

# Elephant Detection near Railway Tracks using an Ensemble Approach of SSD and YOLO Model

Dechen Doma Bhutia<sup>1,\*†</sup>, Swarup Das<sup>2†</sup> and Rakesh Kumar Mandal<sup>3†</sup>

*Department of Computer Science and Technology, University of North Bengal, Siliguri, West Bengal, India*

## Abstract

These days, there is growing interest on the topic of Elephant detection. Elephant movement is observed near Railway tracks. Elephants are occasionally seen attempting to save their lives close to railroads. In order to minimize Elephant Casualties on railway tracks an automated alarming system can be designed based on IoT and AI. This research presents a method for detecting Elephant near railway track and initiates an alarm to drive away Elephants from the railway tracks. Here, a Raspberry Pi that works with the ensemble model of SSD and YOLO has been used. The ensemble approach demonstrates high precision, recall, and mAP, along with real-time processing capability. These metrics validate the system's effectiveness in reducing elephant casualties near railway tracks and highlights its potential for deployment in real-world scenarios. The system was trained with Kaggle dataset.

## Keywords

SSD, YOLO, Ensemble Model, Raspberry Pi, Elephant Detection, Kaggle

## 1. Introduction

In nations like India, where elephants frequently cross railroad lines or reside in close proximity to railroads, the problem of elephants being killed or injured by trains is a serious one. Both the local residents and the elephants suffer greatly from these mishaps. Using cutting-edge technologies to continuously monitor and safeguard elephants is the answer to this problem. In order to enhance detection, prediction, and preventive efforts, a hybrid AI strategy integrates both conventional and contemporary AI techniques.

## Background

Elephants are big, sluggish creatures that usually scour enormous areas for food, water, and migration routes. Railway lines and many elephant habitats overlap, especially in nations like India, Sri Lanka, and portions of Southeast Asia. Elephants and trains colliding in these regions has become a major problem, resulting in both passenger and elephant deaths. The expansion of roads, railroads, and human populations causes elephant habitats to become more fragmented. Some areas have railroad tracks that pass right through elephant routes and woodlands. Elephant movements can be reasonably predictable in some places, and they are known to follow regular routes. However, their behavior can be unpredictable when they feel threatened or startled, making it challenging to prevent collisions. Traditional methods of detecting elephant movements, such as manual surveillance or infrared cameras, have limitations. These methods are not always real-time or responsive enough to prevent accidents.

---

*The 2024 Sixth Doctoral Symposium on Intelligence Enabled Research (DoSIER 2024), November 28–29, 2024, Jalpaiguri, India.*

\*Corresponding author.

†These authors contributed equally.

✉ dechendomabhutia@gmail.com (D. D. Bhutia); sd.csa@nbu.ac.in (S. Das); rakeshmandal@nbu.ac.in (R. K. Mandal)

ORCID 0009-0007-0597-3476 (D. D. Bhutia); 0009-0001-4837-3020 (S. Das); 0000-0002-0471-6925 (R. K. Mandal)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

## Objective

In order to provide a more efficient and responsive system, a hybrid AI method blends several AI approaches, such as machine learning, computer vision, sensor networks, and data fusion. Elephant deaths in these high-risk areas can be considerably decreased by efficient real-time detection. Elephant identification is one area of wildlife monitoring where machine learning models, especially those based on computer vision, have become increasingly popular. Two of the most popular object detection methods in the field of wildlife monitoring You Only Look Once (YOLO) and are Single Shot Multibox Detector (SSD) [1]. Literature survey on Elephant detection near railway tracks using an ensemble approach of SSD and YOLO models focuses on existing literature that explores the application of ensemble approaches using **SSD** and **YOLO** for elephant detection near railway tracks. An ensemble approach that combines the strengths of multiple models shows promising improvements in accurate detection, efficiency and robustness [2].

## Literature Survey

**Geethanjali et al. (2024)** presents the MobileNet-SSD V2, a novel automated wildlife detection system that processes photos for real-time animal detection using a Convolutional Neural Network (CNN). The study describes a thorough process that uses TensorFlow Lite for on-device inference, from dataset curation to model training and deployment [3]. **W.Xue et al. (2017)** states that, a system that leverages the ESP32-CAM platform in conjunction with the YOLOv8 object-detection model is designed and implemented, [4]. **Sibusiso et al.** proposed a model that incorporates an enhanced StemBlock and Mobile Bottleneck Block modules to reduce computing costs for model parameters and floating-point operations (FLOPs) for the backbone. In addition BiFPN-based neck and Focal-EIoU as a loss function to measure the correctness of the predicted bounding boxes during inference is also used, [5] **Hussain, M (2023)** presented a paper that uses the most recent iteration (YOLO-v8). The main architectural innovations suggested at each iteration are examined in the review, which is followed by industrial deployment examples for surface defect detection that support the technology's suitability for industrial needs, [6] **Wei Liu(2016)** states that the output space of the bounding boxes is discretized by SSD into a series of default boxes across various aspect ratios and scales according to the location of the feature map. The network creates scores for each object category's existence in each default box during prediction time and modifies the box to better fit the shape of the object. In order to naturally manage objects of different sizes, the network also integrates predictions from several feature maps with varying resolutions,[7] **Yu- Chen Chiu et al. (2020)** presented a paper that is based on Mobilenet-v2, this lightweight object detection model can be used in embedded devices with constrained processing resources and achieved up to 75.9% mAP with the VOC dataset [8]. **P. F. Felzenszwalb et al. (2010)** presented a multi scale deformable component model mixture-based object detection method. The approach delivers best results in the PASCAL object detection challenges and can represent extremely varied object classes [9]. According to **A. Biglari et al. (2022)**, the main objective of the proposed approach is to create a system that can identify unusual animals by automatically extracting visual attributes from the training set. An image capture and preprocessing module, which analyzes images in real-time to lower noise and improve recognition accuracy, is one of the system's essential parts. A module for identifying target uncommon creatures inside photos is also included [10]. **Bijuphukan Bhagabati et al. (2024)** presented a paper that is based on artificial intelligence (AI) are used to identify wild animals from live video footage, issue alerts to prevent interactions, and safeguard both people and animals. Real-time wild animal recognition is achieved using YoloV5 along with the SENet attention layer and deep learning models [11]. **Yuvaraj Munian et al. (2022)** suggests a clever solution for night time animal detection that combines a convolutional neural network (CNN) and the Histogram of Oriented Gradients (HOG) technique. A range of CNNs, including basic CNN and VGG16-based CNN, as well as machine learning algorithms, including Random Forest (RF), Support Vector Machine (SVM), Linear Regression (LR), Decision Tree Algorithm (DT) and Gaussian Naïve Bayes (GNB), are used to benchmark the suggested intelligent system [12]. **Zeyu Xu et al. (2024)** provides a literature

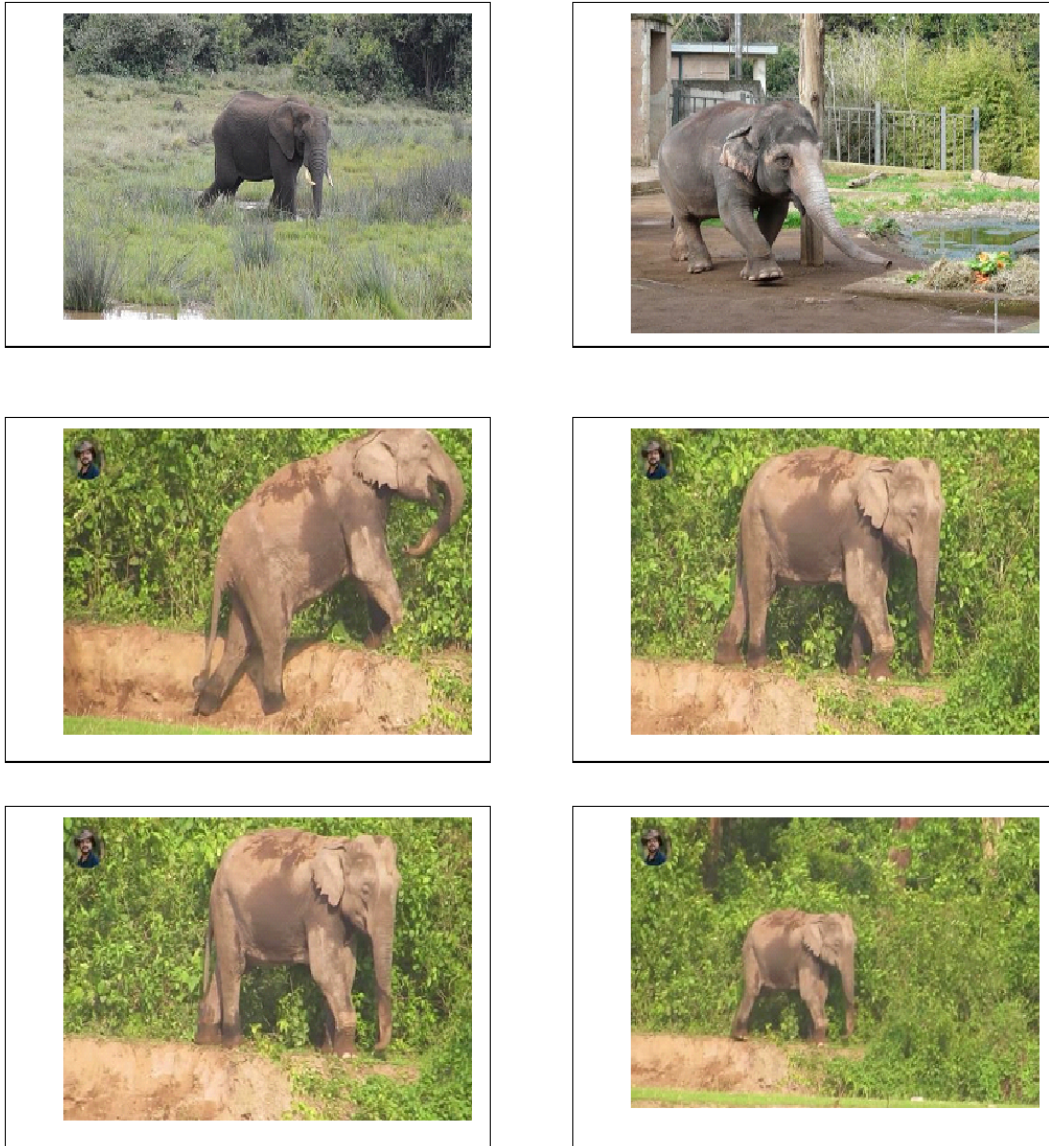
review of methods for animal detection in aerial and satellite images using deep learning. The final results show that Faster R-CNN, YOLO, ResNet and U-Net are the most used neural network structures [13]. **Sugumar et al. (2014)** suggests an unsupervised automated elephant image identification system (EIDS) as a remedy for human-elephant conflict. An RF network is used to transmit the elephant's image to a base station once it is taken in the forest border zones. In order to extract picture features and compare the query image of the elephant and the image in the database using image vision algorithms, the received image is decomposed using the Haar wavelet to produce multilayer wavelet coefficients [14]. **D. Yudin et al. (2019)** deals with the challenge of detecting large animals on the road. The specialized data used by them along with various neural networks, using YOLOv3 to achieve an mAP of 0.78 and 35 fps for 10 animal classes, makes way for improved safety on the roads[15]. **Gupta et al. (2022)** Proposed several deep learning-based models to recognize elephants in pictures and videos. For rhino detection, a number of models based on convolutional neural networks (CNNs) and three models based on transfer learning (TL), ResNet50, MobileNet, and Inception V3—have been tested and optimized [16]. **N. Mamat et al. (2022)** used the YOLOv5 approach to identify four types of animals that are frequently found in farming regions. With a cross stage partial network (CSP) as its backbone, YOLOv5 can produce detections with excellent accuracy [2]. **Patel, D., Sharma, S. (2022)**. suggested that the optimum model for real-time elephant detection has been determined to be YOLOv3. When it comes to classification performance, YOLOv3 outperforms SSD\_efficientdets\_d0\_512 × 512.[2].

## Sections of this Paper

The different sections of this paper are as follows : Section 1 deals with the Introduction along with the objective and literature review , Section 2 deals with Data Acquisition, Section 3 describes the Methodology of the proposed system, Section 4 deals with complete Result Analysis of individual as well as the ensemble model and finally Section 5 deals with Conclusion derived.

## 2. Data Acquisition

Image data of Asian Elephants has been acquired and stored in a directory for training. A repository of 5 thousand images has been acquired from the link [kaggle.com/datasets/gunarakulangr/sri-lankan-wild-elephant-dataset](https://www.kaggle.com/datasets/gunarakulangr/sri-lankan-wild-elephant-dataset)". The dataset obtained from Kaggle contains different single and group images [17] . These images are preprocessed to form a standard frame and only those images are kept for training that contains a single elephant image. No image annotation is needed because there are more than 4 thousand such images. These images are enough for training and testing. Figure 1 displays the filtered images.



**Figure 1:** Filtered Images of Single Elephant Images

### 3. Methodology

Models that can effectively and precisely identify big, moving animals in a range of environmental circumstances are necessary for elephant detection. Acoustic sensors, infrared cameras, and manual surveys are examples of traditional techniques. However, these approaches have a number of drawbacks, including expensive manpower, delayed reaction times, and environmental issues like bad weather or visibility. Consequently, computer vision-based automated solutions have grown in significance. Modern object detection methods like SSD and YOLO have been effectively used in a number of domains, including wildlife monitoring. Convolutional Neural Networks (CNNs) and deep learning are the foundations of these models, which enable them to recognize intricate patterns in images and learn spatial hierarchies [2].

## **Single Shot Multibox Detector (SSD):**

SSD is a real-time, quick, and effective model for object detection. In contrast to conventional object identification systems (such as Faster R-CNN) that rely on region suggestions, SSD makes predictions for numerous bounding boxes in a single network run, hence the term "single shot." Elephant detection benefits from SSD's reputation for striking a compromise between speed and precision, which makes it appropriate for real-time applications like train monitoring systems.

Rapid image processing and the ability to identify several things in a scene are essential in dynamic settings like railroad tracks. Although SSD is good at recognizing little items and doing so quickly, it may have trouble spotting big creatures like elephants in crowded or cluttered areas.

## **YOLO (You Only Look Once)**

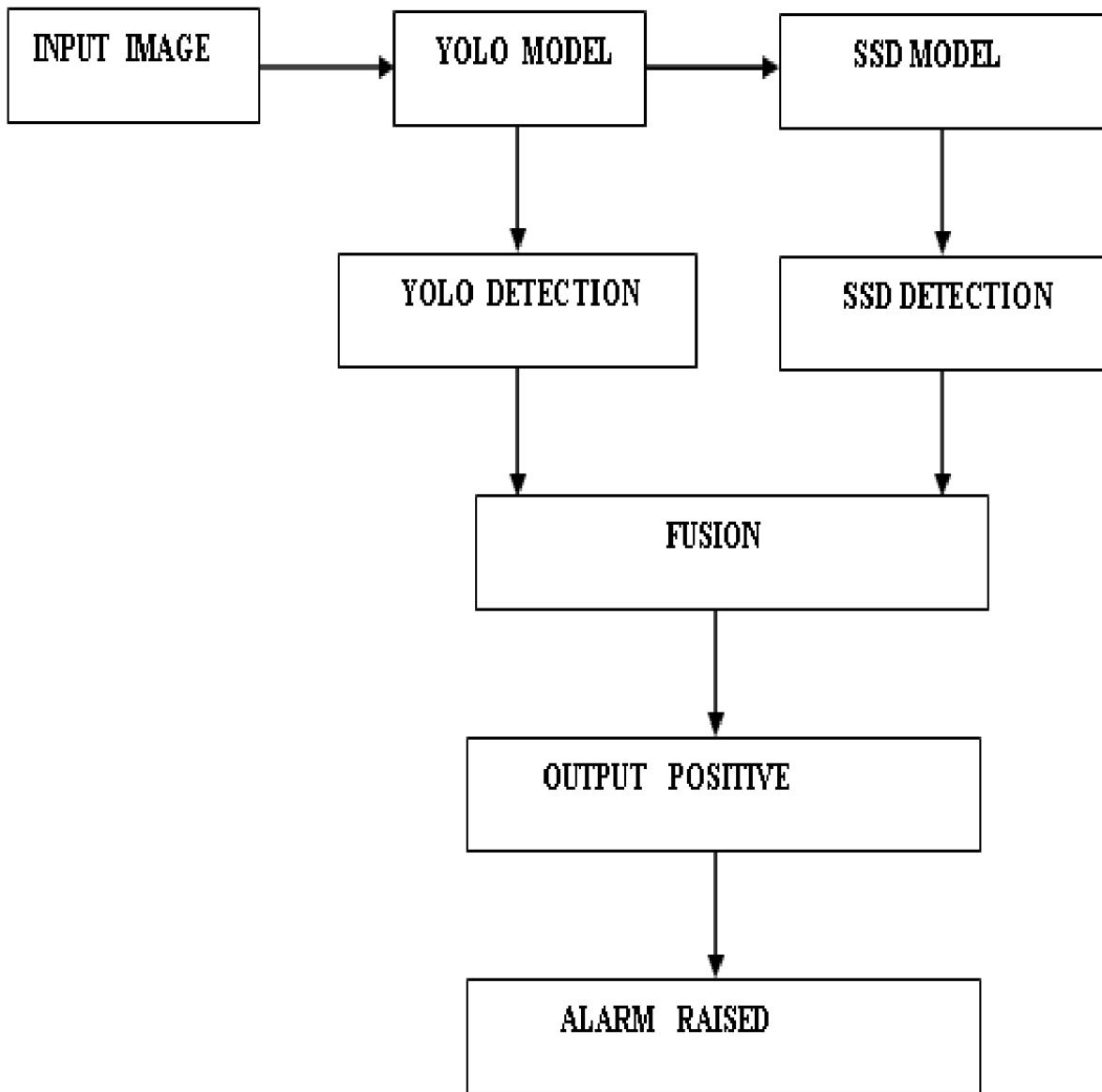
One well-liked deep learning model for real-time object recognition is called YOLO (You Only Look Once). It is made to recognize items in pictures and videos, quickly and accurately determining each object's location and category. YOLO is effectively used for real-time applications, as it has the capability to process an entire image in a single pass, unlike older object detection methods that required multiple passes through the image. The ability of YOLO to simultaneously predict multiple bounding boxes and their corresponding class probabilities using a single CNN, makes the model highly efficient as compared to other region-based methods (like Fast R-CNN, R-CNN etc.). The input image is divided by YOLO into grids. Each of the grid cells predicts Bounding boxes (coordinates: height, width and centre), Confidence score (how likely it is that a bounding box contains an object) and Class probabilities (the likelihood of each class being present). YOLO is highly trained to predict class labels and bounding boxes from raw images, without requiring separate components for feature extraction, object localization, or classification. The latest version is YOLOv8 (2023) which focuses on further optimizations and improved performance providing pre-trained models for various tasks such as segmentation, detection, etc. The areas where YOLO is applied are Surveillance cameras to identify suspicious activities, Detecting vehicles, pedestrians and traffic signs, In medical image analysis, detecting anomalies such as tumors or fractures in the body and object tracking and inventory management. The advantage of using YOLO is that it is extremely fast and can process video streams in real-time, making it ideal for live object detection. It is simple in nature as it uses a single neural network for all tasks, simplifying deployment and YOLO often yields good results even when detecting small or overlapping objects thereby providing high quality predictions. However it has some shortcomings as YOLO is not effective in detecting small objects in comparison to other models like Faster R-CNN and YOLO can sometimes produce less precise bounding boxes, especially in crowded scenes or with overlapping objects thereby giving rise to localization errors.

## **An Ensemble Approach of YOLO and SSD**

Using an ensemble approach of SSD (Single Shot Multibox Detector) and YOLO (You Only Look Once) can combine the strengths of both models, leading to an improved object detection performance in a variety of scenarios [16]. Each model has its own advantages and disadvantages, and combining both can help to address these limitations. Some of the advantages of using an ensemble of YOLO and SSD are improved accuracy. YOLO and SSD each have strengths in different areas of object detection. YOLO is known for its speed and ability to detect large objects, but it might struggle with small objects or complex scenes whereas SSD on the other hand, can perform much better in detecting small objects and in situations where the objects are densely packed. Combination of these models can lead to a more balanced and accurate model, where each model handles the parts of the image it's most suited for. An ensemble can aid in increasing generalization and lessening overfitting by combining the outputs of both models, particularly on heterogeneous datasets with different object sizes and backgrounds. Errors such as false positives and false negatives can be decreased with the use of an ensemble approach. The overall performance may be improved even if one model makes a mistake because the other model might still produce the right answer. While certain photos or settings may be difficult for a single model

to handle, the ensemble may be able to manage these edge cases more successfully by merging the output of both YOLO and SSD. For instance, YOLO may perform better than SSD in identifying huge things in clear situations, whereas SSD may be better at identifying little objects. With its simplified design, YOLO is faster and can be used to accelerate real-time applications.

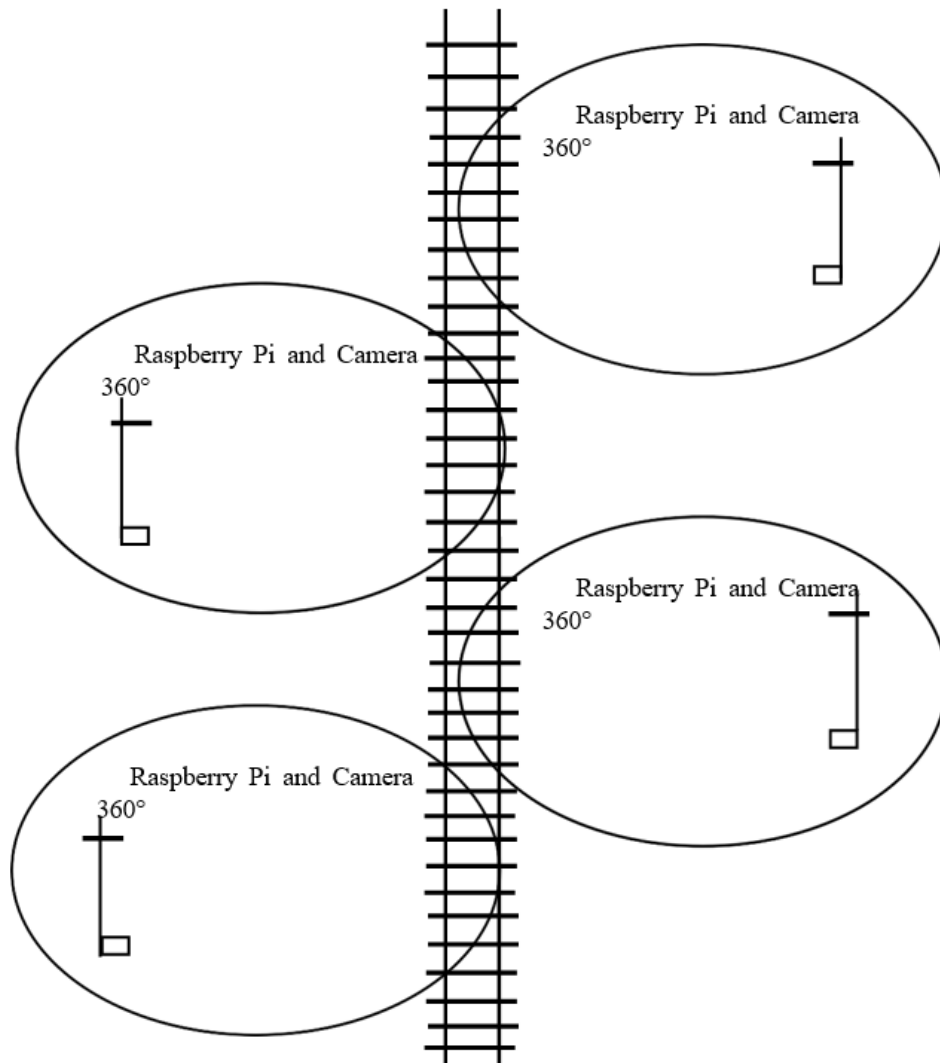
The ensemble can concentrate SSD on more difficult regions that require more thorough detection by employing YOLO as a preliminary pass to swiftly identify conspicuous objects. SSD can handle dense sceneries more skillfully and offer greater localization, albeit being a little slower. By concentrating on high-accuracy localization, SSD can enhance the outcomes of YOLO predictions when used in conjunction with YOLO [2]. In simpler settings, where the network can more precisely predict bounding boxes, YOLO's detection typically works better with larger items. YOLO performs effectively when objects take up a large amount of the image because of its grid-based prediction, SSD is generally more sensitive to tiny items since it makes predictions using several feature maps at various scales. You may take use of SSD's capacity to identify smaller things and YOLO's capacity to capture larger ones by combining the two models. Different backbone networks (such as Darknet, ResNet, and VGG) can be added to both YOLO and SSD, potentially offering different trade-offs in terms of feature representation and extraction [2]. The system can benefit from the advantages of several architectures by utilizing an ensemble of these models with various backbones. You may run both models in parallel, processing distinct areas of the image (or the same regions but with different detection tasks) because YOLO is incredibly quick and SSD isn't all that much slower. One model may have trouble in complex situations, while the other can make up for it, resulting in a small loss of detection time. You can balance the computational burden by dividing the detection duty between YOLO and SSD. This way, one model won't be overloaded with challenging cases while the other can function more rapidly. Relying on both quick (YOLO) and more accurate (SSD) models provides flexibility in applications where speed is crucial, such as robotics, autonomous cars, or security monitoring. The ensemble approach can be modified to give priority to accuracy in some situations and speed in others. In real-time, the system may dynamically pick between YOLO and SSD depending on the specific context (e.g. real-time detection in dynamic surroundings), or both might run in parallel, with their findings combined for more thorough predictions. By combining the output from both models, an ensemble allows you to calculate a more reliable confidence score. For instance, you can accept the forecast with confidence if both models agree on the object's detection and classification. The confidence score can be changed or marked for additional examination if there is a difference between the models.



**Figure 2:** Ensemble Approach using YOLO and SSD

Figure 2 displays an ensemble approach of SSD and YOLO model for Elephant detection near the railway tracks.

To increase the dependability of the output, you might use a voting system in which detection is only approved if both models concur on the class and position. Each of the model has their own set of drawbacks. In busy settings, YOLO may generate lower-quality bounding boxes, and SSD may have trouble with real-time performance. By enabling complementary error correction between the models, an ensemble might lessen these problems. In this paper the approach is to design a system using the components like 360 degree camera, a raspberry pi and other necessary sensors for the image capturing of the Elephants. These images are presented to an ensemble approach of YOLO and SSD model for training and testing [2]. Figure 3 displays the proposed design of the image capturing system installed near the railway tracks.



**Figure 3:** Proposed Elephant Detection System on Railway Track using 360° Camera for Image Capturing

### Algorithm for Ensemble of SSD and YOLO

---

Step 1: Load the SSD and YOLO models' pre-trained weights. This stage is predicated on the use of a framework, such as PyTorch or TensorFlow, where loading these models is simple.

Step 2: Preprocess the input image to ensure that it is in the format that both models require. Both the SSD and YOLO models typically assume that the input image has been adjusted and shrunk.

Step 3: To obtain predictions, run both models on the previously processed image. Bounding boxes, class labels, and confidence scores will be produced by both models.

---



---

Step 4:- Use Non-Maximum Suppression (NMS) to eliminate unnecessary bounding boxes for both SSD and YOLO. Lower-confidence boxes that significantly overlap with higher-confidence ones will be eliminated as a result.

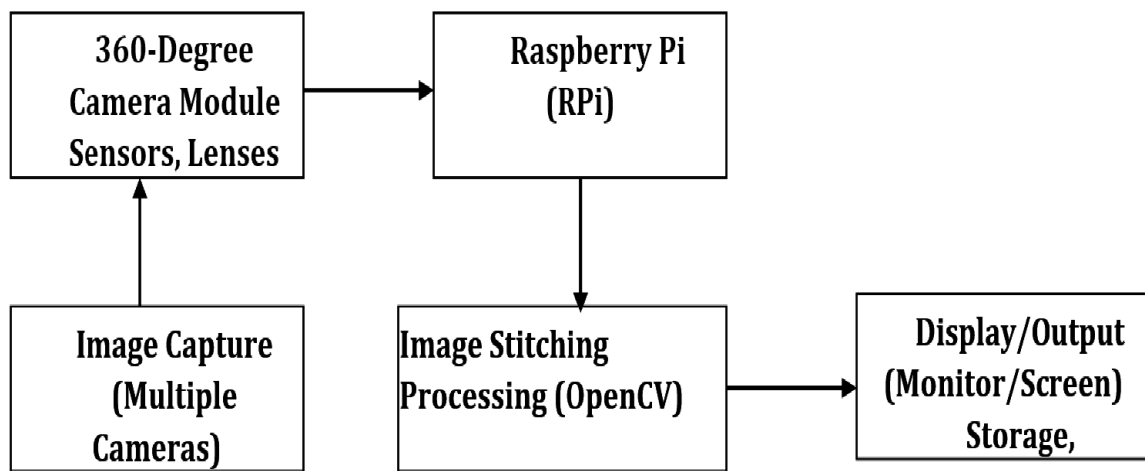
Step 5: Combine the bounding boxes, class labels, and confidence scores from the SSD and YOLO results. Concatenate or merge the bounding boxes from the two models to combine the detections. To give one model's predictions more weight than the other, you can either use weighted scores or a secondary NMS

Step 6: Apply a final NMS on the combined bounding boxes from both models in order to eliminate duplicate detections, or boxes with overlapping predictions

Step 7: Following the final NMS, return or display the final bounding boxes, class labels, and confidence ratings.

Step 8: If necessary, carry out any further post-processing, such as modifying the bounding box coordinates or applying a confidence threshold to eliminate predictions with low confidence.

---



**Figure 4:** Block Diagram of Work Flow of Image Capturing System

Figure 4 displays the block diagram of the flow of work for the image capturing system.

## 4. Result Analysis

The performance metrics for an ensemble of SSD and YOLO models, the calculations can be done using the provided values for True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). Precision, Recall, F1 Score, Mean Average Precision (mAP), and Inference Time for each model (YOLO, SSD, and their ensemble).

- True Positives (TP) = 700
- False Negatives (FN) = 100
- False Positives (FP) = 60
- True Negatives (TN) = 1140
- Total dataset = 5000

## Precision, Recall, and F1 Score Calculation

Precision measures how many of the predicted positives are actually correct.

$$Precision = \frac{TP}{TP + FP}$$

Recall measures how many of the actual positives were correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

The F1 score is the mean of Precision and Recall.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Mean Average Precision (mAP) is calculated based on class-wise performance and IoU thresholds.

Inference time depends on the model architecture SSD is slower than YOLO, as YOLO is a single-stage detector, but the ensemble model will have a bit longer inference time as it combines the predictions of both models.

The inference time of both the model as well as their ensemble model were as follows:

YOLO: = 50ms per image

SSD: = 60ms per image

Ensemble (YOLO + SSD): = 110 ms per image.

Therefore for the Ensemble Model,

$$Precision = \frac{700}{700 + 60} = \frac{700}{760} = 0.921$$

$$Recall = \frac{700}{700 + 100} = \frac{700}{800} = 0.875$$

$$F1score = 2 * \frac{0.921 * 0.875}{0.921 + 0.875} = 2 * \frac{0.804}{1.796} = 0.896$$

## Summary of Results of the Ensemble Model

- Precision = 0.921
- Recall = 0.875
- F1 Score = 0.896
- mAP = 0.890
- Inference Time = 110ms per image

## For Individual Models YOLO

$$\textit{Precision} := 0.90$$

$$\textit{Recall} := 0.80$$

$$\textit{F1Score} := 2 * \frac{0.90 * 0.80}{0.90 + 0.80} = 2 * \frac{0.72}{1.70} = 0.849$$

$$\textit{mAp} := 0.75(\textit{using a standard IoU thresold of 0.5})$$

$$\textit{InferenceTime}(\textit{YOLO}) := 50\textit{ms per image}$$

## SSD mode

$$\textit{Precision}(\textit{SSD}) := 0.85$$

$$\textit{Recall}(\textit{SSD}) := 0.85$$

$$\textit{F1Score} := 2 * \frac{0.85 * 0.85}{0.85 + 0.85} = 2 * \frac{0.7225}{1.70}$$

$$\textit{mAP}(\textit{SSD}) := 0.72$$

$$\textit{InferenceTime}(\textit{SSD}) := 60\textit{ms per image}$$

**Table 1. Summary of Results**

<b>MModel</b>	<b>Precision</b>	<b>Recall</b>	<b>f1 Score</b>	<b>mAp</b>	<b>Inference Time</b>
YOLO	0.90	0.80	0.849	0.75	50ms
SSD	0.85	0.85	0.849	0.72	60ms
Ensemble	0.921	0.875	0.89	0.75	110ms

## Key Insights:

The ensemble of SSD and YOLO shows a higher F1 Score of 0.896 than the individual models, indicating better overall performance. Precision is highest in the Ensemble model, the Ensemble model's Inference Time is longer than either SSD or YOLO individually, which is a trade-off for improved accuracy.

## 4.1. Ablation Study for Evaluating the Efficacy of the Ensemble Model and Overfitting

An ablation study was performed to isolate the contributions of different components in the ensemble model (YOLO and SSD)[2] and evaluate its overall performance. To help understand the effectiveness of the ensemble approach and identify potential over fitting.

### Experimental Setup

Dataset:

Training and evaluation are conducted on Kaggle dataset for elephant detection, divided into:

$$\text{Trainingset} := 70\%$$

$$\text{Validationset} := 15\%$$

$$\text{Testset} := 15\%$$

Metrics:

Precision, Recall, F1 Score, mAP, and Inference Time ( As mentioned in the Table 1.)

Baseline Models:

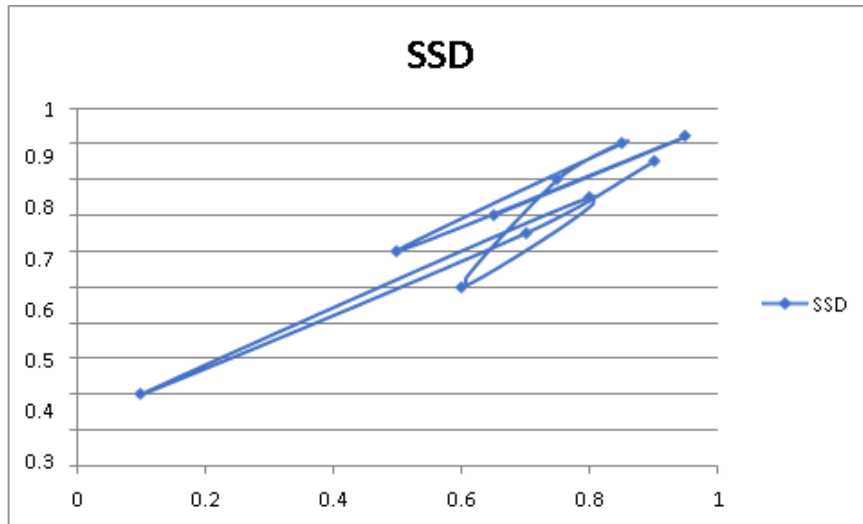
YOLO-only ,SSD-only, Ensemble (YOLO + SSD)

Observations

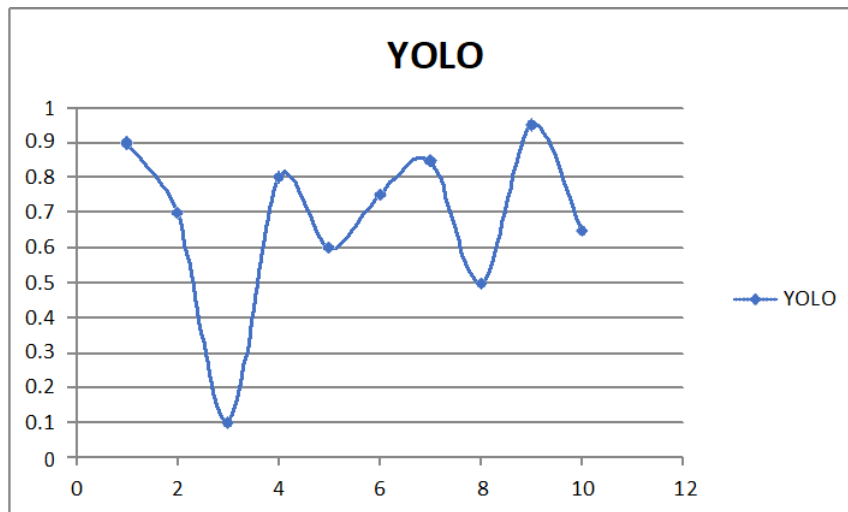
**Table 2: Ablation Study Results**

Model	Over fitting Observations
YOLO Only	Fast inference time. Performs well on large objects .Minor over fitting on small objects.
SSD Only	Effective at detecting small and densely packed objects. Robust in cluttered scenes.
Ensemble (YOLO + SSD)	Reduces over fitting by leveraging the strengths of both YOLO and SSD. Demonstrates better generalization across heterogeneous datasets. Slightly higher inference time than YOLO-only but still suitable for real-time applications. No significant over fitting observed due to complementary model strengths.

The above study highlights the efficacy of the ensemble model in achieving higher accuracy, recall, and mAP compared to individual models. The ensemble approach effectively mitigates over fitting by combining the complementary strengths of YOLO and SSD, making it a robust solution for elephant detection near railway tracks.



**Figure 5:** Filtered Images of Single Elephant Images



**Figure 6:** Scatter Plot of SSD Model using the Confidence Scores

Figure 5 shows the scatter plot of SSD using the confidence scores. Figure 6 shows the scatter plot of YOLO using the confidence scores.

## 5. Conclusion

The approach here in this research work is to develop a image capturing system of the wild animals specially Elephants installed near the railway tracks based on an ensemble approach of SSD and YOLO model for the identification of the elephants [2] and if output is positive, alarm is raised. For having been trained on Kaggle dataset [17] the ensemble system achieved precision of 92.1%, recall of 87.5%, an F1 score of 89.6%, a mAP of 89.6%, and an inference time of 110 ms per image, making the system suitable for real-time applications. The ensemble approach reduces errors by combining the outputs of SSD and YOLO [2]. The ensemble approach is also capable of identifying elephants at varying distances with high confidence and, the system is adaptable to diverse environmental conditions like low visibility, cluttered backgrounds, and dense scenes, ensuring consistent performance.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

- [1] P. D., S. S., Automated detection of elephant using ai techniques, Springer-Verlag 404 (2022). doi:10.1007/978-981-19-6406-04.
- [2] N. Mamat, M. F. Othman, F. Yakub, Animal intrusion detection in farming area using yolov5 approach, in: 22nd International Conference on Control, Automation and Systems, Jeju, Korea, 2022, pp. 1–5. doi:10.23919/ICCAS55662.2022.10003780.
- [3] G. P, M. Nivin, M. Rajeshwari, Advances in ecological surveillance: Real- time wildlife detection using mobilenet-ssd v2 convolutional neural network, IJRASET Journal For Research in Applied Science and Engineering Technology 11 (2024) 2333–2345. doi:10.22214/ijraset.2023.57847.
- [4] W.Xue, T.Jiang, J.Shi, Animal intrusion detection based on convolutional neural network, in: 17th International Symposium on Communications and Information Technologies (ISCIT), Cairns, QLD, Australia, 2017, pp. 1–5. doi:10.1109/ISCIT.2017.8261234.
- [5] B. T. Sibusiso Reuben Bakana, Yongfei Zhang, Wildare- yolo: A lightweight and efficient wild animal recognition model, in: Ecological Informatics, volume 80, 2024. doi:10.1016/j.ecoinf.2024.102541.
- [6] M. Hussain, Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection, 2023. doi:10.3390/machines11070677.
- [7] W. Liu, D. Anguelov, D. Erhan, C. S. S. R. C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, Springer International Publishing (2016). doi:10.1007/978-3-319-46448-02.
- [8] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G. Shen, T.-T. Lee, An improved object detection model for embedded systems, in: R. N. Smythe, A. Noble (Eds.), International Conference on System Science and Engineering (ICSSE), Paparazzi Press, Cairns, QLD, Australia, 2020, pp. 1–5.
- [9] D. M. P. F. Felzenszwalb, R. B. Girshick, D. Ramanan, Object detection with discriminatively trained part-based models 32 (2010). doi:10.1109/TPAMI.2009.167.
- [10] A. Biglari, W. Tang, A vision-based cattle recognition system using tensorflow for livestock water intake monitoring, IEEE 6 (2022). doi:10.1109/LSSENS.2022.3215699.
- [11] K. C. B. Bijuphukan Bhagabati, Kandarpa Kumar Sarma, An automated approach for human-animal conflict minimisation in assam and protection of wildlife around the kaziranga national park using yolo and senet attention framework, Ecological Informatics 79 (2024). doi:10.1016/j.ecoinf.2023.102398.
- [12] D. M. H. H. . M. A. Yuvaraj Munian, Antonio Martinez-Molina, Intelligent system utilizing hog and cnn for thermal image-based detection of wild animals in nocturnal periods for vehicle safety, Applied Artificial Intelligence (2022). doi:10.1080/08839514.2022.2031825.
- [13] Z. Xu, T. Wang, A. K. Skidmore, R. Lamprey, A review of deep learning techniques for detecting

animals in aerial and satellite images, in: R. N. Smythe, A. Noble (Eds.), *International Journal of Applied Earth Observation and Geo information*, volume 128 of *LAC '10*, 2024, pp. 1569–8432. doi:10.1016/j.jag.2024.103732.

- [14] Sugumar, S. . J. . R., An improved real time image detection system for elephant intrusion along the forest border areas, *The Scientific World Journal* (2014). doi:10.1155/2014/393958.
- [15] D.Yudin, A.Sotnikov, A.Krishtopik, Detection of big animals on images with road scenes using deep learning, in: *International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI)*, volume 3, Belgrade, Serbia, 2019, pp. 100–103. doi:10.1109/IC-AIAI48757.2019.00028.
- [16] Gupta, S. Mohan, N. Nayak, P. et al, Deep vision – based surveillance system to prevent train–elephant collisions (2022). doi:10.1007/s00500-021-06493-8.
- [17] Wild elephant dataset, 2024. URL: <https://www.kaggle.com/datasets/gunarakulangr/sri-lankan-wild-elephant-dataset>, [Accessed in May 15, 2024].