019, February). Multiple Real-time object identification using Single shot Multi-Box detection. In 2019 International Conference on Computational Intelligence in Data Science (ICCIDS) (pp. 1-5). IEEE.

**[8]** Chiu, Y. C., Tsai, C. Y., Ruan, M. D., Shen, G. Y., & Lee, T. T. (2020, August). Mobilenet-SSDv2: An improved object detection model for embedded systems. In 2020 International Conference on System Science and Engineering (ICSSE) (pp. 1-5). IEEE.

**[9]** D. Wu, N. Sharma and M. Blumenstein, "Recent advances in video-based human action recognition using deep learning: A review," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2865-2872, doi: 10.1109/IJCNN.2017.7966210. Accessed at: https://ieeexplore.ieee.org/document/7966210

**[10]** S. Hu, X. Jia and Y. Fu, "Research on Abnormal Behavior Detection of Online Examination Based on Image Information," 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2018, pp. 88-91, doi: 10.1109/IHMSC.2018.10127. https://ieeexplore.ieee.org/document/8530188

**[11]** Dhillon, Anamika, and Gyanendra K. Verma. "Convolutional neural network: a review of models, methodologies and applications to object detection." Progress in Artificial Intelligence 9.2 (2020): 85-112, https://doi.org/10.1007/s13748-019-00203-0

**[12]** Ghazal, M., Waisi, N., & Abdullah, N. (2020). The detection of handguns from live-video in real-time based on deep learning. Telkomnika, 18(6), 3026-3032.

**[13]** Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., & Lan, X. (2020). "A review of object detection based on deep learning". Multimedia Tools and Applications, 79(33), 23729-23791, https://doi.org/10.1007/s11042-020-08976-6

**[14]** Younis, Ayesha, et al. "Real-time object detection using pre-trained deep learning models MobileNet-SSD." Proceedings of 2020 the 6th International Conference on Computing and Data Engineering. 2020, https://doi.org/10.1145/3379247.3379264

**[15]** Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).

**[16]** Chiu, Y. C., Tsai, C. Y., Ruan, M. D., Shen, G. Y., & Lee, T. T. (2020, August). Mobilenet-SSDv2: An improved object detection model for embedded systems. In 2020 International Conference on System Science and Engineering (ICSSE) (pp. 1-5). IEEE. doi: https://doi.org/10.1109/ICSSE50014.2020.9219319

**[17]** Heredia, A., & Barros-Gavilanes, G. (2019, June). Video processing inside embedded devices using ssd-mobilenet to count mobility actors. In 2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI) (pp. 1-6). IEEE. doi: https://doi.org/10.1109/ColCACI.2019.8781798

**[18]** Kanimozhi, S., Gayathri, G., & Mala, T. (2019, February). Multiple Real-time object identification using Single shot Multi-Box detection. In 2019 International Conference on Computational Intelligence in Data Science (ICCIDS) (pp. 1-5). IEEE. doi: https://doi.org/10.1109/ICCIDS.2019.8862041

**[19]** Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136.

**[20]** Introducing the Next Generation of On-Device Vision Models: MobileNetV3, Available online: https://ai.googleblog.com/2019/11/introducing-next-generation-on-device.html

**[21]** MS-COCO Dataset: https://cocodataset.org/#home

**[22]** Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

**[23]** Lee, J., Bang, J., & Yang, S. I. (2017, October). Object detection with sliding window in images including multiple similar objects. In 2017 international conference on information and communication technology convergence (ICTC) (pp. 803-806). IEEE. doi: https://doi.org/10.1109/ICTC.2017.8190786

**[24]** Caffe Model. https://caffe.berkeleyvision.org/

**[25]** SSD: Single Shot Detector for object detection using Multi-Box. https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca

**[26]** Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678).

**[27]** Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets:

Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
**[28]** OpenCV: https://opencv.org/about/
**[29]** Numpy & Numarray. https://numpy.org/about/
**[30]** MS-COCO. https://paperswithcode.com/dataset/coco