A YOLOv8-based AI System for Real-Time Endemic Species Threat Detection and Response

Article *in* Journal of Innovative Image Processing · March 2025

DOI: 10.36548/jiip.2025.1.003

Nalayini C M
Velammal Engineering College

A YOLOv8-based AI System for Real-Time Endemic Species Threat Detection and Response

Nalayini C.M.[1], Kalpana V.[2], Hemamalini S.[3], Sathyamoorthy K.[4]

[1]Assistant Professor, Department of Information Technology, Velammal Engineering College, Chennai, India.

[2]Associate Professor, Department of Computer Science and Engineering VelTech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India.

[3]Associate Professor, Department of Artificial Intelligence and Data Science, Panimalar Engineering College, Chennai, India.

[4]Assistant Professor, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.

E-mail: [1]nalayinicm13@gmail.com, [2]kalpanavadivelu@gmail.com, [3]hemamalini.phd2020@gmail.com, [4]pitsathyamoorthy@gmail.com

Abstract

Endemic species are under threat from various factors, including habitat destruction, illegal hunting, and climate change, which necessitate urgent and effective monitoring solutions. This research introduces an advanced AI system for real-time threat detection and response, utilizing the YOLOv8 algorithm. The system is specifically developed for the protection of endemic species. It combines the strengths of YOLOv8 with enhancements like a multi-scale detection module to tackle the challenges of identifying small or camouflaged species and threats across different ecological environments. In the proposed study, a customized dataset that is developed through the application of the Histogram of Oriented Gradients (HOG) technique along with the Firefly Algorithm is employed. This approach facilitates the efficient fine-tuning of all images within the dataset, enhancing the overall effectiveness of the analysis. The Roboflow platform is used for training, validation, and testing the customized dataset for real-time object detection. YOLOv8 achieves 97.9% mAP, 94.7% precision, and 91.9% recall. Threats are recorded in a Blockchain ledger and sent to Twilio SMS alert system, making it cost-effective and efficient. The proposed framework offers high

Journal of Innovative Image Processing, March 2025, Volume 7, Issue 1, Pages 50-73

DOI: https://doi.org/10.36548/jiip.2025.1.003
Received:07.02.2025, received in revised form: 02.03.2025, accepted: 16.03.2025, published: 27.03.2025

accuracy, precision, and recall, minimizing false alarms and facilitating quick interventions, making it suitable for smart environmental management systems and biodiversity conservation.

Keywords: YOLOv8, Roboflow, Object Detection, Blockchain Technology, Twilio.

1. Introduction

Endemic species are facing increasing threats to their unique characteristics and vulnerabilities due to environmental stresses and human activities. Illegal hunting, habitat encroachment, poaching, and competition for resources are putting their survival at serious risk. These species, often confined to specific geographic areas, are particularly vulnerable to such dangers and may ultimately face extinction [1]. The illegal wildlife trade, driven by financial gain, exacerbates the decline of these species through direct hunting and trafficking. Additionally, habitat loss from urbanization, agricultural expansion, and deforestation further reduces the areas where these species can thrive. Human competition and climate change have led to resource scarcity, disrupting the delicate ecosystems they depend on. To ensure the survival and protection of endemic species in an increasingly hostile environment, these interconnected challenges require robust and innovative conservation strategies that extend beyond traditional methods [2].

The long and complex history of wildlife poaching is driven by human greed for resources, land, and animal products. Early instances of poaching can be traced back to the middle ages, when hunting regulations prohibited the hunting of certain wildlife, making it a criminal offense. Over time, poaching evolved from a means of subsistence hunting into a business fuelled by the global demand for exotic goods. The expansion of European powers and the commercialization of wildlife during the colonial era led to widespread exploitation, particularly in Africa and Asia, where big cats, rhinos, and elephants were hunted for their ivory, horns, and skins [3]. The 20th century saw a dramatic increase in poaching, aided by advancements in weapons and transportation that allowed poachers to access remote areas and more effectively target endangered species. Despite conservation efforts and laws aimed at protecting wildlife, the global illegal wildlife trade flourished due to the high demand for luxury items, exotic pets, and traditional medicine. In recent decades, poaching has become a transnational issue linked to terrorism, arms trafficking, and corruption, largely due to the growing involvement of organized crime syndicates. This on-going threat to biodiversity undermines conservation efforts and pushes many species closer to extinction.

The prevention of poaching for endemic species has become more effective and efficient using AI-driven technology and real-time communication networks. Tools like Twilio, Roboflow [4], and YOLOv8 offer a robust set of resources to address the complex challenges of wildlife monitoring and threat response. YOLOv8, a state-of-the-art object detection model, enhances the ability to identify poachers and potential threats by quickly recognizing key features. To improve the accuracy of these models, effective annotation and augmentation of animal monitoring datasets are essential, and Roboflow simplifies this data preparation process. Additionally, Twilio's messaging service enables conservation teams to receive immediate alerts, ensuring swift action when threats are detected. This integration of AI, data management, and communication technologies not only boosts the accuracy and speed of poaching detection but also provides a scalable and cost-effective framework for broader conservation efforts. By utilizing these technologies, conservationists can create a transparent, reliable, and responsive system to protect endangered species from human threats.

1.1 Histogram of Oriented Gradients (HOG)

HOG is a traditional method that focuses on detecting object shapes and edges by analysing gradient directions in an image. HOG descriptors can be incorporated into the training pipeline as an additional input with raw images, potentially improving the model’s understanding of shape-based patterns.

1.2 Firefly Algorithm

The Firefly Algorithm (FA) improves real-time object detection by making YOLOv8 smarter and more effective. It helps identify the most important HOG features, ensuring that only the most relevant details are passed to the model, which improves accuracy. FA also fine-tunes key parameters like confidence-threshold, non-maximum suppression (NMS), anchor boxes, learning rate, and batch size, helping YOLOv8 detect objects more precisely. Additionally, it refines bounding boxes, adjusting their positions and sizes for better alignment with detected objects, leading to higher Intersection over Union (IoU) scores

1.3 YOLOv8 – An Overview

YOLOv8 (You Only Look once, version 8) is an advanced object detection model that builds on the core principles of convolutional neural networks (CNNs). As the latest version in the YOLO series, YOLOv8 [5] aims to deliver high accuracy and real-time performance for tasks such as segmentation, classification, and object detection. It features significant architectural improvements that enhance the capabilities of CNNs for precise object localization and efficient feature extraction. At the core of YOLOv8's architecture are convolutional layers, which play an essential role in capturing spatial information from images. These layers form the basis for object recognition by detecting fundamental patterns like edges, textures, and shapes. The design of YOLOv8 includes a multi-scale detection framework that boosts its ability to recognize objects of varying sizes, ensuring that both large and small items are accurately identified within the same image. The model employs a deep CNN backbone network that has been specifically optimized to balance accuracy and speed. By predicting object centres directly, YOLOv8 simplifies the detection process, enhancing both speed and accuracy compared to traditional methods that rely on fixed anchor boxes. The model's detection head enables real-time performance by utilizing additional CNN layers to predict bounding boxes, class probabilities, and confidence scores all in one go. YOLOv8 sets a new benchmark for object recognition, delivering both speed and precision.

1.4 Development Platform – Roboflow

Roboflow is a powerful framework designed to simplify the creation, maintenance, and implementation of computer vision models. It offers a wide array of features for tasks such as segmentation, image classification, and object recognition. By prioritizing automation and user-friendly interfaces, Roboflow [6] streamlines data management, annotation, augmentation, and model deployment for developers, researchers, and businesses. Its intuitive design allows users to concentrate on training high-performance models without the complications of infrastructure setup and data preparation.

Roboflow excels in model deployment by offering various options customized to different needs. Users can deploy models as hosted APIs, making it easy to integrate them into applications without any difficulty in managing the infrastructure. For applications requiring on-device processing, Roboflow supports deployment to edge devices like the Raspberry Pi and NVIDIA Jetson, enabling real-time inference. The platform's monitoring tools allow users to regularly evaluate and enhance their models, providing detailed performance metrics such as precision, recall, and mean Average Precision (mAP).

Moreover, Roboflow supports continuous monitoring, which helps users track model performance over time. This feature is particularly valuable in dynamic environments where data distributions can change, necessitating frequent updates to maintain accuracy. By offering robust training and deployment tools, Roboflow ensures that models remain reliable and effective throughout their lifecycle.

Roboflow is widely used across various industries. In industrial automation, it assists in identifying defects on production lines and ensuring quality control. In healthcare, it plays a crucial role in analysing medical images, which helps doctors with diagnosing diseases and planning treatments. The platform is also beneficial in retail, where it streamlines inventory management and supports automated checkout processes. In agriculture, Roboflow utilizes drone footage to detect pests and monitor crop health. In wildlife conservation, Roboflow enables real-time monitoring of animal populations, aiding anti-poaching efforts and species preservation. By deploying models on edge devices, conservationists can observe wildlife behaviour in remote locations [7]. Additionally, Roboflow's models enhance public safety in security and surveillance by quickly identifying potential threats.

1.5 Blockchain Technology

Blockchain technology is employed to manage data transparently and securely, ensuring that detection events, patrol activities, and interventions are recorded in an immutable ledger. This multidisciplinary approach not only enhances surveillance capabilities but also establishes a robust, transparent, and scalable framework for wildlife protection. The combination of blockchain technology with advanced AI and communication tools ensures a proactive, secure, and collaborative effort to protect endangered species and their habitats.

1.6 Twilio: A Communication Platform

Twilio is a cloud-based communication platform that allows developers to integrate audio, video, and messaging features into their applications. In the context of wildlife poaching [8] detection systems, Twilio can be invaluable by offering real-time alerts and notifications when risks are detected. With Twilio's messaging services, stakeholders receive instant updates, which enhance response times and boost the chances of preventing poaching incidents.

Twilio ensures quick alarm transmission, minimizing the time between identifying a threat and taking action. It offers broad coverage, allowing notifications to be sent even in remote areas with unreliable internet, with its support for international SMS and phone networks. Users can customize notification workflows to meet specific operational needs, ensuring that alerts reach the right people at the right time [9]. Additionally, Twilio's robust infrastructure provides high uptime and reliable message delivery for time-sensitive applications.

Motion sensors and cameras monitor protected areas in a wildlife reserve equipped with a YOLOv8-based poaching detection system. This system utilizes YOLOv8 for processing and classifying images when a camera captures a suspected poacher. If a threat is detected, rangers and security personnel receive an SMS containing the image and location coordinates, facilitated by Twilio's API [10]. To guarantee a quick response, the nearest ranger station is also alerted through an automated voice call.

2. Related Works

By addressing challenges such as data imbalance and unpredictability, a machine learning system predicts the likelihood of poaching. Gaussian processes enhance prediction accuracy by directing patrols to high-risk areas and optimizing routes. This method has been tested in national parks in Cambodia and Uganda, demonstrating a 30% improvement in snare detection, which aids in animal protection. By integrating with conservation software, the goal is to expand this approach globally, safeguarding endangered species and supporting law enforcement with limited resources [11].

An IoT-based system that incorporates machine learning enhances animal protection by identifying poachers in game reserves. This framework employs image analysis and ambient sensors to monitor activity and improve detection accuracy. Trial results indicate its effectiveness in enhancing surveillance and reducing poaching incidents. This technology-driven approach bolsters conservation efforts across Africa's protected areas by offering a scalable solution to combat illegal hunting [12].

Utilizing probabilistic spatial-temporal graphs, a behavioural model examines the movements of animals and poachers to guide rangers and drones in coordinated patrols. By applying genetic algorithms and dynamic programming to optimize patrol routes, this anti-poaching strategy maximizes wildlife protection. Simulations and extensive data-driven research reveal an over 90% improvement in animal protection. This model provides a robust foundation for effectively preventing poaching through computational techniques and predictive insights [13].

A modified YOLOv8 model enhances UAV-based object detection by incorporating double prediction heads and advanced feature fusion. These modifications streamline the model while boosting the accuracy of small-object recognition. The introduction of a coordinate attention module aids in maintaining target focus amidst complex backgrounds. This approach proves beneficial for aircraft surveillance in challenging environments, such as security operations and wildlife monitoring, as experimental results show significant improvements in detection metrics [14].

Diseases affecting apple leaves, like apple scab and black rot, can be effectively identified using a refined YOLOv8 model. By optimizing convolutional layers and utilizing specific datasets, the model achieves an impressive 99.5% accuracy along with a high recall rate. This system supports proactive management and automated disease monitoring in agriculture, helping to reduce crop losses. Its capability to accurately detect diseases positions it as a vital tool for precision farming and smart agricultural practices [15].

To aid in the early detection of colorectal cancer, a deep learning-based framework employs YOLOv8 to identify polyps in endoscopic images. Enhancements such as a customized evaluation metric and contrast adjustment improve both the speed and accuracy of detection. This method plays an important role in reducing cancer mortality by enabling timely screenings and achieving high precision and recall rates. This strategy highlights the role of AI in automated diagnosis and medical imaging, providing cost-effective solutions to urgent global health challenges [16].

YOLOv8-based convolutional neural networks analyse facial expressions to predict six distinct mood states. The system achieves high precision and recall rates through effective feature extraction and iterative training. This application enables human-computer interaction that responds to users' emotional states, enhancing emotion-aware technology. The model's performance supports its use in customer service, adaptive user experiences, and mental health monitoring [17].

YOLOv8 is utilized by a deep learning network to detect and classify corrosion in industrial structures. Its single-pass detection enables effective analysis, while cloud-based deployment ensures scalability and accessibility. This technology provides accurate and rapid insights into structural integrity, thereby improving maintenance practices. Automated corrosion detection significantly benefits industrial asset management by reducing inspection costs, minimizing downtime, and enhancing safety in critical infrastructure [18]. The study compares deep learning models like DenseNet, ResNet, VGGNet, and YOLOv8 for wildlife species classification on a custom dataset of 575 images of 23 endangered species. YOLOv8, with its advanced architecture and efficient feature extraction, outperforms other models [19] The summary of the literature review is presented in Table 1.

Table 1. Summary of Literature Review

Paper Reference	Algorithm Used	Merits	Demerits
[11]	Machine learning, Gaussian processes	Improved snare detection by 30%, optimized patrol routes, robust predictions under uncertainty	High class imbalance, reliance on historical data, computational complexity
[12]	IoT, Image analysis, Machine learning	Enhanced poacher detection, scalable framework, real-time monitoring	Dependency on sensor network, requires extensive infrastructure
[13]	Probabilistic spatial-temporal graphs, Genetic algorithms, Dynamic programming	High improvement ratio (90%), coordinated patrols, effective resource utilization	NP-complete problem, high computational overhead
[14]	YOLOv8, Double Prediction Heads, Feature Fusion	Improved small-object detection, reduced model complexity, background noise reduction	Tailored for UAVs, may not generalize to other scenarios
[15]	YOLOv8, Convolutional layers, Plant Village dataset	High accuracy (99.5%), effective disease identification, real-time monitoring	Limited to specific plant diseases, potential dataset limitations
[16]	YOLOv8, CLAHE, Custom evaluation metrics	High precision in polyp detection, aids early diagnosis, cost-effective medical screening	Specific to endoscopic images, may not perform well on other datasets
[17]	YOLOv8, Convolutional Neural Networks (CNN)	Accurate mood state prediction, supports emotion-aware systems	Requires extensive training data, sensitive to facial feature variations
[18]	YOLOv8, Cloud deployment (WandB Cloud)	Efficient corrosion detection, cloud	Cloud reliance may introduce latency,

		scalability, reduced downtime	dependent on network availability
[19]	DenseNet, ResNet, VGGNet, and YOLOv8	Advanced architecture and efficient feature extraction	Limited to specific endangered species, potential dataset limitations

3. Proposed Methodology

We have taken the African Wildlife dataset [20] from ultralytics. It involves African wild animals with 1052 training images, 225 validation images, and 227 testing images. It has four classes of African animals such as elephant, rhino, buffalo, and zebra. In the pre-processing stage, a customized dataset is obtained by augmenting 100 human images to identify threat detection and also applying Hog for best feature extraction and Firefly algorithm for better optimization by involving hyper tuning parameters to fine-tune the metrics such as confidence-threshold. Then the customized dataset is uploaded to the Roboflow platform to create a new dataset version with 1463 images in total involving 1174 images for training, 150 images for validation, and 139 images for testing. It has five classes of objects such as elephant, rhino, buffalo, zebra, and humans (poachers). YOLOv8 is trained with the customized dataset in the Roboflow platform for efficient real time objects and threat detection. The proposed model has produced 97.9% mAP, 94.7% precision and 91.9% recall as the best results when compared to the existing real time object detection systems. For continuous monitoring the blockchain technology was added for immutable, decentralized, and transparent storage and a Twilio communication system was used to send alert messages to the respective forest officers to take action immediately after the threat is detected in real-time as shown in Figure 1.

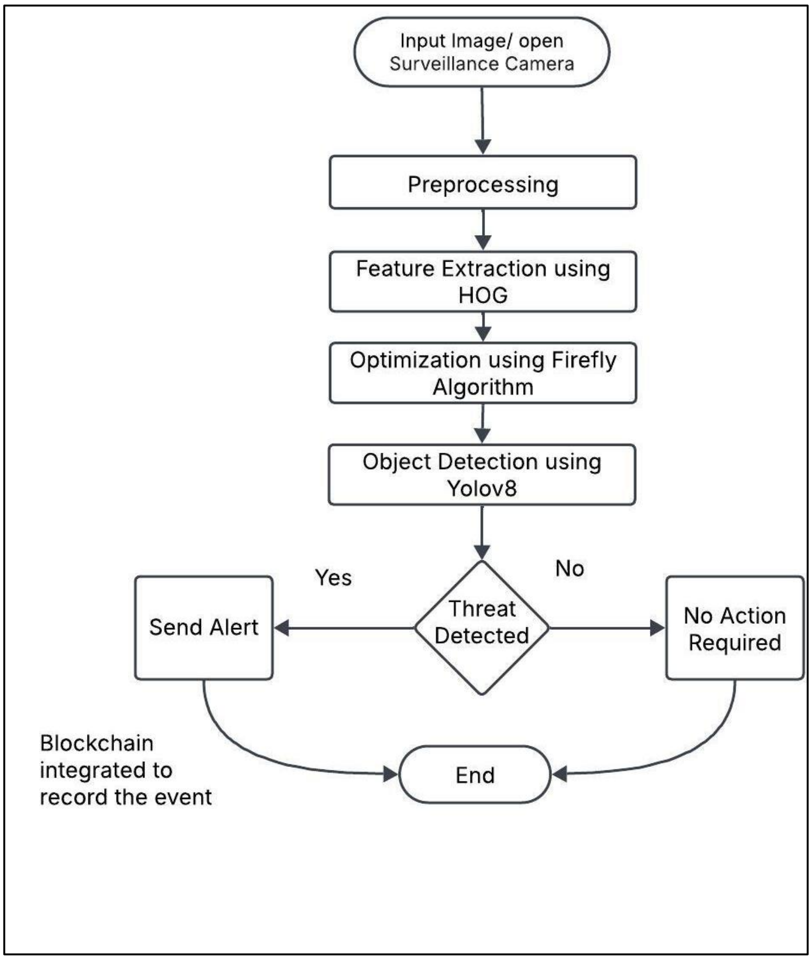

Figure 1. Proposed System Work Flow

3.1 HOG Feature Extraction

Instead of passing raw images to YOLOv8, all the input images are HOG-extracted images to remove unnecessary noise, to ensure stable detection under different lighting and to prevent misclassification. It provides essential features which are required for efficient endemic species detection. Input image is read for image modification and visualization using OpenCV. Hog is applied to analyse the gradients from the input image to capture object shapes (edge orientation). Edge orientation cannot be done for colour images. So the input image is converted into grayscale. Each pixel's intensity is converted into a single value. Sine every object has different edges, HOG computes gradient for every image. Gradient Horizontal, vertical, magnitudes and directions are calculated using sobel filters to find the changes in intensity at each pixel. Hog fragments the input image into non-overlapping small cells with 8*8 pixels. To reduce computational complexity, local edge structures are captured by a histogram of gradient orientations for every 8*8 region. Each pixel's gradient direction is allocated to a bin that is 9 bins covering 0 degrees to 180 degrees. Gradient magnitude will keep track of the bin count. Every block consisting of multiple cells of 16*16 pixels is normalized to make scale-invariant features. HOG visualization enhances the edges which are equivalent to object contours. All the normalized histograms are combined into a feature vector which represents the entire image. HOG analyses the edges, shapes, and key features of an image which helps YOLOv8 model to detect the real-time objects accurately. It minimizes the impact of lighting changes and background noise, ensuring reliable recognition in different circumstances. HOG extracts only the most important shape-based details to make object detection faster and more efficient. HOG provides better accuracy, lower false positives, and improved robustness in all environments

Histogram of oriented gradients (HOG) improves the YOLOv8 model's ability to detect endemic species by

- Improving edge and texture detection to recognize object boundaries faster.
- Reducing noise by removing unwanted background details that mislead exact detection
- Refining gradients to improve the bounding box accuracy
- Providing low-level features like gradients and edges.

3.2 Firefly Algorithm

Firefly algorithm is applied to optimize the image pre-processing parameters such as the HOG parameters (cell size, block size, and orientation bins), brightness and contrast adjustments, and noise reductions which improve the reliability of the object detection. The Firefly algorithm finds the best pre-processing parameters to increase the maximum number of object detection by the YOLOv8 model.

3.3 Blockchain Technology

After the real time endemics are detected by YOLOv8, immediately the information about the objects are stored in a blockchain ledger. With the help of blockchain technology, editing or deleting data is impossible, because blockchain technology is immutable, transparent and decentralized. This helps to track the endemic species details instantly. Every block consists of timestamp to record the time of detecting the endemic species, information about type of species detected, to maintain current and previous hash to link the respective blocks in the blockchain and unique block number to identify the respective block to retrieve the required object detection details. In wildlife environment, blockchain technology helps us to prevent from tampering of data an ensures efficient object tracking.

3.4 Twilio Communication Systems

Twilio communication system is triggered as soon the threat is detected by the YOLOv8 model. It immediately sends an SMS to the forest officers about the threat detected to take immediate actions. Twilio handles multiple object detections and informs the same to the security guards. It provides stable message delivery to ensure reliability.

4. Results and Discussion

American Wildlife Dataset is downloaded from ultralytics.com [20] with 1052 images for training, 225 images for validation and 227 images for testing, and a .yaml file. This existing dataset consists of 4 classes of wild animals such as elephant, rhino, buffalo and zebra. To detect threats in real-time object detection, poacher-related 100 images were added to the existing dataset. Finally, a customized dataset with five classes such as elephant, rhino, buffalo, zebra, human (poacher) were obtained. Two mechanisms, HOG for best feature extraction and Firefly algorithm to optimize images with pre-processing parameters were applied to the images. With the HOG extraction process the given input image of size 128*128 pixels, is divided into 8*8 pixel cells. For each cell, the gradient magnitudes and orientations are calculated. These orientations are then quantized into 9 bins, creating a 9-dimensional histogram for each cell. Cells are then grouped into larger blocks of size 16*16 pixels, and the histograms within these blocks are normalized. The normalized block histograms are concatenated to form the final feature vector, resulting in a high-dimensional representation of the image as shown in Figure 2.

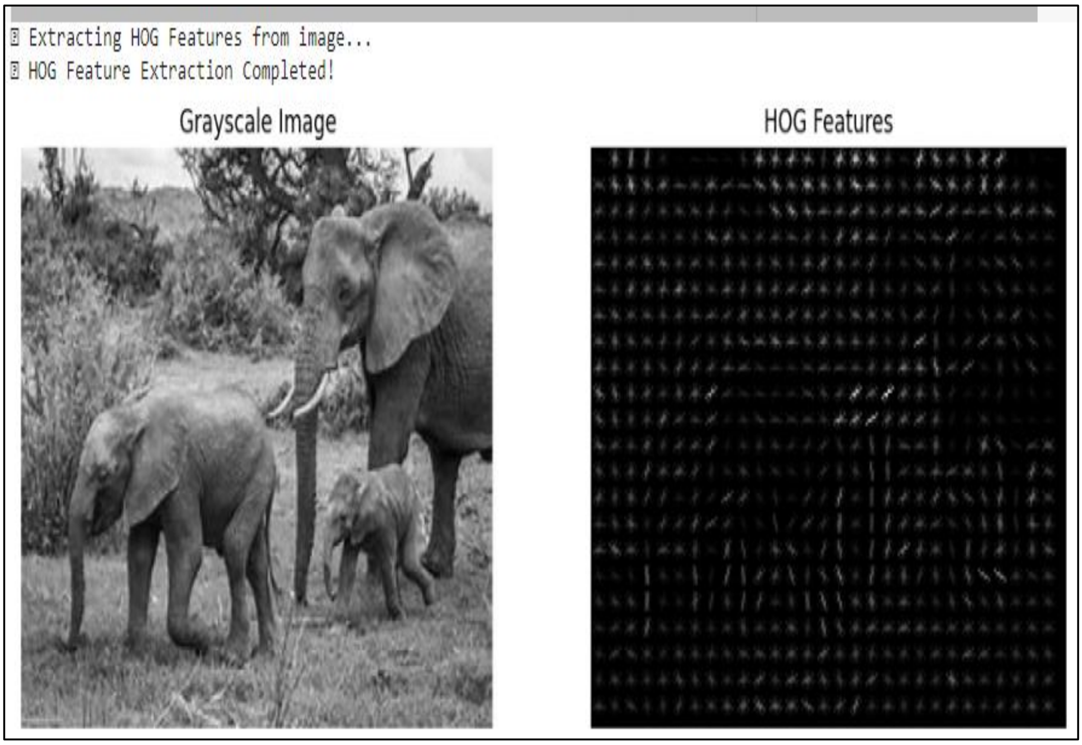

Figure 2. Hog Feature Extraction

The Firefly Algorithm is employed to optimize image pre-processing parameters, including HOG parameters (cell size, block size, and orientation bins), brightness, and contrast. These parameters are dynamically adjusted for each firefly, with noise reduction techniques, to enhance the reliability of object detection. Ten fireflies are initialized with varying brightness and contrast values for the input image. A fitness function, serving as the loss function, is computed for each firefly. Over 50 iterations, fireflies move based on their relative brightness and proximity to other fireflies, aiming to converge towards optimal parameter sets. The algorithm evaluates combinations of pre-processing parameters, selecting those yielding the best brightness and contrast values. The optimal parameters, achieving a confidence of 0.53 and an Intersection over Union (IoU) of 0.41, are then used, as illustrated in Figure 3. The Firefly Algorithm identifies pre-processing parameters that maximize object detection by the YOLOv8 model. Finally, these optimized parameters are applied to the input image before it is processed with the customized dataset. This process of HOG feature extraction and Firefly Algorithm-driven pre-processing optimization is consistently applied to all images.

```
image 1/1 /content/sample_data/elephant.jpg: 512x640 3 elephants, 129.6ms
Speed: 3.5ms preprocess, 129.6ms inference, 0.8ms postprocess per image at shape (1, 3, 512, 640)
▯ Loss Function Output: -1
▯ Evaluating Loss Function (conf=0.80, iou=0.68)...

image 1/1 /content/sample_data/elephant.jpg: 512x640 3 elephants, 138.7ms
Speed: 2.8ms preprocess, 138.7ms inference, 0.8ms postprocess per image at shape (1, 3, 512, 640)
▯ Loss Function Output: -1
▯ Iteration 1/50 of Firefly Optimization...
▯ Iteration 2/50 of Firefly Optimization...
▯ Iteration 3/50 of Firefly Optimization...
▯ Iteration 4/50 of Firefly Optimization...
▯ Iteration 5/50 of Firefly Optimization...
▯ Iteration 6/50 of Firefly Optimization...
▯ Iteration 7/50 of Firefly Optimization...
▯ Iteration 8/50 of Firefly Optimization...
▯ Iteration 9/50 of Firefly Optimization...
▯ Iteration 10/50 of Firefly Optimization...


▯ Optimizing Detection Parameters with Firefly Algorithm...
▯ Running Firefly Optimization Algorithm...
▯ Evaluating Loss Function (conf=0.53, iou=0.41)...

image 1/1 /content/sample_data/elephant.jpg: 512x640 3 elephants, 131.2ms
Speed: 2.9ms preprocess, 131.2ms inference, 0.8ms postprocess per image at shape (1, 3, 512, 640)
▯ Loss Function Output: -1
▯ Evaluating Loss Function (conf=0.59, iou=0.58)...

image 1/1 /content/sample_data/elephant.jpg: 512x640 3 elephants, 131.3ms
Speed: 3.2ms preprocess, 131.3ms inference, 0.8ms postprocess per image at shape (1, 3, 512, 640)
▯ Loss Function Output: -1
▯ Evaluating Loss Function (conf=0.37, iou=0.24)...

image 1/1 /content/sample_data/elephant.jpg: 512x640 3 elephants, 139.0ms
Speed: 2.9ms preprocess, 139.0ms inference, 0.8ms postprocess per image at shape (1, 3, 512, 640)
▯ Loss Function Output: -1
▯ Evaluating Loss Function (conf=0.08, iou=0.56)...

▯ Iteration 50/50 of Firefly Optimization...
▯ Firefly Optimization Completed! Best parameters found: [    0.53423     0.41413]
▯ Best Parameters Found: Confidence=0.53, IoU=0.41
```

Figure 3. Firefly Optimization

Once the dataset is customized with the help of Hog and Firefly algorithms, then the dataset is uploaded into Roboflow platform. A new dataset version was created, comprising a total of 1,463 source images across five classes: 699 zebra images, 741 elephant images, 542 rhino images, 538 buffalo images, and 100 poacher images. The dataset contained 2,620 annotations, with an average image size of 0.31 MP and an average image ratio of 640x480. Roboflow was used to split the source images into a training set of 1,174 images, a validation set of 150 images, and a test set of 139 images. Auto-orient pre-processing was applied. The distribution of train, validation, and test data for the YOLOv8 model is depicted in Figure 4.

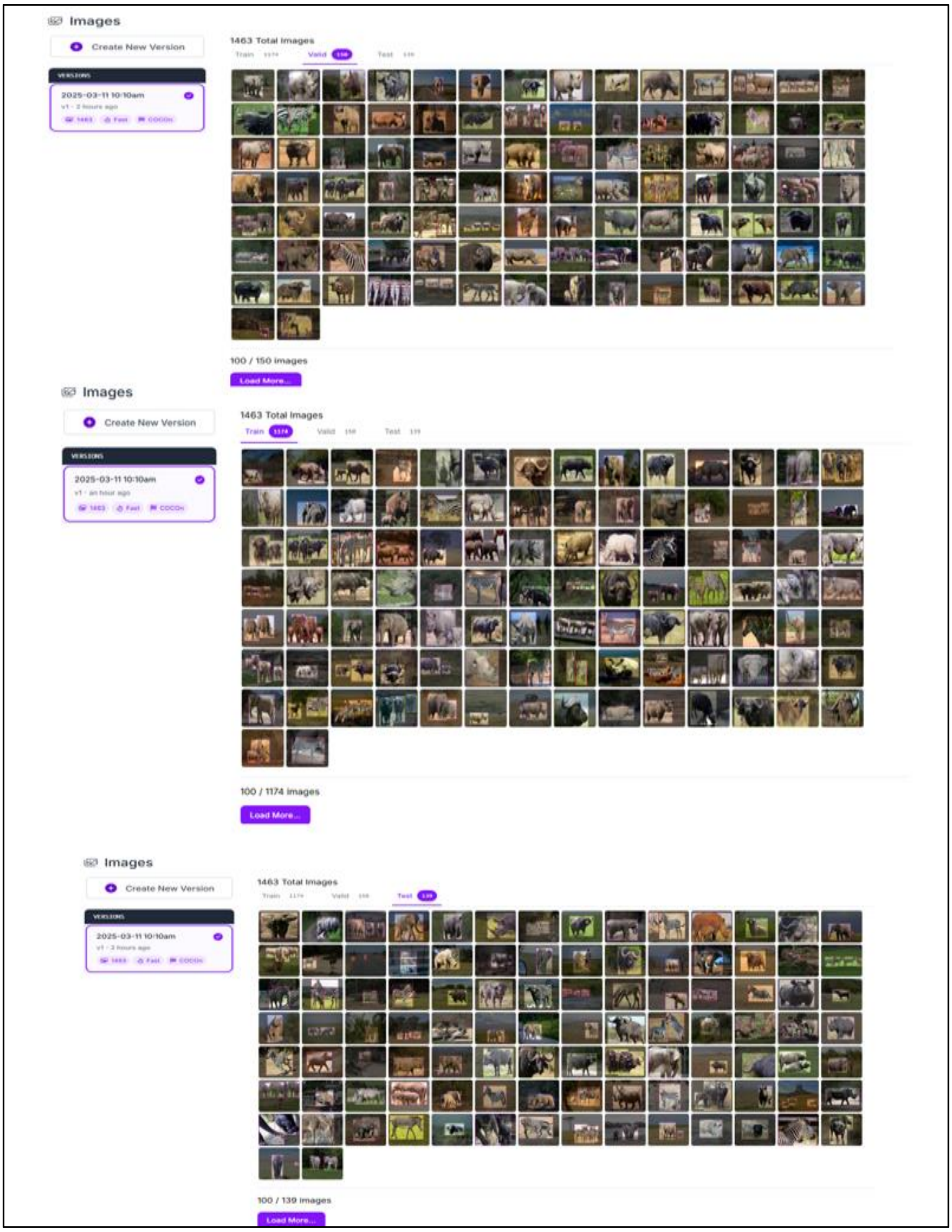

Figure 4. Train, Valid and Test Set Image Data

The training process was carried out using the Roboflow 3.0 Object Detection (YOLOv8 Fast) model type and the MS COCO checkpoint. Detailed training graphs for parameters such as box loss, class loss, and object loss, along with mAP, precision, and recall for both training and validation sets, are clearly depicted in Figure 5.

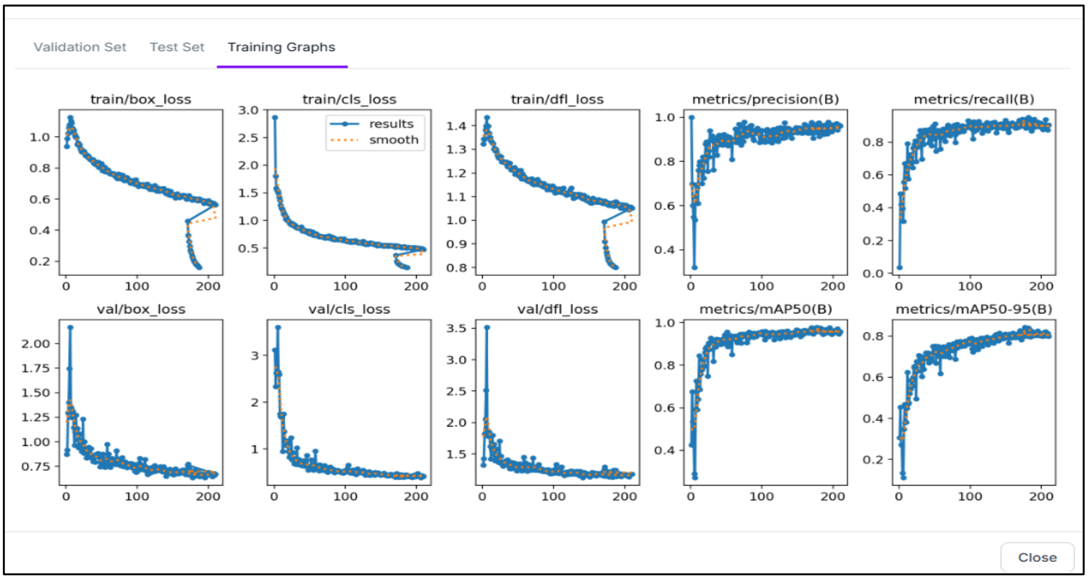

Figure 5. Training Graphs

Now, Image with one object and two objects are given as input to the trained model for validation, it shows object detection with confidence score as shown in the Figure 6 and Figure 7.

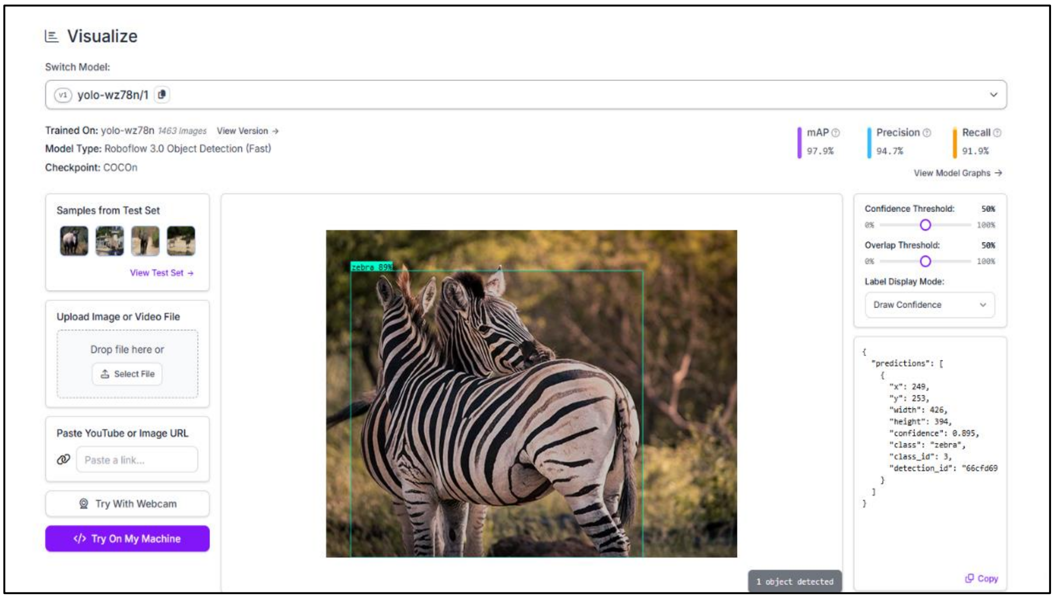

Figure 6. YOLOv8 Trained Model Output: 1 Object Detected with Confidence Score

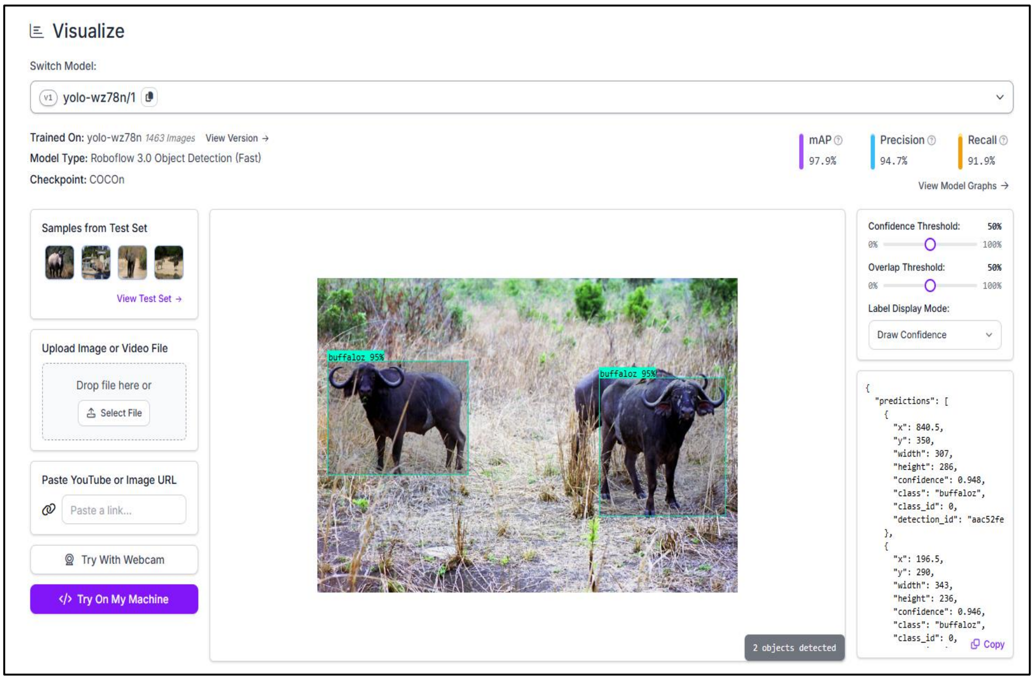

Figure 7. YOLOv8 Model Output: 2 Objects Detected with Confidence Score

Now image with three objects is given for testing into the trained model, it shows 3 objects detected and the information is recorded in the blockchain ledger as shown in Figure 8.

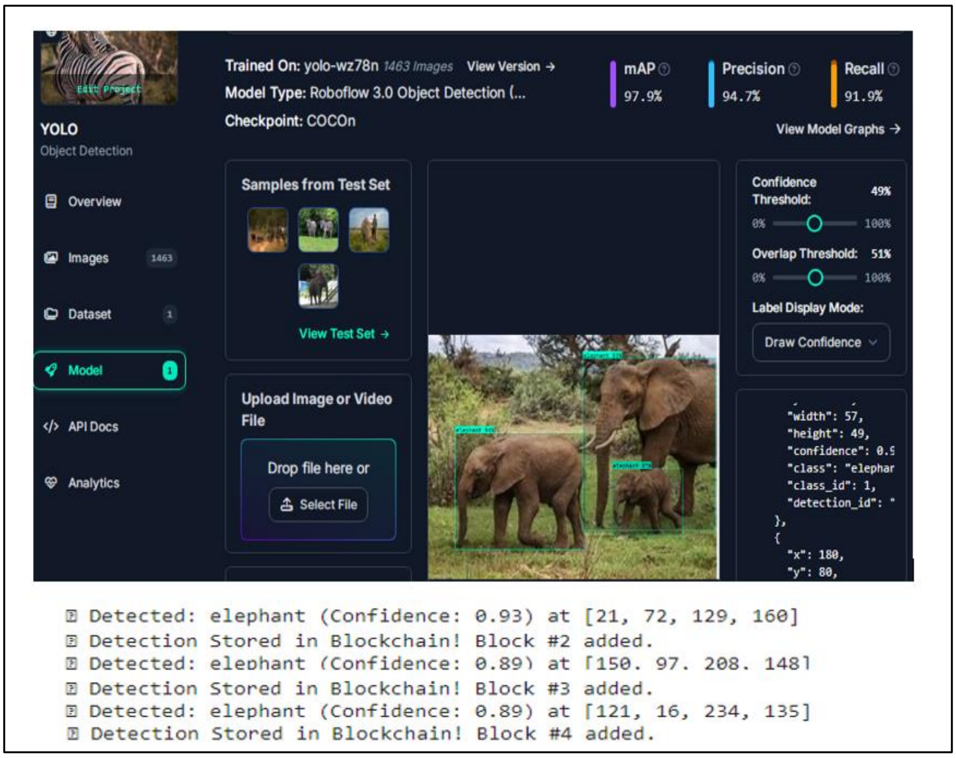

Figure 8. YOLOv8 Model Output: 3 Objects Detected with Confidence Score

A blockchain ledger is implemented using a Python class. The self.chain initializes the blockchain ledger, and the genesis block is created first. Each species detection is hashed using a Merkle tree through the cryptographic SHA-256 algorithm and stored in a new block. A proof-of-work consensus algorithm is used for verification and validation of the object detection data before it is added to the respective block of the blockchain. Each block includes an index, timestamp, detection data, previous hash, and current hash. The previous hash value is the hash of the last block, used to link it to the new block. The current hash is the new hash of the current block, computed from all the species detection data. If any tampering of the data is attempted, the hash will change immediately, and this change will be notified to the server for immediate action. The code converts the previous block hash and detection data into JSON format. The detection of endemic species is printed in JSON format, as shown in Figure 9.


```
Blockchain Ledger:
{
    "index": 1,
    "timestamp": 1741855272.4855871,
    "data": "Genesis Block",
    "previous_hash": "0",
    "hash": "0bf7f64a3345e0311d4745f3c76e4a17f734a0e1b3b1155dee70c473baf0004e"
}
{
    "index": 2,
    "timestamp": 1741855274.9771059,
    "data": {
        "species": "elephant",
        "location": "Unknown Location",
        "timestamp": 1741855274.9770975
    },
    "previous_hash": "0bf7f64a3345e0311d4745f3c76e4a17f734a0e1b3b1155dee70c473baf0004e",
    "hash": "6ff610024a6f534084e379102a34e1a9e388c1d9281b11d700dfd9ffd429f956"
}

{
    "index": 3,
    "timestamp": 1741855274.9774005,
    "data": {
        "species": "elephant",
        "location": "Unknown Location",
        "timestamp": 1741855274.9773977
    },
    "previous_hash": "6ff610024a6f534084e379102a34e1a9e388c1d9281b11d700dfd9ffd429f956",
    "hash": "803166dc70ea6c2e89c3378cfc57e50b8952599b60b4ca576be8d5734bacf3f9"
}
{
    "index": 4,
    "timestamp": 1741855274.977659,
    "data": {
        "species": "elephant",
        "location": "Unknown Location",
        "timestamp": 1741855274.9776556
    },
    "previous_hash": "803166dc70ea6c2e89c3378cfc57e50b8952599b60b4ca576be8d5734bacf3f9",
    "hash": "bfd0dcb1ca043bb96699ed7241e88aa723d42a121bca5da022b8dcf6fd1013b3"
}
```


Figure 9. Object Detections are Recorded in Blockchain Ledger

Now image with two objects is given for testing into the trained model, it shows 2 objects detected and the information is recorded in the blockchain ledger. As soon as the threat is detected by YOLOv8 model, immediately Twilio function is invoked. To set up Twilio, the Twilio library is installed, and a Twilio account, phone number, credentials, and the Twilio Python SDK are integrated for proper SMS alerts. Twilio's REST API client library is imported to send these alerts. A Twilio client is initialized using the Twilio_Account_SID and Twilio_Auth_Token, which represent the authenticated API requests. Twilio_Phone_Number is the registered sender ID, and Recipient_Phone_Number is the destination phone number for the alerts. Alert messages, constructed with the object detection details, are sent to the destination phone number through the sender's phone number, as shown in Figure 10. Twilio forwards the SMS to global telecommunication networks to notify the concerned personnel for timely action.

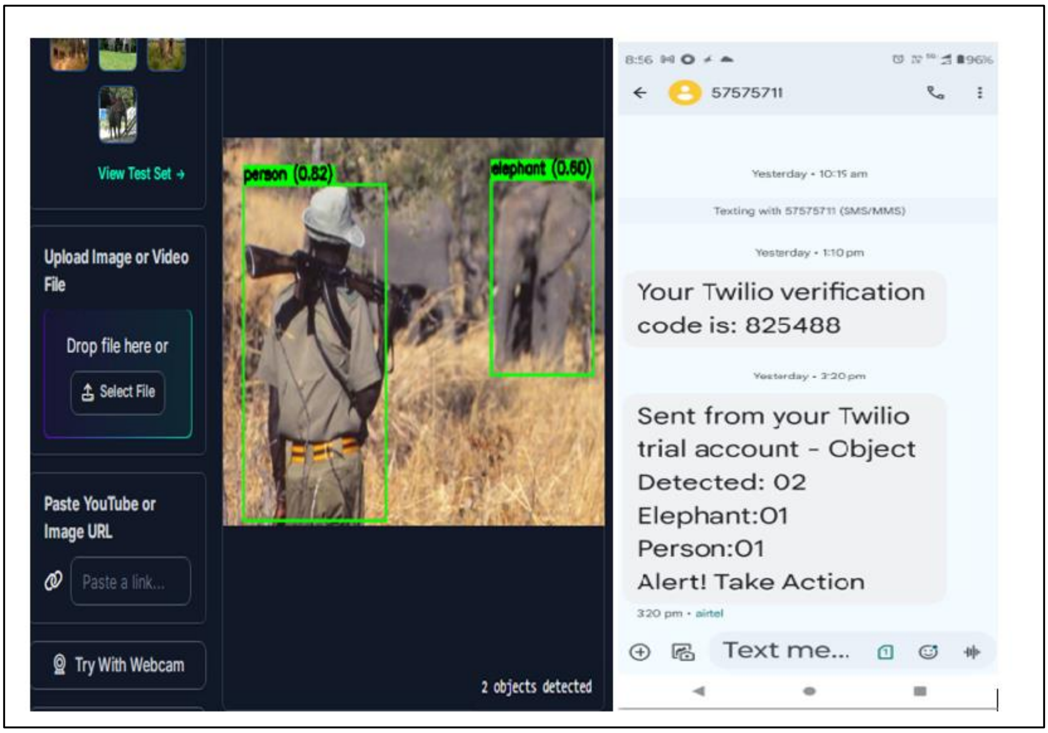

Figure 10. Twilio SMS Alert Notification

4.1 Comparative Analysis

The proposed hybrid model(HOG + Firefly + YOLOv8(Roboflow)) shows superior performance in all aspects such as mAP, Precision, and Recall when compared to Yolov8, YOLOv5 and SSD as shown in Table 2 and Figure 11, demonstrating its effectiveness for real-time endemic species threat detection.

Table 2. Performance Comparison of Object Detection Models

Comparison	mAP	Precision	Recall
Hybrid Proposed Model	97.9	94.7	91.9
Yolov8	91.2	90.6	89.7
Yolov5	89.2	88.8	87.1
SSD	85.3	83.4	82.2

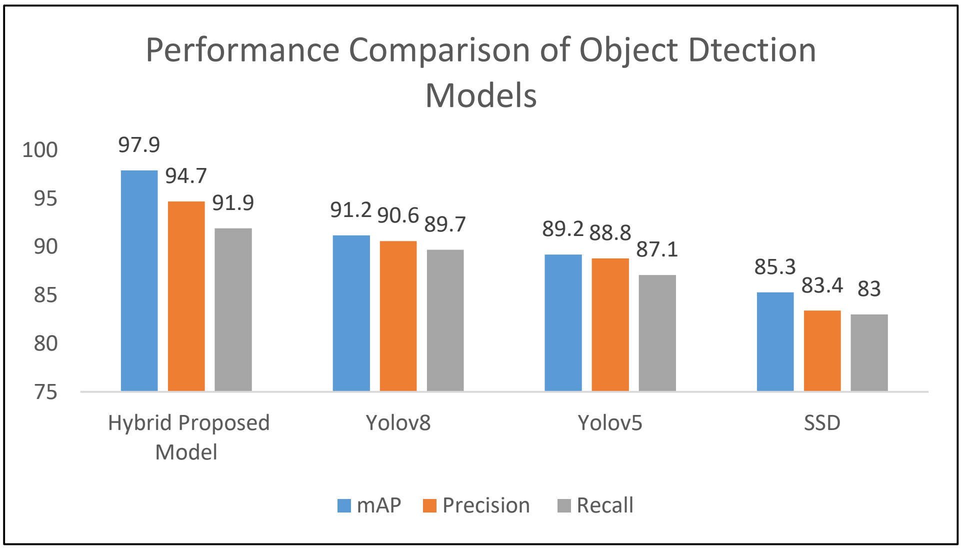

Figure 11. Comparison Chart

5. Conclusion

This enhanced system builds upon YOLOv8's robust CNN-based detection by integrating HOG to capture fine-tuned texture and shape features, aids in distinguishing between similar species even under varying lighting conditions. By selecting only the most relevant HOG features, noise is reduced, and classification accuracy is improved. Firefly Optimization efficiently fine-tunes key hyperparameters, such as learning rate, batch size, and confidence threshold, resulting in better generalization and faster convergence while lowering computational overhead. Additionally, blockchain technology is incorporated to record each detection in a blockchain ledger, providing a decentralized, immutable, and transparent mechanism. The study utilized the Twilio communication system to send real-time notifications to forest officers whenever a threat is detected. In conclusion, this integrated approach not only improves detection precision but also enhances system reliability. In the future work, GPS tracking will be included to facilitate effective action by forest officers. The future work will also address diverse datasets to provide efficient solutions for real-time object and threat detection and will explore more advanced feature extraction techniques, further optimization methods, and a more robust blockchain framework for decentralized, real-time data sharing in conservation efforts using smart contracts.

References

[1] Saheed, Yakub Kayode. "A binary firefly algorithm based feature selection method on high dimensional intrusion detection data." In Illumination of artificial intelligence in cybersecurity and forensics, Cham: Springer International Publishing, 2022. 273-288.

[2] Anbu, M., and G. S. Anandha Mala. "Feature selection using firefly algorithm in software defect prediction." Cluster Computing 22 (2019): 10925-10934.

[3] Mistry, Kamlesh, Li Zhang, Graham Sexton, Yifeng Zeng, and Mengda He. "Facial expression recongition using firefly-based feature optimization." In 2017 IEEE congress on evolutionary computation (CEC), IEEE, 2017. 1652-1658.

[4] Mizuno, Kosuke, Yosuke Terachi, Kenta Takagi, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto. "Architectural study of HOG feature extraction processor for real-time object detection." In 2012 IEEE Workshop on Signal Processing Systems, IEEE, 2012. 197-202

[5] Putra, FAI Achyunda, Fitri Utaminingrum, and Wayan Firdaus Mahmudy. "HOG feature extraction and KNN classification for detecting vehicle in the highway." IJCCS (Indonesian Journal of Computing and Cybernetics Systems) 14, no. 3 (2020): 231-242.

[6] Davis, Matt, and Ferat Sahin. "HOG feature human detection system." In 2016 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, 2016. 002878-002883.

[7] Venkatesan, Supreeta, A. Jawahar, S. Varsha, and N. Roshne. "Design and implementation of an automated security system using Twilio messaging service." In

2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), IEEE, 2017. 59-63.

[8] Choi, Brendan. "Python Network Automation Labs: Ansible, pyATS, Docker, and the Twilio API." In Introduction to Python Network Automation: The First Journey, Berkeley, CA: Apress, 2021. 675-732.

[9] Abdulnabi, Abrar H., Gang Wang, Jiwen Lu, and Kui Jia. "Multi-task CNN model for attribute prediction." IEEE Transactions on Multimedia 17, no. 11 (2015): 1949-1959.

[10] Rangdale, Sonali, Nihar Ranjan, Prathamesh Ukirde, Shruti Pagade, Shantanu Hanchate, and Nikhil Mehtre. "A Computer Vision-based algorithm for accident detection, severity classification and rapid emergency response using YOLOv8 and Twilio Programmable Messaging API." In 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, IEEE, 2024. 1-6.

[11] Xu, Lily, Shahrzad Gholami, Sara McCarthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh et al. "Stay ahead of Poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations (Short Version)." In 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, 2020. 1898-1901.

[12] Edemacu, Kennedy, Jong Wook Kim, Beakcheol Jang, and Hung Kook Park. "Poacher detection in African game parks and reserves with IoT: Machine learning approach." In 2019 International Conference on Green and Human Information Technology (ICGHIT), IEEE, 2019. 12-17.

[13] Park, Noseong, Edoardo Serra, Tom Snitch, and V. S. Subrahmanian. "APE: a data-driven, behavioral model-based anti-poaching engine." IEEE Transactions on Computational Social Systems 2, no. 2 (2015): 15-37.

[14] Wang, Jian, Xinqi Li, Jiafu Chen, Lihui Zhou, Linyang Guo, Zihao He, Hao Zhou, and Zechen Zhang. "Dph-yolov8: Improved yolov8 based on double prediction heads for the uav image object detection." IEEE Transactions on Geoscience and Remote Sensing (2024).

[15] Sharma, Sangeeta. "Apple Leaf Disease Prediction Using modified YOLOv8 Algorithm." In 2024 International Conference on Integrated Circuits, Communication, and Computing Systems (ICIC3S), vol. 1, IEEE, 2024. 1-6.

[16] Khan, Zahid Farooq, Muhammad Ramzan, Mudassar Raza, Muhammad Attique Khan, Areej Alasiry, Mehrez Marzougui, and Jungpil Shin. "Real-Time Polyp Detection from Endoscopic Images using YOLOv8 with YOLO-Score Metrics for Enhanced Suitability Assessment." IEEE Access (2024).

[17] Rios, Franklin Emanuel Ramirez, and Alicia María Reyes Duke. "Building of a Convolutional Neuronal Network for the prediction of mood states through face recognition based on object detection with YOLOV8 and Python." In 2023 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), IEEE, 2023. 1-6.

[18] Yuvaneswaren, R. S., S. Tarun Prakash, and K. Sudha. "A YOLOv8-based model for precise corrosion segmentation in industrial imagery." In 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), IEEE, 2024. 1-6.

[19] Sharma, Subek, Sisir Dhakal, and Mansi Bhavsar. "Transfer Learning for Wildlife Classification: Evaluating YOLOv8 against DenseNet, ResNet, and VGGNet on a Custom Dataset." Journal of Artificial Intelligence and Capsule Networks 6, no. 4 (2024): 415-435.

[20] https://docs.ultralytics.com/datasets/detect/african-wildlife/

Author’s biography

Dr. C. M. Nalayini is currently working as an Assistant Professor in the department of Information Technology at Velammal Engineering College. Her teaching experience is 19.9 years in total. She has completed her Ph.D degree in Information and Communication Engineering at Anna University, Chennai, India. Her current research interest includes Network Security, Machine Learning, Deep Learning and Blockchain Technologies. She authored books such as Programming in C and Data Structures. She has published many Scopus indexed book chapters, various research articles in International Journals and Conferences. She acted as resource person for various events such as Seminar on Cloud, Workshops on Artificial Intelligence, Machine Learning and Exploring Blockchain and Faculty Development Programs on Blockchain Technologies and Generative AI. She has delivered guest lectures for challenging topics such as Algorithm analysis and XML Databases. She has received best paper awards for her effective research in Blockchain and Machine Learning Domains from International conferences and ISTE Chennai Chapter. She has received International Best Researcher Award for her research article from ISSN AWARDS 2024. She is a recognized reviewer for various International Journals.

Dr.V.Kalpana is presently working as Associate Professor in the Department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, Tamilnadu, India. She received B.E, M.E, Ph.D form Anna University, Chennai. She is having around 13 Years of Experience in Teaching. Her research Interest Includes Mobile Adhoc Networks, Machine Learning, Deep Learning and Data Science. She has published papers in various refereed international journals and conferences. She is an active member in IEEE, ISTE and IAENG.

Dr. S. Hemamalini holds a Doctor of Philosophy (Ph.D.) in Image Processing from Anna University, Chennai, Tamil Nadu, awarded in 2023, and a Master of Engineering (M.E.) in Computer Science and Engineering from Sathyabama University, Chennai, Tamil Nadu, obtained in 2011. With 18 years of teaching experience, she has developed a strong academic foundation and expertise in several advanced domains, including Image Processing, Java Programming, Web Technologies, Internet of Things (IoT), Machine Learning, and Deep Learning. Dr. Hemamalini has authored 17 publications, including research articles and book chapters, in prestigious Scopus and Web of Science (WoS) indexed journals and conferences. Additionally, she has contributed to the development of a copyrighted system titled "Stray Dog Defence Alert & Protection System." Her academic excellence is further exemplified by her recognition as the Course Topper for the NPTEL Online Course "Programming in Java" offered through Swayam Central during the period from July to October 2024.

K. Sathyamoorthy is currently working as Assistant Professor in the Department of Computer Science and Engineering in Panimalar Engineering College, Chennai, Tamil Nadu. He has obtained his Under-Graduation (BE) from Anna University, Post-Graduation (M.Tech) from Sathyabama University and he is perusing his PhD in Vel Tech University, Chennai. He has more than 15 years of experience in the industry and teaching field and Published around 10+ research papers in Impact Factor/Scopus indexed journal. His area of interest includes Data Structures, Computer Networks Data science and Image Processing etc.