ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20230709000793
Original scientific paper

EODM: On Developing Enhanced Object Detection Model using Fast Region-based Convolution Neural Networks (FRCNN)

Anuradha B., Karthik S.*, Mythili S., Kavitha M. S.

Abstract: In present scenario, in machine learning technology, computer vision technology and image processing have attained a massive growth. Amongst many branches of image processing and classification, Object Detection (OD) is the major research domain. In several domains such as face detection, self-driving cars, pedestrian detection, and security surveillance systems, object detection (OD) and classification have experienced a significant surge in popularity in recent years. The conventional techniques for object detection, such as background removal, Gaussian Mixture Model (GMM), and Support Vector Machine (SVM), exhibit limitations such as object overlap, distortion caused by environmental factors including smoke, fog, and varying lighting conditions.Though there are several methods developed for OD, the respective field still stumbles upon many confrontations at the real-time implementations. Detecting objects from the undefined background is the major problem to be considered. Hence, machine learning techniques are incorporated for detecting the objects accurately, when the Neural Networks are effectively trained. With that note, this paper develops a new model, called Enhanced Object Detection Model using Fast Region-based Convolution Neural Networks (FRCNN). For producing appropriate results, sensitivity Measurement is carried out based on brightness, saturation, contrast, Gaussian blur, Gaussian Noise and sharpness. Following this, FRCNN is trained for OD and the results are obtained. The model evaluations are carried out based on some evaluation factors with the acquired dataset images. The obtained results are compared with CNN, YOLO. The result shows that the model exemplifies the other compared works in terms of efficiency and accuracy.

Keywords: accuracy; computer vision; CNN; image processing; machine learning; object detection; sensitivity

1 INTRODUCTION

The industrial revolution has recently used computer vision in its work. Deep learning is widely used in the robotics, surveillance, medical, and automation industries. Due to its findings, which are primarily obtained in applications involving language processing, object detection, and picture classification, deep learning has emerged as the most talked-about technology. Outstanding growth is expected during the next few years, according to the market estimate. The availability of powerful Graphics Processing Units (GPUs) and numerous datasets is recognized as one of the primary causes of this [1]. The most crucial components of object detection are image categorization and detection. There are many different datasets accessible. One such extensively used image classification domain is Microsoft COCO. It is an object detection benchmark dataset. The introduction of a sizable dataset makes picture detection and categorization possible [2].

When detecting objects in an image, the classifier first trains the objects on different groups of objects, and once it has reached a certain stage, it can automatically identify all the trained objects with greater accuracy by enclosing boxes around them and identifying the label with which they had been trained. Previous research has demonstrated the effectiveness of using an SVM classifier to detect objects and confine them to text boxes [3]. Convolution neural networks have been widely utilized by researchers to extract characteristics from training items and use those features to categorize new objects [4]. One-stage and two-stage detectors, which are both options for the object detection process, are shown in Fig. 1.

In this study, a framework for building an object detection-based deep learning model that has been trained using a dataset is developed. The robustness of the trained model on degraded images is then assessed. Here, FRCNN is used for the training. Fast R-CNN is an object detection algorithm that addresses some of the issues with R-CNN. It takes a similar approach to its predecessor, but rather than using region proposals, CNN uses the picture to create a convolutional feature map, which is then used to select and warp region proposals from. The distorted squares are reshaped using a RoI (Region of Interest) pooling layer to a predetermined size so that a fully linked layer can accept them. The RoI vector is then used to forecast the region class with the aid of a SoftMax layer [5]. Only one convolution operation is performed on each image to create a feature map.

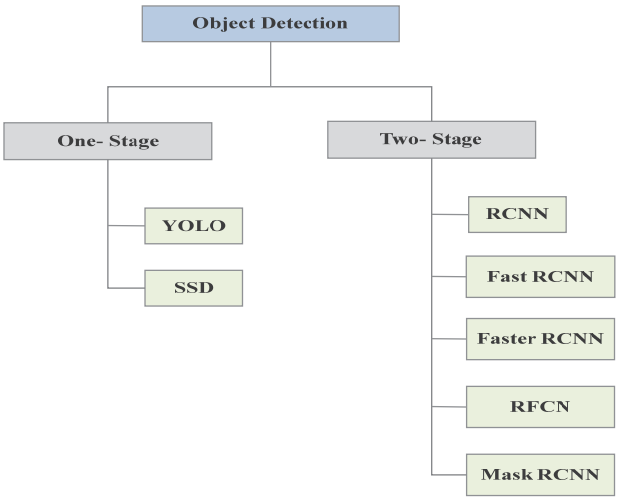

Figure 1 One and two-stage detectors for object detection

Fig. 2 describes the operations in object detection model in general.

The proposed model in this work recognizes objects using machine learning methods. Machine learning methods were used in the model's data security scheme. Sensitivity measurements are made based on brightness, saturation, contrast, blur, noise, and sharpness in order to get the right results. The findings are then collected once the FRCNN has been trained for OD. Then the classifier's effectiveness is observed, it may rapidly and precisely categorize the target subject of an image.

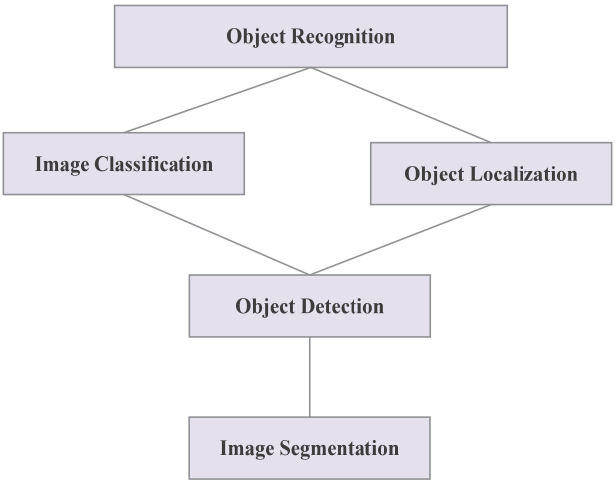

Figure 2 General operations in object detection

The remainder of this work is organized as follows: Section 2 mentions various related works in object detection. Section 3 explains the complete flow and operations in the proposed model in detail. Evaluations and comparisons are presented and discussed in Section 4. The conclusion and future directions of research are given in Section 5.

2 RELATED WORKS

In recent years, object detection has received significant study attention. With the use of effective learning tools, deeper aspects may be quickly found and examined. In order to conduct a comparative analysis and derive useful conclusions for their application in object detection, this work aims to assemble data on numerous object detection tools and algorithms employed by various researchers. The goal of a literature review is to gain understanding of our work. The Fast R-CNN model has been developed as an object detection technique as a result of Ross Girshick's work. In the area of target detection, it applies the CNN approach.

The novel aspect of Girshick's method is that it separates the training of the deep convolution network for feature isolation from the training of the support vector machines for categorization. This is done by proposing a window extraction algorithm rather than the more traditional sliding window extraction procedure in the R-CNN model. They have merged feature extraction and classification into a classification framework in the quick R-CNN approach [6].

In comparison to R-CNN, Fast R-CNN has a training time that is nine times faster. The proposal of isolation region and a little amount of Fast R-CNN are combined into a network template called the region proposal network in the faster R-CNN approach (RPN). Both Fast R-CNN and Faster R-CNN are equally accurate. The study finds that the approach is comprehensive, in-depth. In comparison to R-CNN, Fast R-CNN has a training time that is nine times faster. The proposal isolation region and a little amount of Fast R-CNN are combined into a network template called the region proposal network in the faster R-CNN approach (RPN). Both Fast R-CNN and Faster R-CNN are equally accurate. According to the reseach, the technique is a mixed, deep learning-based object detection system that operates at 5 - 7 frames per second (fps) [7].

YOLO is an additional detection network. You Only Look Once (YOLO) is a one-time convolutional neural network that has been proposed by Joseph Redmon et al. It is used to classify several candidates and predict the frame position. On this method, end-to-end target detection is possible. To tackle the object detection problem, a regression problem is used. The process of assigning the output acquired from the source image to the category and position is finished by a single end-to-end system [8].

The YOLO technique is developed for object detection in [9] that labels the name of the object that it has identified and marks boundaries of the objects with a box. It is claimed that this system is more accurate than the conventional ones. Machine learning algorithms are paving the way for text recognition. A selective search technique for locating the texts in a word document has been suggested by Gomez et al. When compared to previous algorithms, the approach constructs a hierarchy of word hypotheses and achieves an excellent recall rate [10]. The novel method presented in [11] was shown to extract the literary scenes from the natural scenes more effectively when tests using ICDAR benchmarks were undertaken.

The suggested algorithm demonstrated stronger text adaptation to texts in demanding settings as compared to existing methods. When seen in an electron microscopic stack, Convolutional Neural Networks utilized in biomedical image segmentation allow for exact localisation of neural components. Possible object positions were generated using the selective search algorithm for detecting the objects. A small number of data-driven, class-independent, high-quality locations were produced by the Selective Search algorithm, yielding 99% recall and a Mean Average Best Overlap of 0.879 at 10097 locations [12]. A variety of scalable and quick synthetic data were produced by Gupta A [13].

These artificial images were extensively utilized in the training of a Fully Convolutional Regression Network, which successfully conducted bounding box regression and text identification at all places and various scales in an image. It was claimed that the final model beats the current approaches for text detection in natural photos since it was able to detect the texts in the network substantially better. The F-measure on the common ICDAR 2013 benchmark was 84.2%. Additionally, it utilized a GPU to process 15 photos per second.

A modified approach with a revised inception model structure, a specialized pooling pyramid layer, and superior performance has been put forth by Tanvir Ahmed et al. It uses an enhanced YOLO v1 network model that optimizes the loss of function in YOLO v1. The sophisticated use of YOLO is taken from this study. Additionally, an extended experiment is conducted end to end on a PASCAL VOC (Visual Object Classes) dataset. The network is an upgraded version and performs exceptionally well [14].

A more sophisticated Single Shot Detector (SSD) is the foundation of another paper. The authors of this paper suggest that Tiny SSD, a Single shot detection deep convolutional neural network, be introduced. To make real-time embedded object identification easier, TINY SSD was created. It consists of significantly improved layers made up of a stack of non-uniform SSD-based

auxiliary convolutional feature layers and a non-uniform Fire sub-network. The size of Tiny SSD, which is even smaller than Tiny YOLO at 2.3 MB, is its strongest feature. Tiny SSD is a good fit for embedded detections, according to this study's findings [15-17].

A theoretical framework that utilizes frame difference as a means of detecting the presence of a moving vehicle is proposed. The binary frontal view of the car was obtained by applying a symmetrical filter. Subsequently, a limited Boltzmann machine with deep learning was employed on these three layers to detect the car model [18].

A research is conducted on the identification and categorization of cars through the utilization of deep neural network technology. The objective of the study was to derive higher-level characteristics from lower-level characteristics. The experimental findings indicate that the utilization of a deep neural network yields superior results in vehicle classification, as evidenced by an error rate of 3.34%. In contrast, the employment of a typical neural network yields a higher mistake rate of 6.67% [19].

Zhang et al. (2017) conducted a study focused on the identification and categorization of cars through the utilization of a deep neural network. The objective of the study was to extract. The separation of high-level characteristics from low-level features. The experimental findings indicate that the utilization of a deep neural network yields superior results in car classification, exhibiting an error rate of 3.34%. In contrast, the error rate of a conventional neural network is at 6.67% [20].

2.1 Problem Definition

From the literature review the existance of the following research problems is observed:

The task of identifying and locating small things inside photographs poses a formidable challenge that necessitates substantial advancements. CNN-based models frequently encounter challenges in accurately detecting objects that are significantly smaller in comparison to the overall size of the image.

Object detection models frequently encounter challenges when operating in unfavorable environmental conditions, including but not limited to low light levels, inclement weather, and extreme viewing angles. Conducting research aimed at enhancing the resilience of these models under diverse situations is crucial.

The investigation of privacy-preserving object recognition systems, which aim to safeguard sensitive information contained inside photos and counteract adversarial attacks, is gaining prominence in importance.

3 PROPOSED MODEL

Identifying the objects in an image (object localization) and the category to which each object belongs is the definition of the object detection problem (object classification). As a result, the pipeline of conventional object detection models can be roughly separated into the following three stages:

- Region Selection.
- Feature Extraction.
- FRCNN based Classification.

3.1 Region Selection

Since unique objects could appear anywhere in the image and have different aspect ratios or sizes, scanning the entire image with a multi-scale sliding window makes sense. This exhaustive approach has obvious shortcomings even though it can determine every possible position for the objects. Due to the vast number of candidate windows, it requires a lot of processing and produces an excessive amount of redundant windows. However, if a fixed number of sliding window templates is employed, undesired zones might be produced.

3.2 Feature Extraction

Visual features are to be extracted to offer a stable and meaningful representation in order to distinguish between distinct things. However, it is challenging to manually build a robust feature descriptor to accurately characterize all types of objects due to the range of looks, lighting circumstances, and backgrounds. For image enhancement, sensitivity measurements are carried out based on brightness, saturation, contrast, Gaussian Blur, Guassian Noise, Sharpness. Each image set receives these functions separately, after which the model is tested on each one separately.

How to select the ranges for each function was one of the first things to be taken into account in order to undertake a successful sensitivity analysis. To ensure uniformity, a set of criteria is developed for selecting the ranges, which specified the ranges required:

- To incorporate the extrema for each technique.
- To differ spatially from image to image greatly.

The minimum and highest values for each range are the extrema for each approach. In several cases, the extrema themselves were impractical, so we could only go as close to them as possible. This will be clarified in the following portion of this paper.

Ensuring the ranges featured an acceptable number of visual differences in each iteration was the second need that needed to be taken into account. The goal is to cover the widest feasible range for each approach, which means not spacing out each step too closely or too far from the next. For a few reasons, we needed a combination of linear and log spacing in the ranges in order to do this. First, it must be ensured to include both extrema and the original image within the range. In order to make the best use of available space, it is also important to verify that each step in the ranges has sufficient visual distinctions to merit inclusion.

Brightness: The brightness of the image set is adjusted as the first technique. This enables us to produce images that range from being completely black to being whitewashed, trying to imitate a certain type of imperfect image. The following transformations are given here to change the brightness levels.

$$b_r(m,n) = c_r(m,n) \cdot rate \tag{1}$$

$$b_g(m,n) = c_g(m,n) \cdot rate \tag{2}$$

$$b_b(m,n) = c_b(m,n) \cdot rate \tag{3}$$

Where, '*rate*' can be given as, [0%, 2000%] and, $b(m,n)$, is obtained from RGB rates of input images.

Saturation: This section is for balancing the image saturation rate, by calculating the pixel luminance $b_p(m,n)$.

$$b_p(m,n) = 0.299 \cdot c_r(m,n) + 0.587 \cdot c_g(m,n) + \\ + 0.114 \cdot c_b(m,n) \tag{4}$$

In the above equation, $c_r(m,n)$ is red value, $c_g(m,n)$ denotes green and $c_b(m,n)$ denotes the blue value of each image pixel. The equations for balancing saturation and providing new RGB rates are given below.

$$b_r(m,n) = c_r(m,n) - \left(\left(c_r(m,n) - b_p(m,n)\right) \times rate\right) \tag{5}$$

$$b_g(m,n) = c_g(m,n) - \left(\left(c_g(m,n) - b_p(m,n)\right) \times rate\right) \tag{6}$$

$$b_b(m,n) = c_b(m,n) - \left(\left(c_b(m,n) - b_p(m,n)\right) \times rate\right) \tag{7}$$

Contrast: The next method used in this case is to change the contrast setting. The distinction between the highest and lowest pixel values in an image is its contrast. In other words, it establishes how easily an object may be identified inside the image. Maximum contrast would result in all the colors being greatly exaggerated and much deeper than usual, making every object in the image stand out significantly. The image would be entirely gray if there were no contrast. The following formulae are used to modify the contrast levels.

$$factor(f) = \frac{259 \times (a+255)}{255 \times (259-a)} \tag{8}$$

$$b_r(m,n) = f \times \left(\left(c_r(m,n) - 124\right) + 124\right) \tag{9}$$

$$b_g(m,n) = f \times \left(\left(c_g(m,n) - 124\right) + 124\right) \tag{10}$$

$$b_b(m,n) = f \times \left(\left(c_b(m,n) - 124\right) + 124\right) \tag{11}$$

Gaussian Blur: The image set is blurred using a Gaussian filter after various image intensity parameters have been adjusted. Through a semi-transparent screen, which causes the entire image to appear out of focus, is what this replicates. The following equation is used to apply Gaussian blur.

$$b_{GB}(m,n) = \frac{1}{2\pi\sigma^2} e^{\frac{m^2+n^2}{2\sigma^2}} \tag{12}$$

where, '*m*' is the distance from the *x*-axis origin and '*n*' is the *y*-axis origin and '*σ*' denotes the standard deviation. Here, the blur radius denotes the dynamic factor in the computation that controls the modifications of resultant image.

Gaussian Noise: The final object that is frequently seen in images is noise, which can be added in several forms. Gaussian noise, speckle noise, and salt and pepper noise are three of the varieties. These three categories aid in simulating noisy images captured by low-quality or broken cameras. The following equations are used to apply Gaussian noise, the first form of noise, on the images.

$$A_G(z) = \frac{1}{2\pi\sigma^2} e^{\frac{(z-\mu)^2}{2\sigma^2}} \tag{13}$$

$$b_{GN}(m,n) = c(m,n) + A_G(z) \tag{14}$$

where, '$A_G(z)$' is the output of Gaussian Noise and '*z*'' denotes the gray rate of current pixel of the input image.

Sharpness: Although not as directly connected as the other three image qualities stated above, image sharpness and image intensity are related. With this method, the sharpness of the image is altered, effectively accentuating the edges of the objects. The produced image is the original at 0% sharpness. Every object in the final image has artificially boosted edges surrounding it as the percentages increase. The formula utilized for this computation is as follows.

$$b_{sp}(m,n) = c(m,n) + \left(c(m,n) - b_{GB}(m,n)\right) \times s \tag{15}$$

where '*s*' is the sharpness percentage.

Based on the enhanced image, the classification is processed using the FRCNN.

3.3 Fast RCNN Based Classification

Fast R-CNN is an object detection algorithm that addresses some of the issues with R-CNN. It takes a similar approach to its predecessor, but rather than using region proposals, CNN uses the picture to create a convolutional feature map, which is then used to select and warp region proposals from. The distorted squares are reshaped using a RoI (Region of Interest) pooling layer to a predetermined size so that a fully linked layer can accept them. The RoI vector is then used to forecast the region class with the aid of a SoftMax layer. Because it is not necessary to feed the CNN 2000 suggestions each execution, Fast R-CNN is faster than its predecessor. Only one convolution operation is performed to produce a feature map per image. The characteristics and operation of Fast RCNN are described in Fig. 3 below. When compared to R-CNN, this algorithm demonstrates a significantly shorter training and testing time. The algorithm's performance was found to be considerably hindered by the inclusion of region proposals. Selective Search was the algorithm used by Fast R-CNN and its forerunner to choose the region proposals. These ideas are then reshaped and categorised using RoI (Region

of Interest) pooling because this approach is quite time-consuming.

Fig. 4 shows the Fast RCNN model architecture. CNN's Convolutional layers are used to process the entire image and create feature maps. An ROI pooling layer is then used to extract a fixed-length feature vector from each region proposal. The SPP layer, which only contains one pyramid level, is a specific case of the RoI pooling layer. Then, after being fed into a series of FC layers, each feature vector is finally branched into two sibling output layers. All C + 1 categories' softmax probabilities are generated by one output layer, and the revised bounding box coordinates are encoded by the other output layer using four real-valued integers. In these processes, every parameter is optimized end-to-end using a multi-task loss.

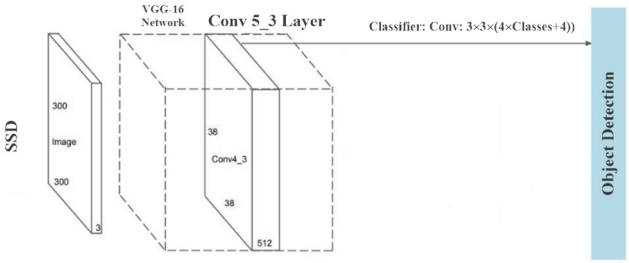

Figure 3 Operation of fast RCNN

Here, the multi-task loss '*LS*' is described as the equation presented below for training the model and bounding-box regression.

$$LS\left(d,c,t^{c},v\right)=LS_{cls}\left(d,c\right)+a\left[c\geq 1\right]LS_{loc}\left(t^{c},v\right) \tag{16}$$

where, $LS_{cls}\left(d,c\right)=-\log d_{c}$ computes the log loss rate for truth class '*c*' and 'd_c' is obtained from the discrete probability distribution $d=\left(d_0,...,d_u\right)$. And, $LS_{loc}\left(t^{c},v\right)$ is described based on the defined offset rates $t^{c}=\left(t_x^c,t_y^c,t_w^c,t_h^c\right)$, where, the box coordinates. Each 't^c' acquires the factor setting for denoting the object features based on their coordinates and the scale invariants.

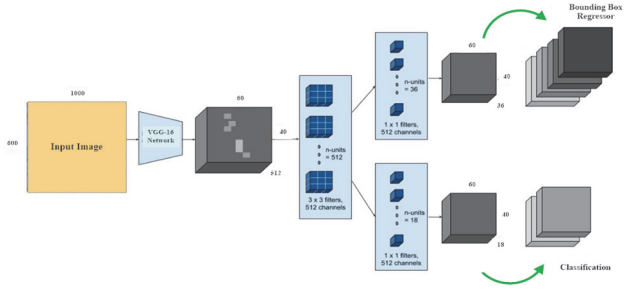

Figure 4 Framework of FRCNN model

Further, the background RoIs are removed using Iverson bracket indicator function [$c \geq 1$]. For providing the robustness aligned with outliers and removing the sensitivity in reporting gradients, smooth loss factor (SL) is obtained to fit the bounding box regressions based on the following function.

$$LS_{loc}\left(t^{c},v\right)\sum_{i\in x,y,w,h}SL_{1}\left(t^{c}{}_{i},v_{i}\right) \tag{17}$$

where:

$$SL_{1}\left(x\right)=\begin{cases}0.5x^{2} & \text{if } \left|x\right|<1\\ \left|x\right|-0.5 & \text{other wise}\end{cases} \tag{18}$$

The Fast R-CNN pipeline needs to be sped up using two more techniques. On the one hand, if training samples, the Region of Interests (ROIs), are selected from different images, back-propagation through the SPP layer becomes incredibly inefficient. After first sampling *N* random images, the fast R-CNN samples mini-batches hierarchically, with $\frac{M}{N}$ *Rols* sampled in each image. *M* denotes the quantity of RoIs. Importantly, compute and memory are shared amongst RoIs from the same image in the forward and backward passes. On the other hand, it takes a long time to compute the FC layers during the forward pass. Singular Value Decomposition (SVD) can be condensed to reduce large FC layers and accelerate testing.

4 RESULTS AND DISCUSSIONS

This section deliberates the experimental and evaluation process of the proposed model based on the classification model for detecting the attacks in real-time IoMT environment.

4.1 Dataset Information and Experimental Setup

ImageNet: The WordNet hierarchy provides the basis for the ImageNet dataset (lexicon database for English). Synset is a meaningful term in WordNet. For each synset, ImageNet offers 1000 photos. According to ImageNet's developers, the WordNet hierarchy would be represented by tens of millions of photographs. More than 14197122 photos are included. Scale-invariant feature transform (SIFT) features are present in ImageNet pictures and synsets to identify local features. The average image resolution is about 480 × 410.

Datasets CIFAR-10 and CIFAR-100: The 80 million tiny picture dataset with tagged annotations is divided into the CIFAR 10 and CIFAR 100 datasets. The CIFAR-10 dataset includes 60000 colored, 3232 resolution images over 10 item categories with a total of 6000 images per object category. It has 10000 test images and 50000 training images. The CIFAR-100 dataset consists of 100 object categories, each with 600 photos (500 training, 100 test). There are 20 super classes made up of 100 object classes. Images are given labels according to the class to which they belong, such as "fine" and "coarse" labels (image belonging to super class). For performing simulation, MATLAB Tool is used to assess the performance of the proposed Model. In addition, the model's efficiency is evaluated using criteria such as sensitivity, specificity, precision, classification accuracy, time efficiency, and model effectiveness.

4.2 Result Comparisons

The proposed model is put to the test using machine learning techniques in object detection. Additionally, by analyzing the *k*-fold cross-validation methodology, which divides the dataset into '*k*' folds at random, the effectiveness of the suggested strategy is assessed. In this study, bias and variation are reduced while maintaining an accurate estimation of error by the use of 10-fold stratified sampling. Additionally, performance criteria such as sensitivity, specificity, precision, classification accuracy, time efficiency, and model performance were used to assess the effectiveness of all explored methodologies, including the suggested approach.

In order to appropriately assess the effectiveness of the system, a special emphasis has also been placed on accuracy due to the distribution of class labels being very imbalanced and non-uniform. Therefore, based on all of the visual data, it can be said that the recommended technique performs better in object detection than all of the compared works.

The assessment metrics for model analysis in attack detection are discussed in this section. The computations are done based on True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), and False Negative (*FN*). The ROC curve between the *TP* and *FP* rates is shown in this instance. Additionally, the following formulas for estimating classification precision, sensitivity, specificity, and error rate.

Classification Accuracy: The percentage of successfully detected data to the entire number of data offered for analysis is the definition of the accuracy rate. The equation is as follows.

$$\text{classification accuracy}(CA) = \frac{\text{No.of correctly classified data}}{\text{Total data processed}} \times 100\% \tag{19}$$

Sensitivity Rate: The following calculation can be used to determine the sensitivity rate based on the T*P* rate from the categories.

$$Sensitivity\ \ Rate = \frac{TP}{TP + FN} \tag{20}$$

Specificity Rate: The formula for calculating the specificity rate, commonly referred to as the real negative rate, is as follows.

$$Specificity\ \ Rate = \frac{FP}{TN + FP} \tag{21}$$

Error Rate: A key factor in determining the effectiveness of a model is its error rate. Here, K-fold validation is used to analyze the classification results, and the average error rate is calculated as:

$$ErrorRate = \frac{1}{a}\sum_{i=0}^{a} \frac{TN_i + FN_i}{\text{Total No.of Images}} \tag{22}$$

Precision Rate: The classification of the data as abnormal determines the accuracy rate (presence of CVD). The equation is as follows.

$$Pr\,ecisionRate = \left(\frac{TP}{TP + TN}\right) \times 100\% \tag{23}$$

The performance of the proposed model is also contrasted with that of other, comparable models. The performance of the suggested model is compared with other similar intelligent models investigated in the literature, such as CNN and YOLO, based on the aforementioned measures. The performance study shows that the suggested approach is superior to other listed methods.

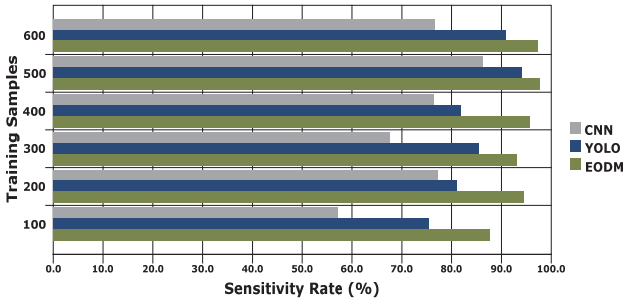

Figure 5 Sensitivity rate vs image samples

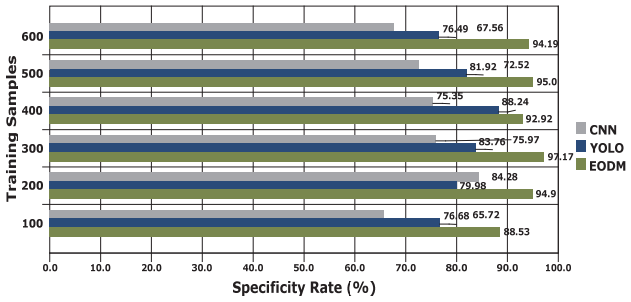

Figure 6 Specificity rate vs image samples

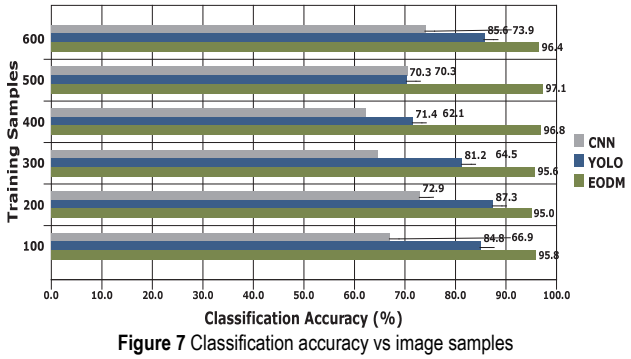

Figure 7 Classification accuracy vs image samples

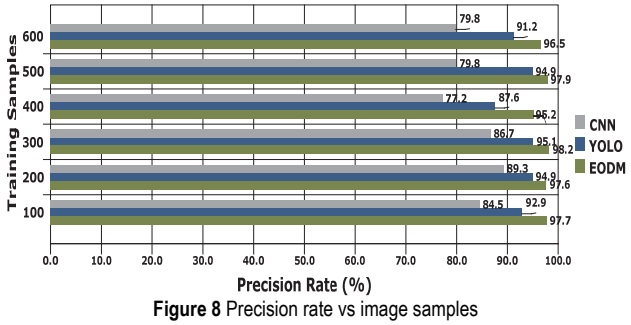

Figure 8 Precision rate vs image samples

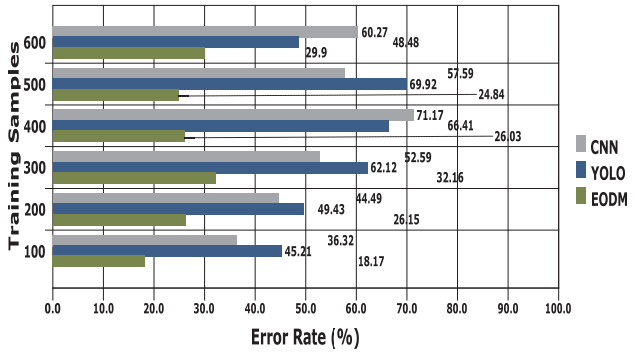

Figure 9 Error rate vs image samples

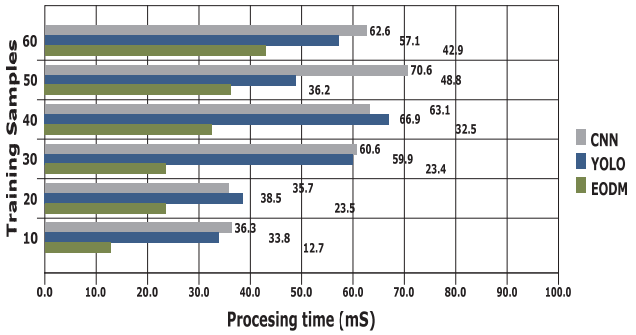

Figure 10 Processing time vs image samples

Fig. 5 shows a comparison of the results with the current model. The model achieves a specificity rate of 95.38%, which is higher than all other models evaluated, during the assessments on the training samples with the numbers {100, 200, 300, 400, 500}. The specificity rate analysis is another crucial evaluation component in a classification model, much like the sensitivity rate analysis shown in Fig. 6. The outcomes show that the recommended has a greater performance rate. The model's classification accuracy of 94.15 percent, which is higher than that of other comparable classification models employed in object identification, serves as evidence of this.

The results for Classification Rate are shown in Fig. 7. The precision rate findings for object detection are shown in Fig. 8 . The suggested model achieves a sensitivity of 94.9 percent for processing with their corresponding amount of training samples. The findings show that the recommended model outperforms rival research in terms of sensitivity. The accompanying image and data show how the model successfully raises the precision rate by using FRCNN for classification.

The error rate should be as low as feasible for a model to be useful because the results have a big impact on how decisions are made. This issue is taken into consideration when conducting the evaluations, and Fig. 9 shows the results based on the data. The graph shows that the suggested model outperforms other models in terms of attack detection error rate. Processing time is another significant component. Results are optimized and time complexity is significantly reduced as a result of successful outlier detection and feature selection. Fig. 10 shows a comparison of processing time graphs. It is clear from the aforementioned graphs that the suggested model performs well; via the analysis process, it was able to attain the highest rates of classification accuracy and precision in object detection, which were 96.23% and 97.86%, respectively.

5 CONCLUSION AND FUTURE WORK

This study presents the Enhanced Object Detection Model (EODM), a highly effective tool for the crucial task of object detection. In today's digital landscape, where countless images are shared daily on the internet, object detection is indispensable. It underpins the development of technologies such as self-driving cars, which rely on real-time image analysis to navigate safely.

The core of our model lies in the utilization of Fast Region-based Convolution Neural Networks (FRCNN) for training. We have also integrated sensitivity measurements based on key image attributes, including brightness, saturation, contrast, Gaussian blur, Gaussian noise, and sharpness, which collectively enhance our model's accuracy.While our model has demonstrated exceptional performance in our evaluations, it is important to acknowledge that the field of object detection is dynamic and continuously evolving. New algorithms and revisions to existing ones emerge each year, driving the field forward.

Further optimization of our model to achieve real-time performance on resource-constrained devices is crucial for its widespread deployment in applications like autonomous vehicles and robotics.

6 REFERENCES

[1] Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. *Procedia Comput Sci*, *132*, 1706-1717. https://doi.org/10.1016/j.procs.2018.05.144

[2] Palop, J. J., Mucke, L., & Roberson, E. D. (2010). Quantifying biomarkers of cognitive dysfunction and neuronal network hyperexcitability in mouse models of Alzheimer's disease: depletion of calcium-dependent proteins and inhibitory hippocampalremodeling. *Alzheimer's Disease and Frontotemporal Dementia. Humana Press*, *Totowa*, *NJ*, *670*, 245-262. https://doi.org/10.1007/978-1-60761-744-0_17

[3] Shin, C. S., Kim, K. I., Park, M. H., & Kim, H. J. (2000). Support vector machine-based text detection in digital video. In Neural networks for signal processing X. *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, *IEEE*, *2*, 634-641. https://doi.org/10.1109/NNSP.2000.890142

[4] Gidaris, S. & Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model. *Proceedings of the IEEE International Conference on Computer Vision*, 1134-1142. https://doi.org/10.1109/ICCV.2015.135

[5] Girshick, R. (2015). Fast r - cnn. *Proceedings of the IEEE international conference on computer vision*, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster r - cnn: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell*, *39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[7] Ding, S. & Zhao, K. (2018). Research on daily objects detection based on deep neural network. *IOP ConfSer Mater Sci Eng*, *322*(6). https://doi.org/10.1088/1757-899X/322/6/062024

[8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings ofthe IEEE conference on computer vision and pattern recognition*, 779-788. https://doi.org/10.1109/CVPR.2016.91

[9] Oral, O., Bilgin, S., & Ak, M. U. (2022). Evaluation of Vibration Signals Measured by 3-Axis MEMS Accelerometer on Human Face using Wavelet Transform and Classifications. *Tehnički vjesnik*, *29*(2), 355-362. https://doi.org/10.17559/TV-20210820150837

[10] Gómez, L. & Karatzas, D. (2017). Text proposals: a text-specific selective search algorithm forword spotting in the wild. *Pattern Recognition*, *70*, 60- 74. https://doi.org/10.1016/j.patcog.2017.04.027

[11] Zhang, Z., Shen, W., Yao, C., & Bai, X. (2015). Symmetry-based text line detection in naturalscenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2558-2567. https://doi.org/10.1109/CVPR.2015.7298871

[12] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention Springer*, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[13] Cham Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 154-171. https://doi.org/10.1007/s11263-013-0620-5

[14] Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., & Nazir, S. (2020). Object detection through modified YOLO neural network. *Scientific Programming.* https://doi.org/10.1155/2020/8403262

[15] Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018). Tiny SSD: a tiny singleshot detection deep convolutional neural network for real-time embedded object detection. *15th conference on computer and robot vision (CRV). IEEE*. https://doi.org/10.48550/arXiv.1802.06488

[16] Šulc, M. & Matas, J. (2017). Fine-grained recognition of plants from images. *Plant Methods*, *13*, 115. https://doi.org/10.1186/s13007-017-0265-4

[17] Ložnjak, S., Kramberger, T., Cesar, I., & Kramberger, R. (2020). Automobile Classification Using Transfer Learning On Resnet Neural Network Architecture. *Polytechnic and design*, *8*(1), 59-64.

[18] Gao, Y. & Lee, H. J. (2015). Moving Car Detection and Model Recognition based on Deep Learning. *Advanc. Sci. Technol. Lett.* https://doi.org/10.14257/astl.2015.90.13

[19] Zhang, Z., Xu, C., & Feng, W. (2017). Road vehicle detection and classification based on Deep Neural Network. *Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science*, Aug. 26-28, Beijing, China, 675-678. https://doi.org/10.1109/ICSESS.2016.7883158

[20] Zhang, Z., Xu, C., & Feng, W. (2017). Road vehicle detection and classification based on Deep Neural Network. *Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science*, Aug. 26-28, Beijing, China, 675-678. https://doi.org/10.1109/ICSESS.2016.7883158

[21] Rodin, C. D., Lima, L. N., Andrade, F. A., Haddad, D. B., & Johansen, T. A. (2018). Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. *International Joint Conference, Neural Networks*, 1-8. https://doi.org/10.1109/IJCNN.2018.8489465

Contact information:

Anuradha B., Associate Professor
Department of Computer Science & Engineering,
SNS College of Engineering,
Coimbatore-641107, Tamil Nadu, India

Karthik S., Professor and Dean
(Corresponding Author)
Department of Computer Science & Engineering,
SNS College of Technology,
Coimbatore-641035, Tamil Nadu, India
E-mail: profskarthik@outlook.com

Mythili S., Research Scholar
Department of Computer Science & Engineering,
SNS College of Technology,
Coimbatore-641035, Tamil Nadu, India

Kavitha M. S., Associate Professor
Department of Computer Science & Engineering,
SNS College of Technology,
Coimbatore-641035, Tamil Nadu, India