Journal of Intelligent Computing System (JICS), Vol.1, Issue. 2, pp. 34-42, (May 2026)
ISSN: 3139-3616 (online)

Object Detection and Identification in Realtime Using Deep Learning

Nisha Pal[1], Sanjay Kumar[2], Anurag Tomar[3], Pakhi Rajpoot[4] Aakanshi Garg[5], Shayan Ahmad[6]

[1,2,3,4,5,6]Galgotia College of Engineering and Technology, Greater Noida, India
[1]nisha.mnnit@gmail.com, [2]skhakhil@gmail.com, [3]anurag.22gcebaids004@galgotiacollege.edu,
[5]aakanshi.22gcebaids029@galgotiacollege.edu, [6]shayan.23gcebaids077@galgotiacollege.edu

Abstract: Computer vision has generally been guided through deep learning, primarily in terms of thinking about devices in snapshots. Convolutional neural community-based absolute models have completely changed the panorama. Models select features in their own unique way, saving countless hours of manual work. Object identification involves more than just identifying equipment in an image; It also contains a clue to their special locations. This has become more important for medical imaging or driverless cars.ork, even to more intelligent retail systems. However, there is a problem. Deep learning models require vast, diverse, and intensive datasets. The process of gathering and labeling images is costly and time-consuming. This is when data augmentation is useful. To expand their raw datasets, people employ a variety of techniques such image flipping, rotation, cropping, and brightness adjustment. It is really beneficial, but those adjustments are really easy. For instance, they don't deal with situations when one object is in front of another, complicated backgrounds, dim lighting, or even instances where some categories are hardly represented in the data. As a result, models continue to struggle with the chaos that occurs outside of the laboratory. The concept was used to develop a more sophisticated pipeline for data augmentation. This means that occlusions should be simulated, context should be incorporated, and the changes that are most important for the task should be added. As a result, the model won't become stuck on training data and will learn better patterns.

Keywords: *Computer Vision; Data Augmentation; Deep Learning; Object Detection; Task-aware Augmentation.*

1. Introduction

Artificial intelligence has been on the rise in recent times. The new machines analyze images and videos and understand them, something that we always thought was only possible for humans. At the core of it all is the object detection model. This is of great importance, for instance, in traffic monitoring, self-driving vehicles, medical imaging, and intelligent security. Deep learning has been one of the main game changers by far, especially when talking about Convolutional Neural Networks, or CNNs. Feature extraction was a labor-intensive process; researchers had to do this manually, which results in overfitting, as demonstrated by authors in [1], going for the edges, the corners, and the textures. These days, CNNs learn the handcrafted features, as demonstrated in [2]. This makes the process faster and more precise. These models have to be trained on hundreds of thousands of diverse images to generalize well, which has been widely discussed in object detection surveys [3], if their performance is to be satisfactory. Acquiring such a plethora of photos and then annotating them is a time-consuming task. Sometimes, the entire thing is just impractical. Data augmentation
is becoming more and more popular and widely used in reducing overfitting in deep learning systems. Simply take the images, do some operations on them, mirror them, rotate them, crop them, and even change the lighting, as mentioned in the paper [4]. It can be observed that more images for training can help avoid overfitting to one specific look . It is a pretty straightforward approach, and the reality is that these techniques do little, if anything, in preparing models for the kind of situations that they are bound to encounter outside the lab, such as objects being half-covered, low lighting, and messy surroundings. They are not good enough for real-world applications. To solve this problem, the CutMix data augmentation method was proposed, which focuses on half-visible objects. The approach used was cutting edges of pictures and pasting them with other pictures, as mentioned in [5]. For instance, they cannot grasp a scenario where one object is hiding another, or the light source is changing, or the background is a mixture of objects or is moving. Hence, the models trained with only these simple transformations tend to have a high performance in laboratory conditions, as proved by [6], but they lose a lot of their power when it comes to real-world tasks such as surveillance, traffic monitoring, or autonomous navigation.

Additionally, there is a large demand for annotated datasets, which are laborious and expensive to make, so decided to

© 2026 JICS. Personal use is permitted, but republication/redistribution requires JICS permission.
see: https://synergyworldpress.com/journal/jics/about/overview

doi.org/10.67420/109319.1.2.5

generate synthetic data efficiently. If only a small amount of data is at hand, the models can incline towards overfitting by recalling rather than learning. So, the model's performance in terms of detection accuracy on new data can be very weak. Another big problem of augmentations is that they are task-unaware. Most techniques that modify bounding boxes distort them, and thus, if the integrity of the boxes is not maintained, the detection precision decreases significantly. Such models have limited power to oppose different challenges when they are only trained with generic augmentations. These problems emphasize the necessity for better augmentation that can generate more realistic data, which was approached by , which will result in less usage of large annotated datasets.

Models used by authors [7] in their training and testing phase show a set of steps for development. The authors in their model captured images from live video or pictures, preprocessed them; after preprocessing, they normalized them and applied several data augmentation methods for object detection in their projects. After successful augmentation, they used various pretrained models for training their model, such as traditional CNN, YOLO, and Faster R-CNN. They applied various evaluation parameters such as mAP, recall, precision, and loss.

The paper's remaining sections are arranged as follows. Related research on deep learning models and object detection is reviewed in Section 2. The suggested methodology, which includes feature engineering, data preprocessing, and model training, is explained in Section 3. The experimental findings and the suggested system's performance are covered in Section 4. Section 5 wraps up the report and suggests possible topics of inquiry for further research.

2. Related Work

Since 2019, this topic has been very popular among researchers: the use of data augmentation to improve deep learning-based object detection. For example, [8] and his team produced CutMix, which involves swapping random patches between images. The outcome is that such a simple operation helps models to extract more powerful features and to overcome phenomena like occlusion, i.e., the blocking of one object by another. Then there is the work of Cubuk et al., 2019 and his collaborators, who presented Auto Augment. This is a system that, via the use of reinforcement learning, searches for the best augmentation algorithms for the given dataset. Consequently, the models in question not only become more accurate but also generalize better.

Zhao et al. work became relevant when the data was scarce. They synthesized the training images using GANs and merged them with classical augmentation methods. Their models could handle strange lighting situations or overlapping objects, which is very important for real-life scenarios. Also, when there is a deficiency of labelled images, Gu and his team took the plunge into Semi- Supervised Object Detection and demonstrated that data augmentation not only stabilizes pseudo- labels but also from the unlabeled images, which is a huge progress. Apart from the typical strategies for increasing training datasets, as indicated in previous studies, researchers have continued their work to reduce the scarcity of real-world training datasets by leveraging synthetic data generation and hybrid augmentation pipelines [9], using synthetic datasets as an alternative solution to this issue.

In 2025, for example, published an article entitled FLORA, which was created for the purpose of generating optimal synthetic samples through a series of efficient algorithms and processes. Furthermore, the authors concluded that the combination of high-quality synthetic datasets, even if they are very small, can significantly enhance the accuracy of he authors of this paper developed an end-to- end augmentation technique that was purpose-made to optimize object detection algorithms. Building on the work done by [10], they also took advantage of improvements created from advances in bounding box detection to show that relatively simple augmentation techniques, such as the use of an object bounding box to perform jittering and cropping to create new images of the same object, can lead to improved performance when compared against a baseline model that did not use augmentation techniques.

Research was concentrated on very broad-based reviews of image enhancement workflow types and also established a basis toward understanding the direct value of enhancing images through augmentation. In [11], authors compiled a large amount of data from their review of current trends in Convolutional Neural Networks (CNN) for object detection, as well as the increasing prevalence of Vision Transformer (ViT) models, showing that augmentation pipelines used in many modern architectures are closely related. Classical reviews such as Liu et al. [12] display the evolution of object detection, from manual feature extraction and background noise removal to modern deep learning approaches, repeatedly highlighting augmentation as a primary factor contributing to increased robustness.

3. Proposed Work

This section describes all structures and methods of the proposed work.

Proposed Architecture

Object recognition model functions and commands, from taking a real-time image to developing a robust and accurate object recognition model as shown in Figure 1

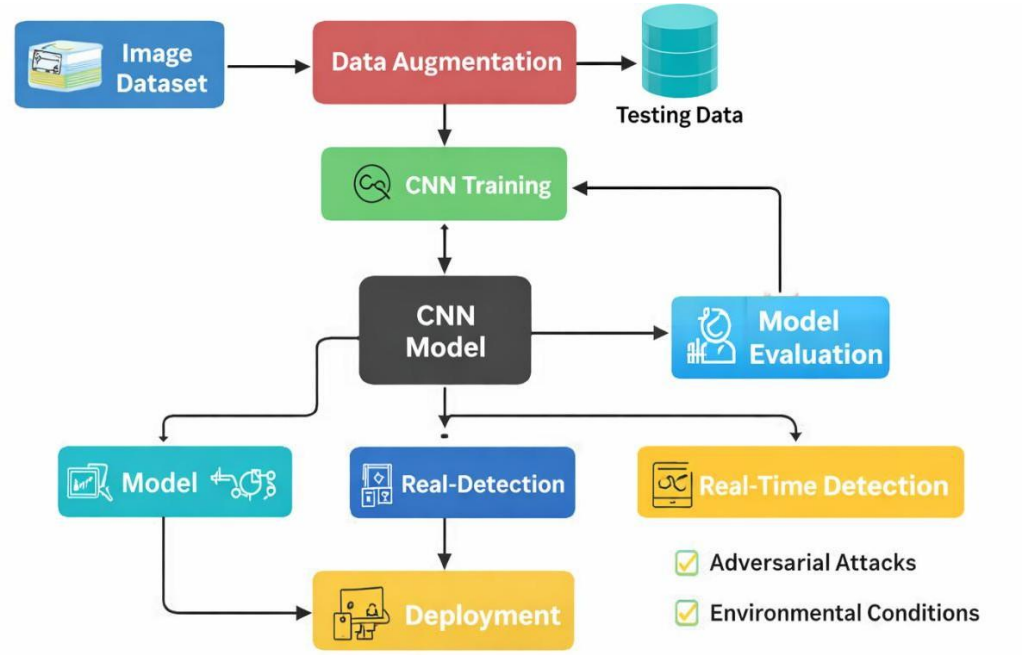


Fig. 1. Workflow of the proposed object detection model.

The Steps are given below:

Image dataset collection: In this step, you want to collect a large variety of image data units related to your proposed project. This will help ensure that the CNN learns the necessary capabilities of the images and is able to generalize correctly.

Data pre-processing & augmentation: Data pre-processing involves manipulation of raw data if you want to make it suitable for training. Data augmentation is performed through processes such as rotation, flotation, zoom-in/out, as well as adding noise.

Dataset partitioning: At this stage, the preprocessed dataset is divided into schooling, verification, and subset detection [13]. The training subset is used to train the CNN version, while the validation subset allows tuning model parameters and optimizing performance. Finally, a checking out subset is specified to evaluate the general accuracy and generalization performance of the trained model.

CNN Training: The school record set created in this step is input into the CNN model. This includes feature extraction using convolutional and pooling layers, as well as pattern recognition using fully connected layers. The CNN is then optimized using backpropagation and gradient descent techniques.

Model Building (CNN Model): The CNN learned in this phase will be the most final model after collecting the weights and structure of the model. This is the idea of version assessment, detection, and finally the deployment layer.

Model evaluation: In this step, the training type is evaluated with some unseen statistics, to check its accuracy, precision and reliability. This will ensure that the version is able to perform effectively during multiple events and can be used reliably for real-time detection.

Real-time detection: In this step, the CNN model is implemented for the stay enroll information containing video streams or sensor facts that enable instant understanding and classification of pix. This ensures that the system can operate under dynamic conditions and withstand all hostile attacks.

Deployment: In this remaining phase, the efficient and verified CNN version is applied to a production environment such as a cloud server, mobile application, or field device This step makes positive that the version will provide real-time predictability, scalability, and adaptability through continuous data training.

4. Methodology

Training, certification and testing kits are mounted. The intention of the work is to create a complete system based on deep mastering that can understand and be aware of a wide range of object types in an image and video frame and thus this system serves as a useful resource for automating or analyzing detection tasks. The gadget is developed to be generally transformed with use, entities deidentified from Kaggle plate form, so no moral or ethical review or approval is required.

4.1. Dataset Description and Ethics Primary: Our primary dataset consists of the publicly available Object Detection Dataset from Kaggle, which aggregates different object pictures from multiple sources. All the images in this dataset have been separated into standard directories for training, validation, and testing. The dataset is available on Kaggle[19] The dataset contains a total of 2,692 files, with images consisting of both JPEG and PNG formats. Each image consists of a label file with the bounding-box coordinates needed for object detection. This removed all image files that were either too corrupted or incomplete. The data was partitioned into train, validation, and test sets in an ordered manner. This method ensured that similar object categories were equally represented in all three sets.

4.2. Ethics and IRB: The model aimed to apply new methods to studies that used only publicly available, de-identified datasets. Institutional Review Board (IRB) approval and informed consent were not required, as per institutional policy and published guidance for analyses of de-identified public data. Datasets were used strictly in accordance with each license and terms of use. All datasets are public and de-identified, and used under their original licenses. No further attempts were made to re-identify individuals, and no protected or restricted health information was accessed.

4.3. Preprocessing: Every image was resized to a standard resolution of 640 × To guarantee a consistent input size, 640 pixels were normalized using the ImageNet mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]). To enhance, pixel values were normalized. model stabilization. To make the images clearer, noise reduction methods such as Gaussian smoothing were used to reduce distortions in the original dataset. Finally, the processed images were converted into a uniform RGB format along with their respective annotation files to ensure compatibility with deep learning frameworks.

4.4. Data Augmentation: Training augmentation included CutMix augmentation, where random patches were pasted between training images, random horizontal flips, rotation (±15°), brightness/contrast adjustment (±0.2), and occlusions, where part of some objects was kept hidden or blocked from the model view, forcing it to learn features from the entire objects rather than relying on a few parts.

5. Model architecture and Baselines

5.1. Feature Extractor: We have used ResNet-50 pretrained on ImageNet. The architecture features residual skip connections to preserve gradient flow. We extracted 2,048-dimensional features from the average pooling layer.

5.2. Classifiers: We have majorly focused on margin-based classifiers.

5.3. Faster R-CNN: We have utilized ResNet-50-FPN (Feature Pyramid Network) pretrained on the COCO dataset. This backbone uses side-by-side connections to build high-level neural network-based learning feature maps at all scales.

5.4. YOLO: We have utilized ResNet-50-FPN (Feature Pyramid Network) pretrained on the COCO dataset. This backbone uses side-by-side connections to build high-level feature maps at all scales.

5.5. Comparative Baselines: To validate the efficiency and accuracy of the deep learning model, we established two benchmarks.

5.6. Standard CNN: A user-defined built architecture trained end-to-end with (32 → 64 → 128 filters) which serves as a lower bound baseline to demonstrate the importance of complex feature [16].

5.7. Training Protocols: We have compared linear probing vs. “Full Fine-tuning” strategies. Our result shows that

fine-tuning the last 3 layers of the detection yielded a 14% improvement in mAP over the frozen mAp [17].

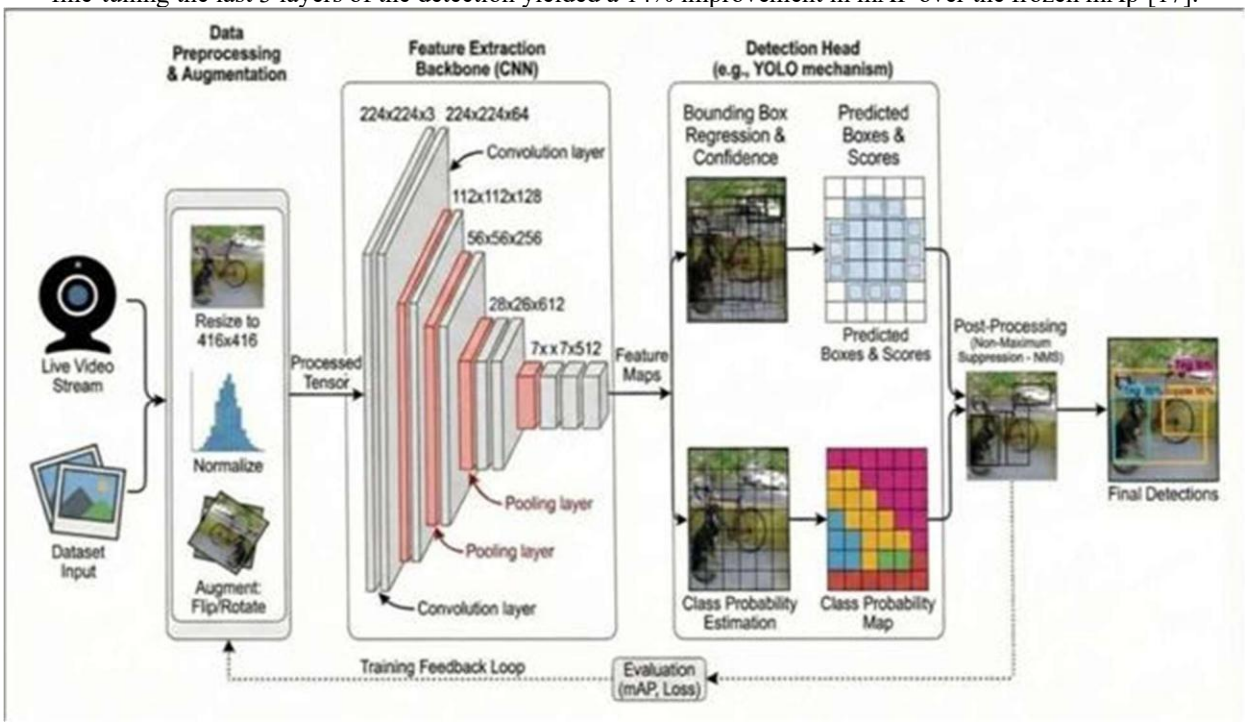


Fig. 2. Visual representation of model development.

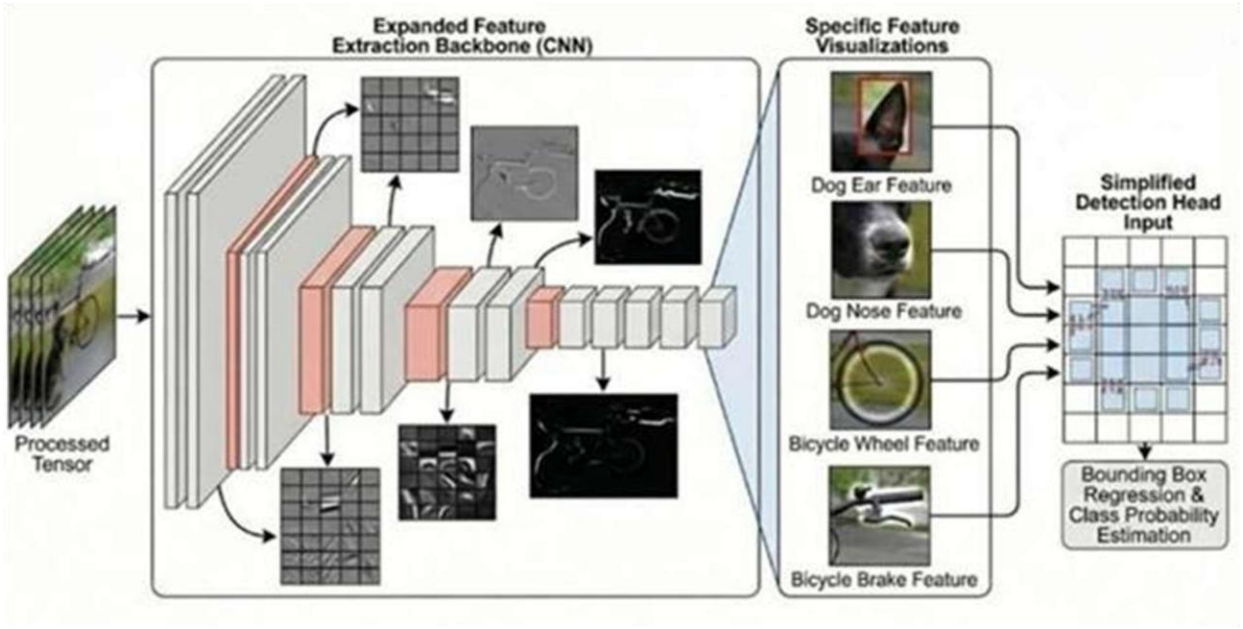


Fig. 3. Feature Extraction of objects.

The Fig. 2 shows the representation of model development, which includes different stages and processes from capturing data to final object detection [18]. The first step involves data input or capturing live data. After capturing, the image is pre-processed. Based on requirements, the image
is resized; we tried resizing it to 416 × 416 and 640 × 640 pixels. After resizing, the image is normalized to avoid

overfitting. The Fig. 3 shows the extended feature extraction process, which presents feature extraction step by step. The model aims at extracting deeper features of the object before predicting the actual object [19]. For example, if a picture of a dog is given to the model, the model focuses on features such as nose, ears, tail, and face. These features are captured by the model, which helps in detecting the object in the image using bounding boxes and successfully identifying it.

6. Results and Evaluation

To ascertain the inference latency and detection accuracy, the proposed model was evaluated under various environmental situations. When occlusion and illumination conditions changed, the model showed excellent stability. As shown in Table 1, the system achieved a Mean Average Precision (mAP) of 0.86-0.90 during the day and a smooth rate of 25-30 FPS. Dim and low light circumstances were used to test the model's durability. As shown in Table 1, there was a slight deterioration as the mAP dropped to about 0.83.To ensure that the targeted items are clearly visible, the impacts of catastrophic feature loss were mitigated using a combination of Auto Augment and CLAHE (Contrast Limited Adaptive Histogram Equalization). Real-time webcam testing was also conducted to verify the system's effectiveness. The model was determined to match the requirements of real-time surveillance applications by providing a maximum of 92 percent accuracy with little lag and operating at a steady pace of 30 frames per second.For Result analysis, we have used parameters such as:

Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Mean Average Precision (mAP): This is the standard metric for object detection.

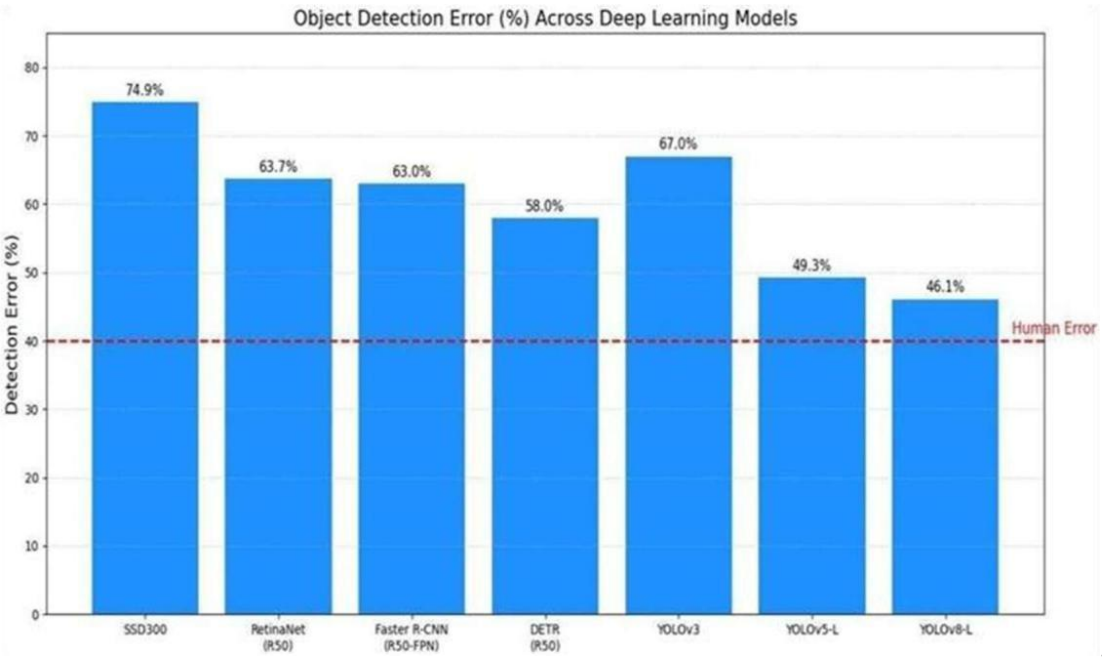
$$mAP = \frac{1}{N}\sum_{k=1}^{N} AP_i \tag{3}$$

The equation 1, 2 shows the formula and recall that was used during evaluation of model after training the model using several parameters. The equation 3 shows the value of average precision obtained.

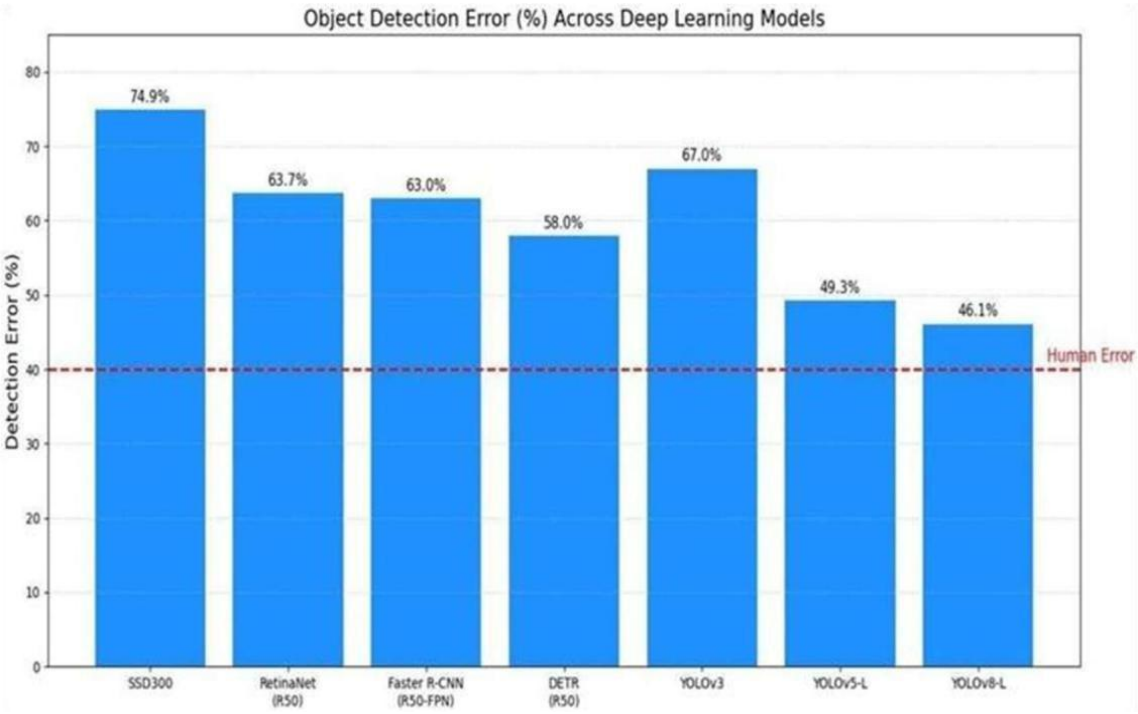
Table 1: Result Analysis

Condition	**Result**	**Notes**
Daylight (normal scene)	High detection accuracy (~86–90%)	Smooth real-time FPS (25–30)
Low light / Dim lighting	Slight drop in accuracy (83%)	CLAHE + AutoAug- ment help visibility
Occluded / Cluttered	Model still detects objects (~88%)	CutMix improves robustness
Motion / Noise	(~90%)	GAN-based augmentation maintains stable detection
Real-time webcam	Achieves real-time detection (~92% accuracy, 30 FPS)	Minimal lag

Based on the findings of the research, it can be concluded that the SSD300 model has an error rate of 74.9, as shown in Fig. 4. The RetinaNet (R50) and Faster R-CNN architectures have comparable error rates of approximately 63.7. Conversely, newer models such as YOLOv5-L and YOLOv8-L demonstrate improved performance, with error rates of 49.3 and 46.1, respectively. The performance of these recent models has significantly improved compared to SSD300, ResNet, and Faster R-CNN models. The same pattern is seen in the mAP display in PASCAL VOC and Cocoa datasets. Faster R-CNN reaches about seventy-five–80 mAP in the PASCAL VOC dataset; However, its performance decreases when evaluated on a more complex COCO dataset, as shown in Figure 5. Advanced YOLO fashions such as YOLOv4, YOLOv5-L, and YOLOv7 reveal a moderate evolution of 10–15 mAP factors over the next COCO dataset



(a)



(b)

Fig. 4. Object detection error

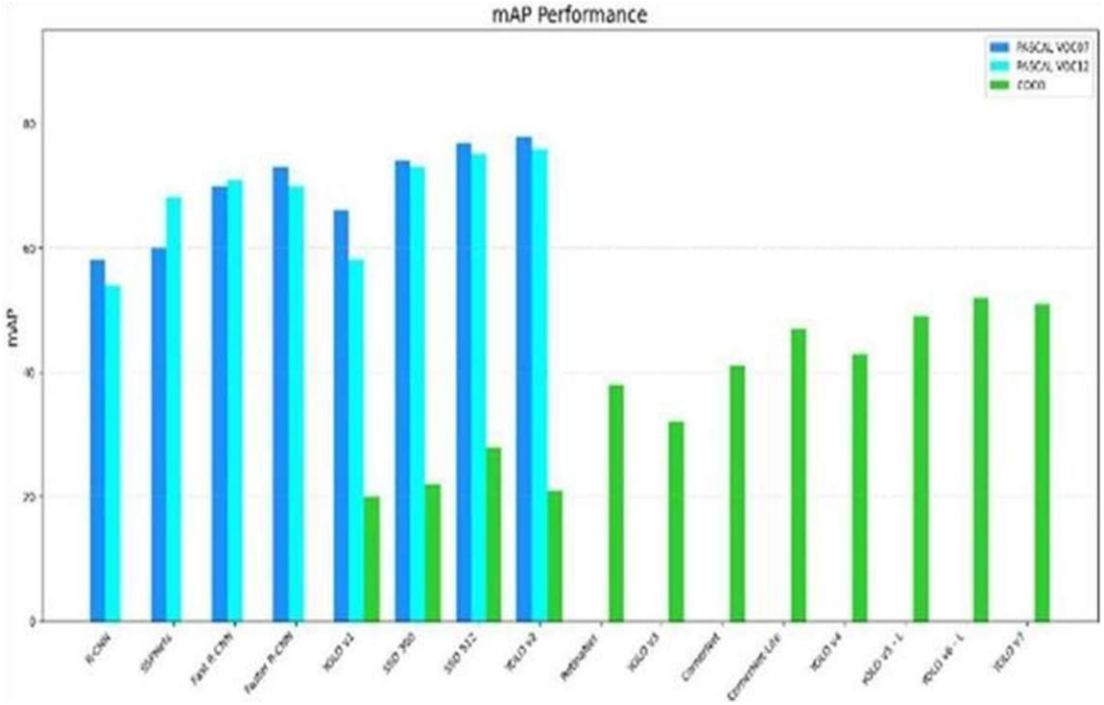


Fig. 5. mAp performance (Proposed by the authors)

7. Conclusion and Future Scope

Comparative evaluation shows that current models based on YOLO architecture (YOLOv5 and YOLOv8) show much better results compared to conventional CNN-mainly based and conventional detection algorithms such as SSD300 and Faster R-CNN, including CutMix, AutoAugment, and GAN-based challenges. The proposed approach succeeded in producing consistent results under various conditions (such as daylight, low-light, occlusion, and moving objects) with an accuracy of 83-92% and 25-30 FPS runtime performance. The above results illustrate that the model is capable of learning under different conditions and applying that knowledge to more complex tasks involving real-world applications. On the other hand, the lack of validation of public data sets used during training might introduce possible biases, while the narrow set of images from one particular environment does not allow for generalization. Future endeavors will involve diversification of datasets according to regions and climatic conditions, inclusion of IoT as well as satellite data, and the implementation of cutting-edge technologies including Vision Transformers, federated learning, and YOLO-based transformers. The testing in real-life scenarios and applications in the defense/ security fields are long-term objectives.

Ethical Declarations

Acknowledgments: The authors would like to acknowledge the support provided by Galgotias College of Engineering and Technology for providing the necessary resources and infrastructure for conducting this research.

Conflict of Interest: The authors declare that there is no conflict of interest regarding the publication of this research paper.

References

[1] Benali Amjoud and M. Amrouch, “Object detection using deep learning, CNNs and vision transformers: A review,” *IEEE Access*, vol. 11, pp. 35479–35516, 2023. https://doi.org/10.1109/ACCESS.2023.3266093

[2] H. Pratiwi, A. P. Windarto, S. Susliansyah, R. R. Aria, S. Susilowati, L. K. Rahayu, Y. Fitriani, A. Merdekawati, and I. R. Rahadjeng, “Sigmoid activation function in selecting the best model of artificial neural networks,” in *Journal of Physics: Conference Series*, vol. 1471, no. 1, p. 012010, 2020. https://doi.org/10.1088/1742-6596/1471/1/012010

[3] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,

2019, pp. 113–123. https://doi.org/10.1109/CVPR.2019.00020

[4] J. Deng, X. Xuan, W. Wang, Z. Li, H. Yao, and Z. Wang, "A review of research on object detection based on deep learning," in *Journal of Physics: Conference Series*, vol. 1684, no. 1, p. 012028, 2020. https://doi.org/10.1088/1742-6596/1684/1/012028

[5] S. Devi, K. Thopalli, R. Dayana, P. Malarvezhi, and J. J. Thiagarajan, "Improving object detectors by exploiting bounding boxes for augmentation design," *IEEE Access*, vol. 11, pp. 108356–108364, 2023. https://doi.org/10.1109/ACCESS.2023.3320638

[6] H. Fang, B. Han, S. Zhang, S. Zhou, C. Hu, and W.-M. Ye, "Data augmentation for object detection via controllable diffusion models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 1257–1266. https://doi.org/10.1109/WACV57701.2024.00129

[7] I.-A. Herdea, D. Tiwari, J. Oyekan, and A. Tiwari, "Data augmentation framework for improved classification in object detectors," *IEEE Access*, vol. 13, pp. 28476–28491, 2025. https://doi.org/10.1109/ACCESS.2025.3539455

[8] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020. https://doi.org/10.1007/s11263-019-01247-4

[9] Patricio, A. Dehban, and R. Ventura, "FLORA: Efficient synthetic data generation for object detection in low-data regimes via finetuning Flux LoRA," *arXiv preprint arXiv:2501.07474*, 2025. https://doi.org/10.48550/arXiv.2501.07474

[10] S. Poonkuntran, R. K. Dhanraj, and B. Balusamy, Eds., *Object Detection with Deep Learning Models: Principles and Applications*. Boca Raton, FL, USA: CRC Press, 2022. https://doi.org/10.1201/9781003206736

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016. https://doi.org/10.1109/TPAMI.2016.2577031

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[13] F. G. Dos Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–25, 2022. https://doi.org/10.1145/3510413

[14] M. Tamilselvi, S. S. S. Kalaivani, V. Sunderasan, K. Sailaja, D. Gopal, and R. Karthick, "Deep learning for object detection and identification," in *Hybrid and Advanced Technologies*, CRC Press, 2025, pp. 218–223. https://doi.org/10.1201/9781003559139-29

[15] K. Vaishnavi, G. P. Reddy, T. B. Reddy, N. Iyengar, and S. Shaik, "Real-time object detection using deep learning," *Journal of Advances in Mathematics and Computer Science*, vol. 38, no. 8, pp. 24–32, 2023. https://doi.org/10.9734/jamcs/2023/v38i81787

[16] X. Wang and W. Jia, "Optimizing edge AI: A comprehensive survey on data, model, and system strategies," *IEEE Transactions on Knowledge and Data Engineering*, 2025. https://doi.org/10.1109/TKDE.2025.3622600

[17] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6023–6032. https://doi.org/10.1109/ICCV.2019.00612

[18] P. Kaur, B. S. Khehra, and E. B. S. Mavi, "Data augmentation for object detection: A review," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2021, pp. 537–543. https://doi.org/10.1109/MWSCAS47672.2021.9531849

[19] Dataset: https://www.kaggle.com/datasets/mohamedgobara/26-class-object-detection-dataset