Deep Neural Network Based Smart Intrusion Detection and Alerting System

Karthikeyan N[1], Soumya Ranjan Mahapatro[2]
Assistant Professor[1], Associate Professor[2]
School of Computer Science and Engineering[1],
School of Electronics and Comm. Engineering[2],
Vellore Institute of Technology, Chennai Campus,
Tamil Nadu, India,[1,2]
narenkarthikeyan.mecse@gmail.com[1],
soumyaranjan.mahapatro@vit.ac.in[2]

Akash R[3], Rakesh Teja P V[4]
UG Scholars,
School of Computer Science and Engineering,
Vellore Institute of Technology, Chennai Campus,
Tamil Nadu, India,
akashrajkumar499@gmail.com[3],
rakeshteja926@gmail.com[4]

***Abstract* - With the increasing prevalence of surveillance systems, the need for an automated system that detects the presence of intruders from the surveillance camera and send an alert message to the user immediately has become very crucial. The research paper aims to provide a deep learning (DL) model-based solution to accurately detect the presence of human figures in the frame. The approach includes using YOLOv8 (You Only Look Once – Version 8) object detection model to detect human figures, paired with the implementation of the Non-Maximum Suppression technique to remove duplicate bounding boxes around humans. Additionally, Gaussian Noise Reduction technique has been used to remove Gaussian blur from the image for obtaining a clearer image of human figure present in the frame. The research paper demonstrates a procedure to quickly send alert message to the user when a human figure is detected in the camera using the Telegram application. The paper also demonstrates the implementation of frame skipping method based on periodic intervals for optimizing the computational efficiency. In addition to this, the paper also demonstrates a comparative study between the frequently used human detection algorithms namely YOLOv8, YOLOv4, SSD and HOG and provides statistical proof on how YOLOv8 outperforms the other standard models.**

Keywords: **Deep Neural Network, Intrusion Detection, YOLO model, DL based Human detection, YOLOv8 .**

I. INTRODUCTION

Over the years, the demand for a reliable real-time surveillance system has increased across various applications. Several deep Learning based models are used to identify intrusion detection. One of the most popular model is YOLOv8 [1] which can detect humans and track their activity. YOLO uses a single stage object detection approach. In this approach, the image is divided into various grids. The predictions for the objects and bounding boxes are done directly at each grid cell. The YOLO model performs the entire detection process in a single pass. It enables YOLO to operate efficiently and swiftly, making it well-suited for real-time applications. Two major process are crucial in object detection. The first process is object classification. It is a method of identifying the nature of an object and classifying it into a pre-defined category such as human, car, cat, etc. YOLO has various classes for different type of objects. When the image is passed into the neural network, key features from the image such as edges and textures of the object are extracted by the neurons [2]. This data is now compared with the pre-trained data. Based on the comparison, predictions are made, and neurons return a prediction score called confidence score [3]. If there is a human in the image, the confidence value of the human class will be high and the value for the remaining classes will be very less. This helps in classifying that the object present in the image is a human. After identifying the nature of an object in object localization [4], the next stage is to identify and locate the object. To highlight the position of the object, rectangular boxes called bounding boxes are drawn around the object. The position of the identified object is determined with the help of the extracted features such as the edge pixels of the object. The features [5] are formed as a map called feature map. This map maps to the edge pixels of the image. The pixels that correspond to the object are identified with the help of this map. After this process, it is necessary to determine the coordinates of the bounding box in order to draw box around the object. For this purpose, the feature map in the convolutional layer contains localization heads. Localization heads make multiple predictions and as a result, multiple boxes are drawn around the object [6]. It is necessary to find the most precise and confident box and eliminate the others to prevent clustering and overlapping of boxes. For this purpose, a post processing technique called non-maximum suppression [7] is applied. This filters out redundant boxes ensuring that only the confident ones are present. The human detection system uses the human class of the YOLO model to detect and track humans. The video is sent frame by frame into the neural network. The neurons then make predictions and return confidence scores for each frame. Based on the confidence scores, bounding boxes are drawn. The Non maximum suppression algorithm eliminates redundant bounding boxes by choosing the prediction with the highest confidence score among overlapping predictions. To the part of the frame where boxes are drawn, gaussian blur is used to reduce the gaussian noise. The gaussian blur method works by averaging the color values of pixels within a defined radius, with more weight given to pixels closer to the center. This creates a gradual transition between colors, reducing image noise and thus enhances the detailing in the image. Finally, screenshot of the frame where detection occurs in the video is sent to the user along with an alert message through telegram. J.Redmon et al.[3]

proposed YOLO model which uses a single stage architecture for object detection. In this architecture, the neural network performs predictions for the bounding boxes and class probabilities from the input image in a single evaluation [8]. When image is passed into the neural network, the neurons act on each of the divided grid cells and makes predictions during one evaluation. This approach eliminates the need for intermediate stages like region proposal generation, which works by generating a set of regions in an image and using sliding window approach to let the neurons analyse each region. Pipelining is required to implement such methods. But YOLO eliminates the need for multiple stages, thus making it faster and efficient.

To understand the working of YOLO model, Geethapriya.S et al. [6] has provided a detailed overview on the working of the model. The bounding boxes are predicted using grid labels. The labels are generated by the neurons when working on a particular grid. They include parameters such as the center coordinates, width and height. With this data, the coordinate and dimensions of the bounding box is determined, and the boxes are drawn around the object. The loss function involves calculating the difference between the predicted bounding or detection region and the ground truth region coordinates. It is used during the training of the model to adjust the model's parameters so that it can better predict bounding boxes. Y M Jaswanth Kumar et al. [8] introduced a real-time object detection model employing the YOLO algorithm to aid visually impaired individuals by detecting objects around them and generating automated voice to guide them safely reducing the need for human assistance to them. Their research demonstrates the impressive capabilities of the YOLO model in real-time object detection scenarios. YOLO has a significantly lesser inference time to find the type of object compared to the other models due to its single stage architecture approach, where different parts of the neural network analyse different parts of the image simultaneously and return the predictions scores spontaneously, thus making it faster and more suitable real-time applications. Wei Liu et.al.[9] proposed SSD (Single Shot MultiBox Detector) designed to detect objects by using default boxes, that are predefined detection regions distributed across various aspect ratios and scales. These default boxes serve as the basis for predicting object classes and refining bounding box adjustments. SSD makes use of multiple feature maps, allowing the model to adeptly address objects of varied sizes. The predictions from these feature maps are combined to enhance the overall detection performance. SSD uses a single stage approach unlike the traditional methods thus providing faster performance. SSD model does not differentiate the different levels of features that are present in the image making it less efficient for detecting small and tiny objects in the image. To address this problem, Songmin Jia et al.[10] introduced a modified SSD model. This model combines shallow layers, which capture broad information, with deep layers that focus on fine object details. This results in a feature fusion that provides a more precise analysis of the detected object. M. Kachouane et al. [11] suggested a method based on the Histogram of Oriented Gradients (HOG) model. The model has a two-stage architecture. The first stage involves extracting low-level features from an image and create a feature descriptor and the second stage involving the usage of a machine Learning algorithms like Support Vector Machine (SVM) to classify the type of image based on the feature descriptor. The HOG model splits the image into multiple cells and extracts the local features present in it and combines all of them together for normalizing the result. Due to this technique the model may not properly the capture the high-level context of the image. Also due to its two-stage architecture it usually takes more time when compared to modern deep learning models like YOLO.

II. Proposed Scheme: DL Based Smart Intrusion Detection and Alerting System

In this section, a detailed explanation on how the YOLO model works and what goes inside detecting the human figures present in the frame starting from detecting the presence of object, finding the class to which the object belongs to calculating how confident or how certain the model is that the predicted object including its class label is correct. The working model of YOLO is shown in figure 1.

A. Objectness score

The objectness score signifies the presence of object at a particular pixel of the image. The YOLOv8 model uses a Sigmoid Activation Function to predict whether the object is present or absent in the image. The Sigmoid Activation Function is a type of binary classification method where the output is in the range of 0 to 1., is given in eqn. 1

$$\alpha(Z) = \frac{1}{1+e^{-z}} \quad (1)$$

where ‘z’ represents the objectness score predicted.
If the predicted score is very large (approaching positive infinity), the sigmoid activation will output a number close to 1. This implies that there is a high probability of an object being present in the image. If the predicted score is very small (approaching negative infinity), the sigmoid activation will output a number close to 0. This signifies that there is very less confidence of an object being present in the image. If the confidence score of the object being there is more than the threshold value set, a bounding box will be drawn to indicate the presence of an object in the image.

B. Class Prediction

The YOLOv8 model has multiple classes in it and each class corresponds to the type of object like Humans, Dog, Cat etc. The model uses Softmax Activation to determine the class of the object. It is a type of multi-class classification method in which probability scores are assigned to all the classes in the model and the class with highest probability value will be considered as the class to which the detected object belongs to. The summation of probabilities assigned to all the classes sum up to 1. The softmax activation formula is given in eqn. 2.

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \quad (2)$$

where ‘zi’ represents the raw score for class $_i$ and softmax(z)$_i$ represents the transformed probability of class $_i$.

C. *Confidence Score*

YOLO partitions an image into a grid of dimensions SxS upon passing it into the neural network. Each grid cell is processed by neurons and these neurons generate predictions for detection regions along with their respective confidence scores. The confidence scores represent the model's confidence level regarding the existence of an object within the area. IOU, or Intersection over Union, is an evaluation metric used to measure the overlap between a bounding box and its corresponding ground truth box, evaluating the accuracy of localization. IOU is computed by dividing the intersection area by the union area between the bounding box and the ground truth box, as shown in equation 3..

$$IOU = \frac{Area\ of\ Intersection\ (A \cap B)}{Area\ of\ Union\ (A \cup B)} \quad (3)$$

By utilizing the coordinates of both the predicted bounding region and the ground truth box, the YOLO model identify the intersection and union region, calculates the area and returns their ratio as IOU. Object confidence score (OC) indicates the model's accuracy and certainty regarding the presence of an object within the box. OC is calculated using eqn. 4

$$OC = Pr_{(Object)} * IOU \quad (4)$$

Where $Pr_{(Object)}$ denotes the probability that the object is present inside the bounding box and IOU is Intersection of Union given in eqn. 3. This probability value is obtained from the objectness score which is determined using sigmoid function. To determine whether the predicted object belongs to a specific class, OC from eqn. 4 is multiplied with the class probability $Pr(Class_i|Object)$. This value is referred to as class confidence score, commonly called as confidence score and is given in eqn. 5.

$$CS = Pr(Class_i|Object) * Pr(Object) * IOU \quad (5)$$

Where CS denotes the Confidence_Score.

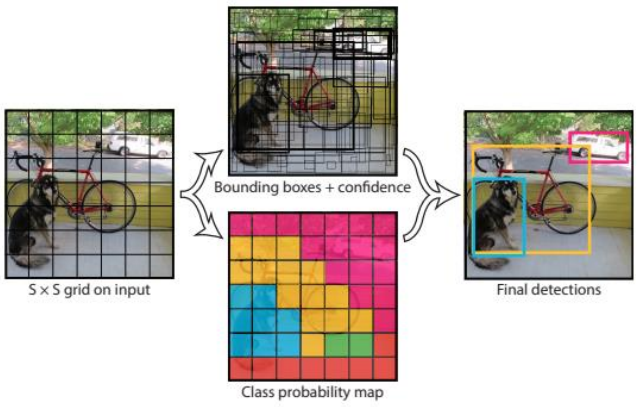

Figure 1 Working of YOLO model [1]

To ensure the confidence is accurate, a confidence threshold is established. If the confidence value exceeds this threshold, the prediction is reliable and valid. The bounding boxes with higher confidence scores are initially drawn around the object. After applying non maximum suppression, redundant bounding boxes are removed, and the final box is displayed. The class probability map assists the algorithm in recognizing the existence of multiple objects belonging to various classes within the same image. With the help of the map, different bounding boxes are assigned their respective class names. The class names and the confidence score are generally displayed above the bounding box in the final result.

III. BLOCK DIAGRAM OF THE PROPOSED SCHEME

Initially, the YOLO model has been set to video as input then set the Confident threshold score as per requirement. The workflow of proposed scheme of DL based Intrusion detection system is shown in figure 2. To make the model computationally efficient, implement the method of skipping frames at a periodic interval and only every 'n^{th}_frame' (n can be value of our choice) is processed. The value of the variable 'n^{th}_frame' decides the number of frames to be skipped before processing a frame and process the frame only if it is a multiple of the variable `$n^{th}$_frame` then pass the n^{th}_frame into the YOLO model to get detections and draw bounding boxes on the humans detected based on confidence scores. To eliminate redundant and overlapping bounding boxes, the Non-Maximum Suppression (NMS) technique is applied. For each box detected, extract the region, apply Gaussian blur function to reduce noise, and replace the original region with the noise-reduced region and send an alert message along with the screenshot of the detected noise reduced frame using the Telegram bot. By analyzing every n^{th}_frame instead of all the frames, we strike a balance between real-time processing and reducing computational overhead. If the n^{th}_frame value is higher, fewer frames are analyzed, resulting in faster processing but potentially missing some detections. If it is lower, more frames are analyzed, but processing may become slower.

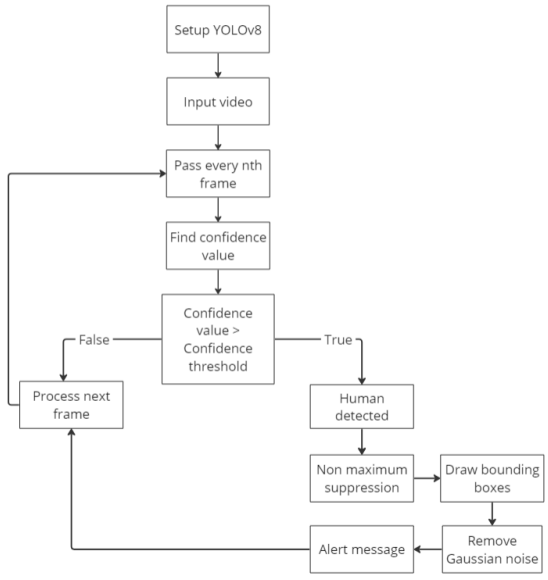

Figure 2 Block Diagram of DL Based Smart Intrusion Detection and Alerting System

IV. PERFORMANCE METRICS

In this section, the efficiency of this model is evaluated and compared with other standard models by the various performance metrics that are True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN), Precision(P), Recall(R), F1_Score, and standard deviation. The details of various performance metrics have been discussed. TP finds the number of cases where intruders are present in the video frames, and the model correctly predicts them as intruders. FN finds the number of cases where intruders are actually present in the video frames, but the model fails to predict them. Whereas TN finds the number of cases where intruders are not present in the video frames, and the model does not predict them and FP denotes the number of cases where intruders are not present in the video frames, but the model makes false prediction by detecting non-human instances or other objects.

A. Precision

Precision (P) is defined as the ratio of true positive predictions to the sum of true positive and false positive predictions as shown in eqn. 6. Precision provides information regarding the reliability of positive predictions.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

B. Recall

Recall (R) measures the ability of the model to identify all positive instances. It is the ratio of correctly predicted positives to the total actual positives, given in eqn. 7.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

C. F1_Score

F1_score is the harmonic mean of precision and recall, given in eqn. 8. It provides a balanced overview about the model, which is valuable in scenarios with imbalanced class distribution, by considering both the precision and recall of a model.

$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (8)$$

To determine how accurate the model is in detecting humans, accuracy percentage is calculated using eqn. 9.

$$Accuracy\_Percentage = F1\_Score * 100 \quad (9)$$

D. Standard Deviation

This paper considers Standard Deviation as an Evaluation Metric which signifies the reliability of the model. To find the standard deviation of the model, consider all the instances where True positive is detected. Sum up the confidence score of all the instances where True positive is detected and find the total number of occurrences of True positive. Find the mean of confidence scores. Then, find the standard deviation of all the confidence scores. If the standard deviation is higher, it signifies that the model lacks the reliability and is inconsistent in predicting the presence of human since there is variation in confidence scores though there is presence of human. If the standard deviation is less, it signifies that the model is reliable, consistent and accurate in correctly predicting the presence of human with almost similar confidence score. For example, if at one iteration, the model predicts a confidence score of 0.8 and at the very next iteration, it predicts the confidence score as 0.2. This is an example of the model being unreliable because though there was presence of human at both consecutive iterations which are separated only by a second, there is very little change in the position of the detected object. The model shows high deviation in confidence scores, thus making it less trustworthy in real-time scenarios. The standard deviation between confidence scores can be calculated using eqn. 10.

$$standard\ Deviation\ (\sigma) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (10)$$

Where N represent the number of frames processed, x_i represents the confidence score of i^{th} frame and μ represents mean of confidence scores.

V. EXPERIMENTAL RESULTS

A confusion matrix presents the models predictions alongside the actual outcomes, displaying TP, TN, FN and FP. Confusion matrices help in understanding the performance of the model. In Table 1, the confusion matrix of all four models is given to showcase the performance metrices of the model.

TABLE I. CONFUSION MATRIX OF ALL MODELS

Model	***TP***	***FN***	***FP***	***TN***
YOLOv8	73.6%	23%	3.4%	__
YOLOv4	66.7%	27%	5.3%	__
SSD	52.4%	39.7%	7.9%	__
HOG	48.9%	37.6%	13.5%	__

The value for TN is left empty because all possible bounding boxes in the frame where human is not present is true negative. It is practically impossible to count all such boxes as it tends to a very large value or infinity. From the confusion matrix of the models, it can be observed that the YOLOv8 model has the highest number of TP and lesser FN and FP compared to the other models. Higher TP indicates more true detections which is very crucial for the proposed system. SSD has a moderate TP and FP values, but more FN compared to the other models. HOG has high number of FP making it unreliable for the system because alert message will be sent even for false cases if the model predicts a bounding box.

TABLE II. COMPARISON RESULTS OF PRECISION, RECALL, F1_SCORE FOR ALL FOUR MODELS

Model	***Precision***	***Recall***	***F1 Score***
YOLOv8	0.956	0.762	0.84
YOLOv4	0.926	0.712	0.80
SSD	0.869	0.569	0.68
HOG	0.784	0.565	0.65

From Table 2, it is observed that YOLOv8 has a very high precision value. This suggests that YOLOv8 has a very good accuracy of positive predictions than compared to other models. SSD has produced a good precision, suggesting that its positive predictions are accurate, BUT not as high as YOLO. HOG has a considerably low precision value. This is because of its high FP.

Higher FP reduces the positive prediction rate and thus a lower precision value is obtained. The YOLOv8 model demonstrates a relatively high recall value, indicating that it effectively captures most of the TP instances. YOLOv4 SSD and HOG models have a moderate recall value, suggesting that it captures a lower proportion of TP instances compared to YOLOv8.

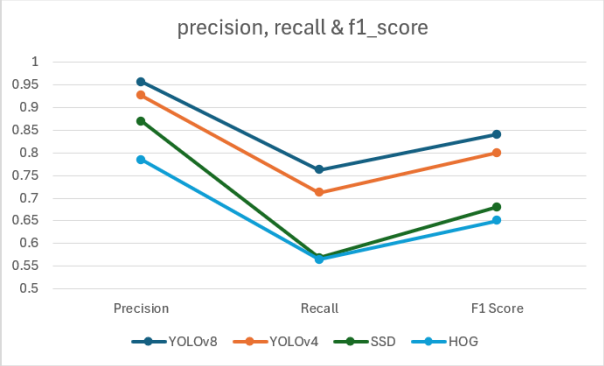

Figure 3. Comparison of Precision, Recall, and F1_Score values

TABLE III. COMPARISON RESULTS OF ACCURACY FOR ALL FOUR MODELS

Model	***Accuracy (%)***
YOLOv8	*84.80*
YOLOv4	80.51
SSD	68.76
HOG	65.68

YOLOv8 has a very good accuracy percentage of about 84% making it the best choice for the system. On computing the standard deviation for the models, the following results were obtained, is tabulated in Table 4.

TABLE IV. STANDARD DEVIATION OF ALL MODELS

Model	***Standard Deviation***
YOLOv8	*0.196*
YOLOv4	0.221
SSD	0.234
HOG	-

The lesser the standard deviation, the more the model tends to the reliable and consistent in making predictions. The HOG model does not give confidence scores for each iteration. Hence, standard deviation metric is not applicable here.

YOLOv8 due to its single scale feature map prediction and single stage architecture performs faster and more accurate than SSD and HOG. The YOLO model also offers flexibility in model size through its YOLOv3, YOLOv4, YOLOv8 tiny versions for a resource limited environment. Also due to its continuous architecture evolution along the incorporation of new features and refined parameters guarantees enhanced performance compared to its previous traditional models, rendering it well-suited for real-time applications. HOG uses a traditional two stage architecture making it slow and inefficient compared to the single stage architecture models.

Figure 4. (a) – (c). Resultant Images of YOLOv8 model at different intervals

Figure 5. (a) – (c). Resultant Images of YOLOv4 model at different intervals

Figure 6. (a) – (c) Resultant Images of SSD model at different intervals

Figure 7. (a) – (c) Resultant Images of HOG model at different intervals

On implementing the intrusion detection system using the four models, the following results were obtained. YOLOv8 performs exceptionally well in most of the cases. Even in instances of low light, the model managed to accurately predict the intruder as shown in fig. 4. (a)-(c). The model is also able to identify two intruders in Fig. 4. (b) and detects them accurately. YOLOv4 performs well to predict the intruder accurately, and gives the results close to YOLOv8 model, as shown in Fig. 5(a) - 5(c). Even when they are very close to each other, the model is able to differentiate them and predicts two bounding boxes for each of them. SSD is able to identify only one intruder in Fig. 6 (a) and fails to detect both of them in the next case in Fig. 6(b) - 6(c). The model also fails in some low light instances. Similarly, HOG fails to identify two intruders and puts a box only for one of them. In fig. 7. (a) - (c), the model makes a false prediction and fails in some low light instances similar to SSD. From the comparisons and results obtained, YOLO has clearly outperformed the other models. On implementing the system using YOLO, the model successfully detected intruders and the alert message was sent spontaneously, as shown in Fig. 8.

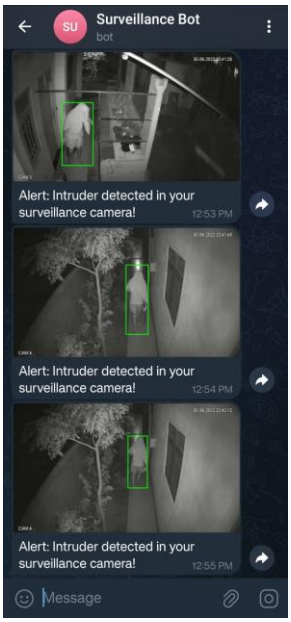

Figure 8. Notification received on the user's device

VI. CONCLUSION

The research has implemented a YOLO based intrusion detection system combined with Non-Maximum Suppression and Gaussian Noise Reduction which has proved to be efficient and robust in fulfilling the requirement. To illustrate the effectiveness and efficiency of the proposed DL-based intrusion detection scheme, various standard performance measurements are employed, and the results are compared against all three standard models. From the observations of comparison table 2-4, the implementation of frame skipping method at appropriate intervals has also proved to be an effective approach and has brought a right balance between managing computational resources and accuracy in detection. The experimental results have also shown that the YOLO based model has significantly outperformed the SSD and HOG models. The proposed scheme has clearly showed the practical applicability of the proposed model for real-time applications.

REFERENCES

[1] Nguyen Thai-Nghe., Huu-Hoa Nguyen., Wonhyung Park., & Quang Thai Ngo, "Human Intrusion Detection for Security Cameras Using YOLOv8." Intelligent Systems and Data Science, pp. 220-227, 2023

[2] Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. “Scalable object detection using deep neural networks”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2147-2154, 2014

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. “You only look once: Unified, real-time object detection”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[4] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. “Overfeat: Integrated recognition, localization and detection using convolutional networks”. arXiv preprint arXiv:1312.6229, 2013.

[5] Ren, S., He, K., Girshick, R., Zhang, X., & Sun, J. “Object detection networks on convolutional feature maps”. IEEE transactions on pattern analysis and machine intelligence, Vol. 39(7), pp. 1476-1481, 2016.

[6] Geethapriya, S., Duraimurugan, N., & Chokkalingam, S. P. “Real-time object detection with Yolo”. International Journal of Engineering and Advanced Technology (IJEAT), Vol. 8(3S), pp. 578-581, 2019.

[7] Hosang, J., Benenson, R., & Schiele, B. “Learning non-maximum suppression”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4507-4515, 2017.

[8] Kumar, Y. J., & Valarmathi, P. “YOLO Based Real Time Human Detection Using Deep Learning”. In Journal of Physics: Conference Series, Vol. 2466(1), pp. 012-034, IOP Publishing, 2023.

[9] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. “Ssd: Single shot multibox detector”, pp. 21-37,. Springer International Publishing, 2016.

[10] Jia, S., Diao, C., Zhang, G., Dun, A., Sun, Y., Li, X., & Zhang, X. “Object Detection Based on the Improved Single Shot MultiBox Detector”. In Journal of Physics: Conference Series, Vol. 1187(4), IOP Publishing, 2019.

[11] Kachouane, M., Sahki, S., Lakrouf, M., & Ouadah, N. “HOG based fast human detection”. In 2012 24th International Conference on Microelectronics (ICM), pp. 1-4. IEEE, 2012.

[12] Dalal, N., & Triggs, B. “Histograms of oriented gradients for human detection”. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, pp. 886-893, IEEE 2005.

[13] Patwary, M. J. A., Parvin, S., & Akter, S. “Significant HOG-histogram of oriented gradient feature selection for human detection”. International Journal of Computer Applications, Vol. 132(17), 2015.

[14] Viraktamath, D. S., Navalgi, P., & Neelopant, A. “Comparison of YOLOv3 and SSD algorithms”. Int. J. Eng. Res. Technol., Vol. 10(2), pp. 193-196, 2021.

[15] Li, Meian, et al. "Research on object detection algorithm based on deep learning." Journal of Physics: Conference Series. Vol. 1995. No. 1. IOP Publishing, 2021.