International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Automated Leopard Alert And Reporting Mechanism Using Deep Learning

Neha Vora
Faculty of Computer Science
Pacific Academy of Higher Education and Research University
Udaipur, Rajasthan, India
e-mail: nehavora1989@gmail.com

Divya Shekhawat
Faculty of Computer Science
Pacific Academy of Higher Education and Research University
Udaipur, Rajasthan, India
e-mail: divya.shekhawat23@gmail.com

Abstract—Today, rapid infrastructure development is taking place in major metropolitan cities, but unfortunately, this progress often involves the destruction of forest reserves, leaving wild animals homeless. The resulting environmental invasion forces these animals to venture into the cities, posing threats to citizens. In Mumbai, there have been numerous sightings of leopards and other wild animals near forested areas. Leopards have been known to attack street dogs, people, and vehicles, making it necessary to work on this problem. This paper suggests the utilization of deep learning models and object detection techniques to detect leopards and other potential threats. By integrating this technology with security applications, citizens can be made aware of the existence of wild animals in their vicinity. This research primarily focuses on addressing the concern of leopard sightings in Mumbai. The objective is to automate leopard detection and reporting using an object detection algorithm. In the proposed system, images of leopards are collected from an existing dataset available on Roboflow, comprising a total of 1000 samples. The proposed model's performance is evaluated using Mean Average Precision (mAP) & detection speed. The proposed method achieves an impressive mAP of 95.9% at a speed of 37 frames per second.

Keywords- Leopard detection, Deep Learning, Object detection, Roboflow

I. INTRODUCTION

The increase in population demands rapid infrastructure development. A city like Mumbai with limited land mass and high number of immigrations has always demanded construction in every corner. This has led to destruction of forest lands in many ways. The destruction of forest reserves has forced wild animals specially leopards to move towards the city. As per the records of 2017, Mumbai's Sanjay Gandhi National Park has reported around 41 leopards and including the Aarey colony the figure can increase by 51, as reported in a news article by Times of India [1]. Today the number must have significantly grown. The attacks have always been a challenge to the residents near the forest area especially the Aarey colony and around the National Park vicinity. According to a recent news in Times of India, Intekhab Farooqui, a local activist and Shiv Sena Kamgar Sena's Head, said in his interview that: "The victim, Sunita Gurav, was walking outside her house in the night when a leopard leapt towards her and injured the back of her head. Rapid urbanization, tree cuttings and human encroachments inside the green zone of Aarey has led to such leopard attacks". For the same article Wildlife warden Sunish Subramanian said: "Many more hutments and houses have come inside Aarey, which is actually a buffer zone for the adjoining Sanjay Gandhi National Park. Hence, wild species like leopards who do not recognize the boundaries of forest and human habitats, are likely to attack if there is dense human population in forested areas" [2]. Midday reported an incident on 6th November, 2022 at about 8 pm, Ram Yadav aged 61, a farm worker near forest patch, was injured by a leopard. In another incident at the Aarey Milk colony on October 2022, a 16-month-old girl named Eitika Lot was killed in another Leopard attack [3]. According to a report by Times Now news website in 2019, in the early hours of Monday, a leopard attacked two stray dogs close to SEEPZ in Andheri (East), Mumbai. A surveillance camera managed to capture one of the frightening strikes [4].

There have been many such reporting's and leopard sightings from many years. The forest department sets trap camera and trap cages to capture these beasts. But the citizens need more for their safety. Leopards are known to live in hideouts and strike suddenly. Therefore, there is an urgent need for an alarm instantly when a leopard has been sighted. Today, fortunately most buildings and streets are under CCTV surveillance, keeping an eye on everything. The cameras are of high resolution with night vision to monitor every corner. Along with this, Security apps like MyGate, ADD ERP, ApnaComplex and more have replaced the traditional intercom. These Apps notifies the house owner as soon as the visitor passes the society's security. Only when the house owner approves the visit on the app, the visitor is allowed. The house owner can check and approve the visitors remotely from any location provided the user is connected to the internet. We propose a comprehensive object detection model that detects leopards from CCTV footages and integrates the findings with a Security application.

Using computer vision techniques for object detection, the model can identify and detect leopards. This provides incredible provision to citizens to detect leopard threats around them. Single Shot Detector, R-CNN (Regions with Convolutional Neural Networks) and Fast R-CNN, YOLO are some of the several object detection techniques that can be used [5]. In this paper we have used YOLO to detect leopards as its performance is better than many object detection algorithms. Since its debut, the YOLO algorithm has generated remarkable specifications. Its speed and accurateness are better than several top algorithms for object detection. Redmon et. al. [6] unveiled YOLO, an object detection model which can recognize multiple items in a photo while completing all object recognition phases in a single neural network. Through the use of bounding box coordinates & class prospects instead of image pixels, the object detection problem is reframed by YOLO as a single regression problem. The YOLO algorithm can be utilised for a variety of Computer Vision (CV) activities involving animals, Ariel images, the military, autonomous vehicles, sports, hospitals, and others etc. [7]

Over time YOLO has evolved from YOLOv1 to YOLOv8. For object detection in the leopard dataset, a deep learning approach is applied. It is able to acquire & retrieve the characteristics learned directly from the inputs, as opposed to prior machine learning methods that needed custom defining the features to be recovered from the inputs. The newest YOLOv8 algorithm is used in this research.

II. LITERATURE REVIEW

Automatic wildlife surveying and animal monitoring have become essential tools for conservation efforts and wildlife management. In recent years, CV techniques have played an important role in addressing the challenges associated with aerial video animal detection, especially in complex natural environments. This review of literature explores various studies and developments in the area of animal detection & segmentation by means of deep learning-based methods, with a focus on CNN and the popular “You Only Look Once” (YOLO) object detection algorithms.

Animal detection in aerial videos presents unique challenges due to the complexity of natural surroundings. One suggested method involves using global patterns of pixel motion, estimated through optical flow methods, to detect moving animals against the background [8]. This research has practical applications, such as observing locomotive behaviour to prevent animal disruptions in residential areas [9] [10].

The majority of deep learning-based animal detection systems are dominated by CNNs. Network’s depth is determined by the number of layers in its architecture, as deep learning requires neural networks with numerous layers. CNNs are a type of feed-forward neural network that comprises three main layers: convolutional layers, pooling layers, and fully connected layers [11].

Norouzzadeh et al. conducted research using deep neural networks and achieved automatic animal detection with an impressive accuracy of over 93.8% [12]. The key role of convolutional layers in this process is to create feature maps, acting as automatic feature extractors. Convolutional layer’s output is downsampled by pooling layers. Finally, neurons from the input feature maps are joined to the internal neurons in the fully connected layer, enabling the network to make accurate predictions.

In another study by Saleh et al., the researchers frequently utilized transfer learning to create CNNs that are more accurate while using fewer resources [13]. It is a technique where a new categorization model is built using previously learned weights from a base model. This approach reduces training time and resource usage while achieving improved accuracy levels and requiring less data for training.Overall, CNNs have proven to be a potent tool for animal detection, & researchers have explored various approaches, together with transfer learning, to improve their efficiency and accuracy in detecting animals, including in challenging environments like urban and highway traffic scenarios.

In a study [14], transfer learning was employed using VGG-16 as the base model. VGG-16, with 33 stacked convolutional layers, achieved an impressive top-5 test accuracy of 92.7% on the ImageNet dataset, which consists of over 14 million images from 1000 classes. Maxpooling layers with 2x2 filters were used to reduce the volume size. The classification process involved a SoftMax classifier with 1000 channels for each class, followed by two fully linked layers, each with 4,096 nodes [11].

Meenatchi K et al [15] developed a Wild Animal Detection System using Artificial Intelligence on a Raspberry Pi. The system locally identifies wild animals in photographs. If a dangerous animal is detected, it employs a GSM module to transmit a message and emits an ultrasonic buzzer to scare away the animal. This comprehensive solution aims to mitigate harm to both people and animals, as well as protect valuable resources, through the deployment of a deep convolutional neural network-based animal detection system.

In their research on "Real-time Animal Detection and Prevention System for Crop Fields," R Lathesparan et al [16] utilized two Convolutional Neural Network (CNN) classification models combined and tested on a Raspberry Pi as the processing unit. The system captures pictures of animals using a thermal sensor on Arduino, which triggers a picture whenever an animal is detected. The researchers incorporated sudden light flashes, ultrasonography, and bee sounds to scare away the animals. The categorization model achieved an accuracy rate of 77 percent.

Verma et al. [24] worked with a camera-trap database, utilizing candidate animal proposals to develop a verification step that determines whether a specific patch in the image is background or an animal. Their results on a conventional camera-trap dataset revealed an accuracy of 91.4%, showcasing the effectiveness of their approach in animal detection.

Dhillon et al. [25] proposed a deep learning-based system for detecting wild animals from highly cluttered natural forest images. Their detection method using Residual Network (ResNet) outperformed existing systems. In the context of deep learning techniques for animal recognition, segmentation, and detection, a survey was conducted to provide a concise analysis and comparison of various approaches [26].

In another study [27], the focus was on animal object detection and segmentation in wildlife monitoring videos captured by camera-traps. The experimental results demonstrated that their framework achieved superior animal object detection accuracy, surpassing state-of-the-art approaches, including Faster-RCNN, by up to 4.5%.

The rapid advancement of computer vision has led to neural networks dominating object detection techniques. Deep learning-based methods initially started with text classification and then evolved to identify human behaviour, ranging from simple tasks to complex group activities. The YOLO object identification algorithms, particularly between versions 1 and 8, have outperformed traditional algorithms in terms of capability and performance.

Addressing the challenge of face detection in complex and real-world environments, a customized version of the YOLOv3 object detector was proposed in a paper [28]. This specialized model was designed for accurate and fast face detection in surveillance and biometrics scenarios.
Additionally, researchers in the field of industrial automation focused on automating the monitoring, categorization, and segregation of industrial gears in assembly lines. Three object identification models—Faster RCNN, YOLO, and SSD—were evaluated, and YOLOv4 was chosen as the preferred model due to its optimal trade-off between accuracy and detection speed [30].

A novel variation of the YOLO family called YOLOv5 was introduced by a researcher named Glenn and his team. This version, as highlighted by Nepal and Eslamiat [17], utilizes PyTorch instead of Darknet and incorporates CSPDarknet53 as its structural support. YOLOv5 resolves the issue of repetitive gradient information found in YOLOv3 and YOLOv4.

Another breakthrough in real-time object detection is YOLOv7, which has been revolutionizing the computer vision field. This model was trained using only the MS COCO dataset [18]. YOLOv7 offers remarkable features and improvements, making it ground-breaking in the computer vision industry.

The YOLOv7 model has shown extraordinary progresses over its former versions on Graphics Processing Units (GPU) V100. It outperforms all former models by attaining a speed of 30 FPS or more while sustaining high accuracy levels. YOLOv7 has achieved the highest mean Average Precision (mAP) of 56.8%. Notably, YOLOv7 succeeded to reduce approximately 40% of parameters and 50% of computation, leading to amplified accuracy and decreased inference cost in real-time object detection. The model also demonstrated faster inference speed and better detection accuracy.

Terven et al. [19] provided an overview of the YOLO series, starting from its commencement and covering its evolution until YOLOv8, which was released in January 2023 by Ultralytics. YOLOv8 has made some notable improvements, including eliminating anchors, resulting in fewer box predictions and faster Non-maximum Suppression (NMS). During training, YOLOv8 utilizes mosaic augmentation, but this augmentation is disabled for the final ten epochs due to potential harmful effects when used continuously. YOLOv8 can be installed as a PIP package or executed through the Command Line Interface. The model offers five scaled variants: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x.

Patel et al. [20] conducted a study on object detection in hockey sports using YOLOv3, achieving a high accuracy of 91.3%. The model demonstrated the potential of YOLOv3 in sports analytics, especially in the detection and tracking of objects in dynamic environments like hockey games.
Another advancement in object detection is the defect detection model built on YOLOv5 by Yao et al. in 2021. This model excels in accurately and swiftly detecting faults. To enhance its ability to identify minute flaws, a small object identification layer was added. Additionally, the model incorporated the layer of squeeze-and-excitation and full Intersection over Union (IoU) of the loss function to improve regression accuracy. The Cosine Annealing algorithm was employed to enhance the model's performance after being trained using transfer learning. YOLOv5 achieved an mAP@0.5 of 94.7%, representing an approximate 9% increase compared to the original method.

Liu et al. [21] conducted a study on railway signal lights detection and found YOLOv5 to be highly useful in their experiment. The model was trained on a subway scenes dataset with signal lights and achieved a remarkable running speed of 100 FPS. Furthermore, the model exhibited an average recall rate and accuracy of 97.2%, highlighting its efficacy in detecting and recognizing railway signal lights for safety and operational purposes.

Hao et al. [22] proposed a lightweight method to enhance the speed and accuracy of YOLOv5. Their modified model, called Light-YOLOv5, was tested on a dataset that included fire scenario examples. The results demonstrated that Light-YOLOv5 achieved an impressive Frames Per Second (FPS) of 91.1 and increased the mAP by 0.033. Notably, the mAP of the modified model was 6.8% higher than YOLOv7-tiny, showcasing the efficiency of the algorithm in real-time object detection tasks.

To address operational lag, inspection and certification expenses, and undetected harm from human error, Hussain et al. [23] introduced a framework built using YOLOv7. The authors suggested a domain variance modelling approach to overcome the problem of data scarcity by generating representative data samples. The proposed framework's mAP is 91.1%, indicating its effectiveness in resolving the identified issues.

In their research, a proposed model called RescueNet utilized deep learning to identify survivors. The implementation results demonstrated high precision, recall, F1-score, and mAP, indicating the model's effectiveness in detecting and counting affected individuals and animals. The RescueNet model has significant potential to aid rescue systems in providing timely responses and saving lives during flood disasters [29].

As for YOLOv8, a current version of the YOLO variant, authors who tested its performance against other variations observed a substantial improvement in performance. An experiment was conducted to analyse and validate the performance of YOLOv8, suggesting its promising capabilities [28].

These studies collectively highlight the continuous advancements and innovations in YOLO-based models to address various challenges in object detection, real-time processing, and rescue operations. The improvements in speed, accuracy, and efficiency hold great promise for future applications in diverse fields.

III. METHODOLOGY

A. Experiment Setup

The proposed architecture for leopard detection using YOLOv8 involves utilizing the YOLO variant to identify and locate leopards in different images. YOLOv8 is an innovatory object detection model that has been improved to achieve higher accuracy and efficiency in detecting objects. The proposed architecture comprises of multiple layers of convolutional neural networks to process the input set of images and grab relevant features for detection.

The dataset used for training, validation, and testing encompasses an assorted set of pictures containing leopards in various postures, backgrounds, and lighting conditions to represent a real world set up. Annotated bounding boxes are provided for the detected images, suggestive of the exact location of the leopards in each image. To train the YOLOv8 model to precisely identify leopards from the CCTV Frames this annotated dataset is crucial for training the model.

The experiment was carried out with Google Colab, a platform which provides free coding notebooks, cloud virtual machines, a GPU, and a Tensor Processing Unit (TPU) for carrying out time- consuming and difficult computations. The model was trained, validated and tested using Google Colab. All the results were saved on Google Drive and are accessible for further use. The platform is built on Windows and has access to all of a physical computer's resources (Windows OS). Its user can access Google Drive for saving and importing files. The model's performance was assessed using a variety of assessment metrics.

B. Dataset Preparation

This experiment made use of the Roboflow Public Dataset (Roboflow, n.d.).1000 photos in all were obtained from the Roboflow Public Dataset. 1000 photos with 1045 leopard class annotations were included in the training dataset. On the basis of the ratio 70:20:10 of the number of annotated photos, the dataset was divided into training, validation, and testing. 701 photos, or 70% of the dataset, were used for training, 200 images, or 20% of all images, were used for validation, and 99 images, or 10% of all remaining images, were used for testing.

C. Proposed Architecture

The proposed system is designed as a two-module architecture: Leopard detection and Alarm. The main objective of the system is to detect leopards in real-time footage captured by CCTV cameras and raise an alarm through an associated Security mobile application.
The system operates as follows:

1) Leopard Detection Module:

- Real-time footages from CCTV cameras are fed into the system.
- The footages are sent to a Cloud-based model for pre-processing and object detection.
- The Cloud-based model, which is precisely trained for leopard detection using YOLOv8 identifies the presence of leopards.
- Once a leopard is detected in the footage, the model sends an alert or trigger the Alarm module.

2) Alarm Module:

- The Alarm module is associated with a Security mobile application.
- When the Leopard Detection module detects a leopard in any CCTV camera footage, it instantly raises an alarm.
- The alarm signal is sent to the associated Security mobile application in real-time.
- The Security mobile application receives the alert and notifies the relevant security personnel or stakeholders about the leopard sighting.

By combining real-time footage analysis, Cloud-based object detection, and immediate alarm notification, the proposed system provides a robust solution for alerting security teams to the presence of leopards in the monitored areas. This can significantly enhance wildlife safety and minimize potential risks to both humans and animals.

It's vital to ensure that the system is continuously monitored & optimized to maintain its effectiveness in real-world scenarios. Regular model updates, dataset enrichment, and performance evaluation are essential for improving the system's accuracy and responsiveness in detecting leopards and preventing false alarms. Additionally, considerations for privacy and data security should be addressed to ensure compliance with relevant regulations and protect the privacy of individuals within the monitored areas. The proposed architecture is displayed in figure1

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

D. Performance metrics

Using the created dataset Precision, Recall, and Average Precision (AP) were computed and contrasted to assess model's performance. The following formulae were used to compute the metrics.

Precision(P): By dividing the number of correctly identified leopards by the total number of leopards discovered, the classification accuracy is determined. It can be written as

$$Precision = \frac{True\ Positive}{(True\ Positive + True\ Negative)}$$

Recall(R): In a dataset, it is the ratio of precisely observed leopards to the total number of leopards.

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)}$$

F1-score: It is used to achieve a balance between precision and recall. The harmonic of recall and precision is how the F1-score is represented. Compared to accuracy metrics, it is considered to be a more accurate measure.

$$F1 - score = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right)^{-1}$$

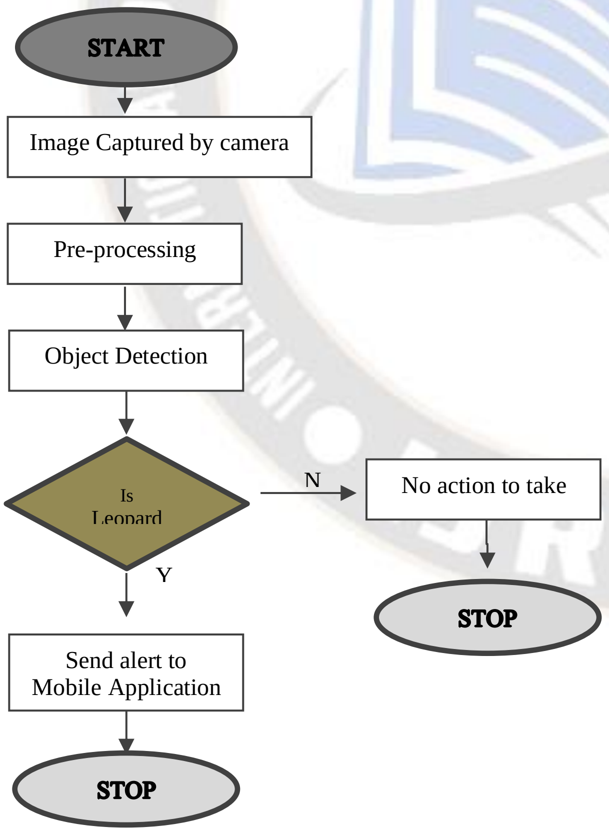

Figure 1. Proposed Architecture

True positives are positive samples classified correctly, false positives are negative samples classified incorrectly, and false negatives are positive samples classified incorrectly.

MAP and AP: The average precision for each class is known as MAP. At different thresholds. the AP provides total weighted precision. Either the mean average of all classes or the sum of all IOU criteria are used to determine the MAP value.

FPS:- Number of images the model processes per second is frames per second i.e. FPS. It is a very important speed performance parameter.

IV. RESULTS

The experiment involved training the YOLOv8 model on different images of leopards. Table I displays the parameters used during the training of the neural network.

The gradient descent optimization algorithm plays a crucial role in adjusting the model's weights in relation to the loss function. The learning rate parameter determines how much the model's weights are modified in each iteration. A higher learning rate may result in faster convergence, but it could also lead to overshooting the optimal solution. On the other hand, a lower learning rate may lead to slow convergence. Finding an appropriate learning rate is essential for optimizing the model's performance.

Stochastic Gradient Descent (SGD) is employed as the optimization algorithm. It accumulates the gradients from earlier stages and determines the direction in which the weights should be adjusted. This allows the algorithm to efficiently optimize the model's parameters.

Due to the large volume of data used in the experiment, the concept of batch size is utilized. The batch size refers to the number of samples the algorithm uses in each iteration to train the network. By processing data in batches, the training process becomes handier and technically efficient.

The number of epochs represents the total number of times the entire dataset undergoes training. Each epoch involves passing the entire dataset through the network. Training for multiple epochs allows the model to learn from the data multiple times, which can improve its ability to generalize to unseen examples.

In YOLOv8, lr0 and lrf are used to control the learning rate schedule during training. lr0 denotes the initial learning rate, and lrf represents the final learning rate at the last epoch of training. By using a learning rate schedule, the model can adaptively adjust the learning rate during training, which may lead to better convergence and overall performance.

TABLE I. TRAINING PARAMETERS ON YOLO FOR LEOPARD DETECTION

Parameters	Value
lr0	*0.01*
Lrf	*0.01*
Momentum	*0.937*
Weight decay	*0.0010078125*
Optimizer	*SGD (lr=0.01)*
Epochs	*100*
Batch size	*16*
Image size	*640*

The default values for both lr0 and lrf in YOLOv8 are set to 0.01. During training, this value is used to regulate the learning rate. Fine-tuning these parameters and experimenting with different learning rate schedules can help optimize the model's performance in leopard threat detection.

Overall, the experiment's results and the discussion of training parameters provide valuable insights into the effectiveness and efficiency of the proposed system in alerting users of leopard threat detection through the security application.

Table II displays the experimental result of our model on our dataset. It was determined that 100 epochs would be needed to train the model. The model that was trained showed promising performance.

TABLE II. RESULTS OF YOLOV8 MODEL

Precision	*0.951*
Recall	*0.915*
F1 Score	*0.932*
mAP@0.5	*0.959*
mAP@0.5:.95	*0.597*
Weight	*136.7MB*
No. of Parameters	*68124531*

The mAP@0.5 value of YOLOv5 is 0.959 with a precision of 0.951& a recall of 0.915. The Precision-Confidence curve, Recall -Confidence curve & Confusion matrix is displayed in figure 2,3 & 4 respectively.

Mean Average Precision (mAP) and detection speed are the criteria taken into consideration while assessing the suggested model's performance.

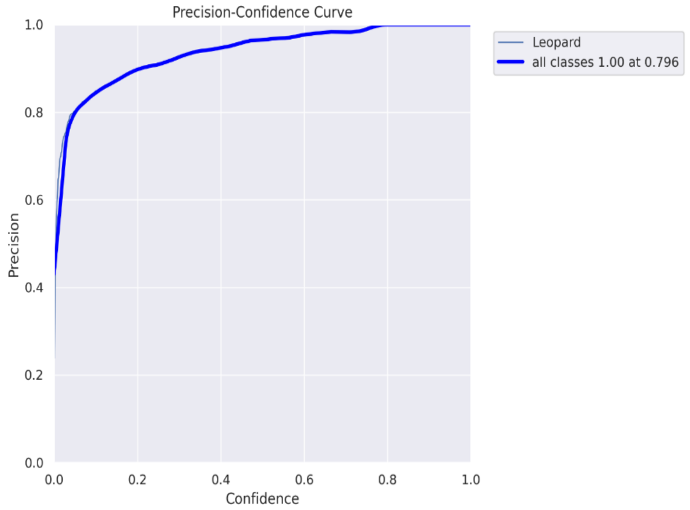

Figure 2- Precision-Confidence Curve

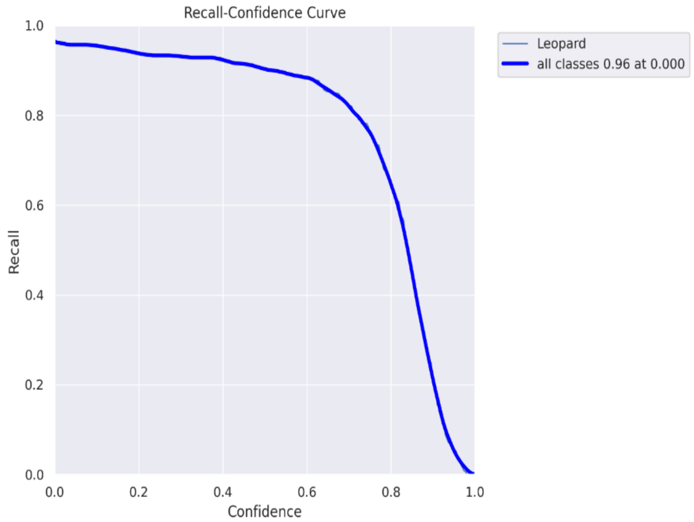

Figure 3- Recall-Confidence Curve

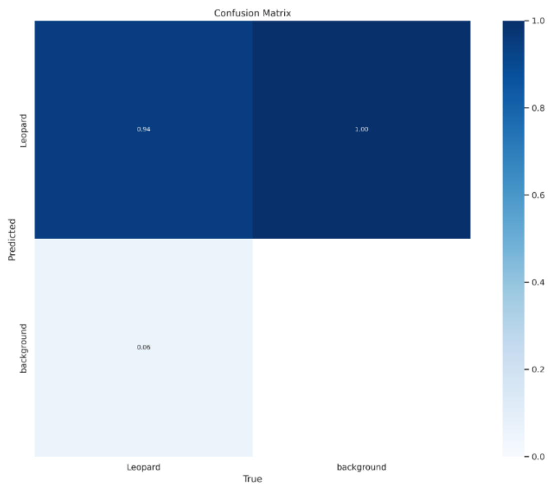

Figure 4- Confusion matrix

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

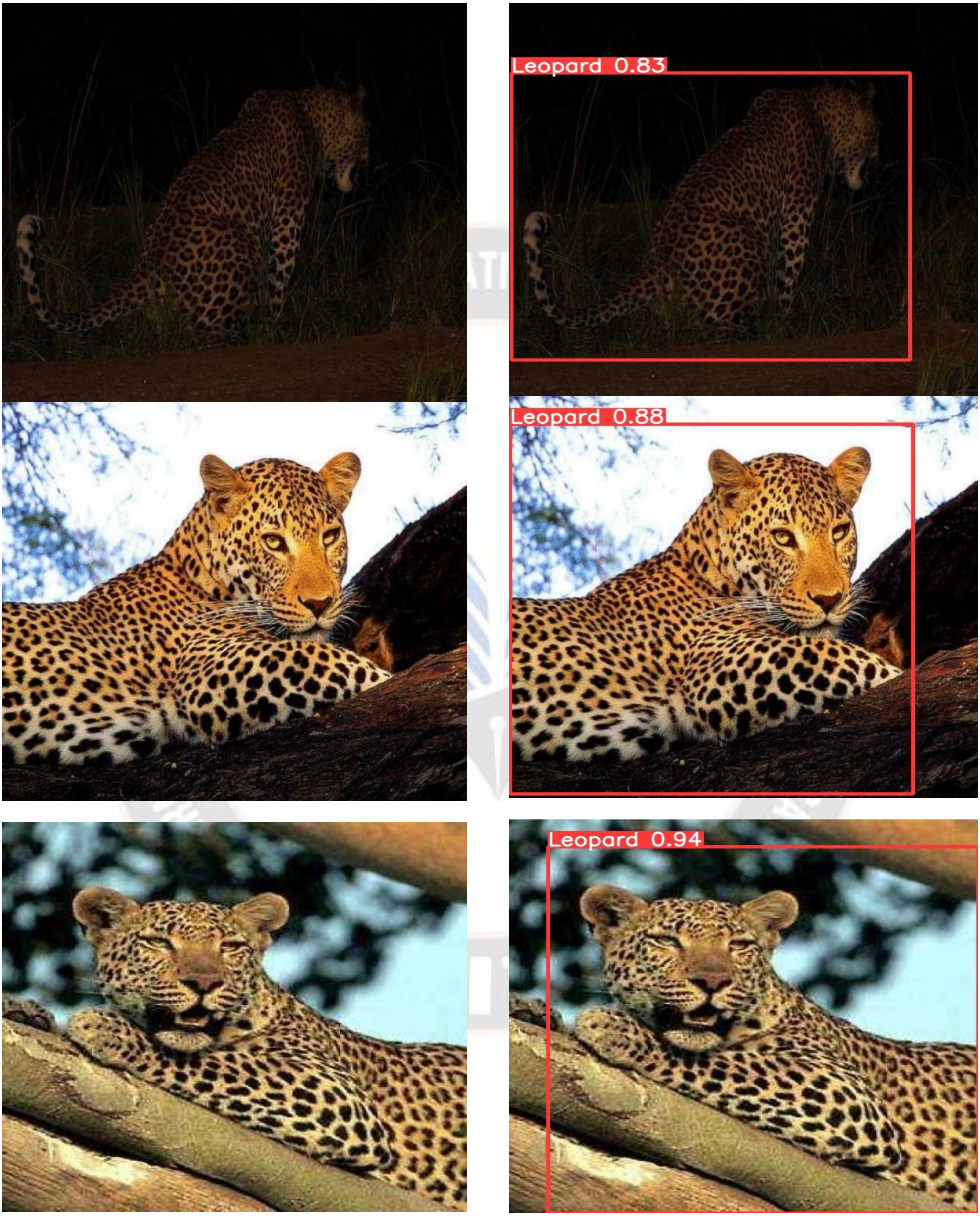

Figure 5- Testing results of the proposed system

The mAP and detection speed of the proposed method are 95.9% at 37 fps. The proposed model has achieved an accuracy of 0. 959. It is found that the YOLOv8 model performs good for all the performance metrics. The Testing results of the proposed system are displayed in figure 5.

V. CONCLUSION

This paper presents the most recent version of YOLO for detecting leopards in residential areas and cautioning users via a security application. The model has been precisely intended to attain high accuracy, even during night-time. The results indicate that the model's accuracy in leopard detection exceeds 95%, which is highly effective for the proposed application.

In addition to accuracy, the inference speed of the algorithm was also a crucial metric considered in the study. The model demonstrated a notable inference speed of more than 35 frames per second (FPS), allowing for real-time detection and rapid alerts to ensure timely responses to potential leopard sightings.

It is noteworthy that while there are some examples of deep learning algorithms being applied to leopard detection in existing works, this paper represents a revolutionary effort. To the best of the authors' knowledge, this is the first of its kind study to influence deep learning techniques for automated leopard detection and providing real-time information to people. This innovative application of deep learning technology marks a critical need in enhancing wildlife safety and justifying potential risks to humans and animals in populated areas.

Moreover, the paper intends to contribute to the field of wildlife detection using deep learning and inspires further research and application of similar algorithms for detecting other wild species beyond leopards.

The successful implementation and high accuracy of the proposed model for leopard detection open up possibilities for its application in other contexts and scenarios. By sharing their findings, the authors hope to inspire further studies and innovations in the realm of wildlife conservation and safety using cutting-edge deep learning algorithms.

This research presents an Automated Leopard Alert and Reporting Mechanism that addresses the rising concern of leopard sightings and potential attacks in metropolitan cities like Mumbai. The rapid urban development and destruction of natural environments have forced leopards and other wildlife to move closer to human settlements, posing a threat to public safety.

The proposed solution utilizes deep learning-based object detection, employing the YOLO algorithm, to detect leopards from CCTV footage. The model achieved a notable accuracy of 95.9% and a high detection speed of 37 frames per second. By integrating these detection capabilities with existing security applications and surveillance systems, real-time alerts can be provided to citizens about leopard presence in their vicinity. This sanctions residents to take necessary defenses and authorities to respond promptly, minimizing potential conflicts and ensuring the safety of both humans and leopards.

The application of computer vision techniques and deep learning models, mainly YOLO, exhibits the potential of object detection in addressing wildlife threats. The elasticity of object detection technology encompasses beyond leopard detection, offering applications in multiple arenas, including aerial image analysis, military operations, self-governing vehicles, and more.

The Automated Leopard Alert and Reporting Mechanism presents a practical and hands-on approach concerning leopard sightings in residential areas. By using deep learning and along with existing smart security systems, it offers an effective solution for safeguarding public safety.

Future work demands expanding the dataset used for training, combining a vivid and challenging image conditions, such as rain and hostile weather. Training the model on a larger and more wide-ranging dataset will improve its robustness in real-world scenarios. Moreover, continuous research and implementation of automated alert systems can contribute to fostering harmonious coexistence between humans and wildlife in the face of urbanization.

In conclusion, the Automated Leopard Alert and Reporting Mechanism provides a valuable contribution to wildlife conservation efforts and public safety in urban areas. By utilizing advanced technologies and embracing further research, we can continue to develop innovative solutions that protect both human communities and wildlife in rapidly evolving urban landscapes.

REFERENCES

[1] https://timesofindia.indiatimes.com/city/mumbai/sgnp-census-confirms-41-leopards-27-of-them-new/articleshow/62812269.cms
[2] https://timesofindia.indiatimes.com/city/mumbai/mumbai-woman-hospitalised-after-leopard-attacks-her-at-aarey-colony/articleshow/95459869.cms?from=mdr
[3] https://www.mid-day.com/mumbai/mumbai-news/article/mumbai-leopard-attacks-38-year-old-woman-in-aarey-colony-23255164
[4] https://www.timesnownews.com/the-buzz/article/video-leopard-attacks-two-stray-dogs-in-mumbai-watch-chilling-footage/525364
[5] Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics, 10(3), 279.
[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
[7] Gorriz, J. M., Ramirez, J., Ortiz, A., Martinez-Murcia, F. J., Segovia, F., Suckling, J. & Ferrandez, J. M. (2020). Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. Neurocomputing, 410, 237-270.
[8] Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., ... & Phung, D. (2017, October). Animal recognition and identification with deep convolution neural networks for automated Animal monitoring. In Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on (pp. 40- 49). IEEE.
[9] Kumar. S., & Singh, S. K. (2016). Monitoring of pet animal in smart cities using animal biometrics. Future Generation Computer Systems.
[10] Xue, C., Wang, P., Zhao, J., Xu, A., & Guan, F. (2017). Development and validation of a universal primer pair for the simultaneous detection of eight animal species. Food chemistry, 221, 790-796.
[11] Karen Simonyan, A. Z. (2015) 'very deep convolutional networks for large-scale image recognition', pp. 1–14.
[12] Norouzzadeh, M. S. et al. (2017) 'Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning', pp. 1– 17. Available at: http://arxiv.org/abs/1703.058 30.
[13] Saleh, K., Hossny, M. and Nahavandi, S. (2018) 'Effective vehicle-based kangaroo detection for collision warning systems using region-based convolutional networks', Sensors (Switzerland), 18(6). doi: 10.3390/s18061913.
[14] Willi, M. et al. (2019) 'Identifying animal species in camera trap images using deep learning and citizen science', Methods in Ecology and Evolution, 10(1), pp. 80–91. doi: 10.1111/2041210X.13099.
[15] Wild Animal Detection System Using Deep Convolutional Neural Networks Meenatchi K1 , Thibishini V International Journal of Scientific Research in Science and Technology Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com) doi : https://doi.org/10.32628/IJSRST 1 , Vaisnavi K1 , Mrs. R. Ahila 2
[16] Real-time Animal Detection and Prevention System for Crop Fields R Lathesparan#, A Sharanjah, R Thushanth, S Kenurshan, MNM Nifras, and WU Wickramaarachi, 13th International Research Conference General Sir John Kotelawala Defence University
[17] Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors, 22(2), 464
[18] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.
[19] Terven et al.(2023). A comprehensive review of yolo: from yolov1 toYolov8 and beyond arXiv:2304.00501v1
[20] Patel et al(2023).Object detection in hockey sport video via pretrained yolov3 based deep learning model. ICTACT Journal on Image and Video Processing, February 2023, Volume: 13, Issue: 03
[21] Liu, W., Wang, Z., Zhou, B., Yang, S., & Gong, Z. (2021, May). Real-time signal light detection based on yolov5 for railway. In IOP Conference Series: Earth and Environmental Science (Vol. 769, No. 4, p. 042069). IOP Publishing.
[22] Hao, X., Bo, L., & Fei, Z. (2021). Light-YOLOv5: A Lightweight Algorithm for Improved YOLOv5 in Complex Fire Scenarios.
[23] Hussain, M., Al-Aqrabi, H., Munawar, M., Hill, R., & Alsboui, T., (2022). Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections. Sensors, 22, 6927.
[24] Verma, G.K., Gupta, P. (2018). Wild Animal Detection Using Deep Convolutional Neural Network. In: Chaudhuri, B., Kankanhalli, M., Raman, B. (eds) Proceedings of 2nd International Conference on Computer Vision & Image Processing . Advances in Intelligent Systems and Computing, vol 704. Springer, Singapore. https://doi.org/10.1007/978-981-10-7898-9_27
[25] Dhillon, A., Verma, G.K. (2018). Wild Animal Detection from Highly Cluttered Forest Images Using Deep Residual Networks. In: Tiwary, U. (eds) Intelligent Human Computer Interaction. IHCI 2018. Lecture Notes in Computer Science(), vol 11278. Springer, Cham. https://doi.org/10.1007/978-3-030-04021-5_21
[26] V. Palanisamy and N. Ratnarajah, "Detection of Wildlife Animals using Deep Learning Approaches: A Systematic Review," 2021 21st International Conference on Advances in ICT for Emerging Regions (ICter), Colombo, Sri Lanka, 2021, pp. 153-158, doi: 10.1109/ICter53630.2021.9774826.
[27] Z. Zhang, Z. He, G. Cao and W. Cao, "Animal Detection From Highly Cluttered Natural Scenes Using Spatiotemporal Object Region Proposals and Patch Verification," in IEEE

Transactions on Multimedia, vol. 18, no. 10, pp. 2079-2092, Oct. 2016, doi: 10.1109/TMM.2016.2594138.

[28] Silva, L.P.e., Batista, J.C., Bellon, O.R.P., Silva, L. (2019). YOLO-FD: YOLO for Face Detection. In: Nyström, I., Hernández Heredia, Y., Milián Núñez, V. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2019. Lecture Notes in Computer Science(), vol 11896. Springer, Cham. https://doi.org/10.1007/978-3-030-33904-3_19

[29] Prabhu, B.V.B., Lakshmi, R., Ankitha, R. et al. RescueNet: YOLO-based object detection model for detection and counting of flood survivors. Model. Earth Syst. Environ. 8, 4509–4516 (2022). https://doi.org/10.1007/s40808-022-01414-6

[30] Sureshkumar, S., Mathan, G., RI, P. et al. Deep learning framework for component identification. Int. j. inf. tecnol. 14, 3301–3309 (2022). https://doi.org/10.1007/s41870-022-00895-z