Wildlife Animal Classification and Alert System

Pannaga P Raj [1], *Sanjana S R*[2], *Shalini P Tengil* [3], *Swarna B O* [4], *Sayed Aftab Ahamed* [5]

[5]Assistant Professor, Department of IS&E, JNNCE, Shivamogga, India

[1,2,3,4] Student, Department of IS&E, JNNCE, Shivamogga, India

ABSTRACT- *Animals in the wild are in grave danger due to ongoing hunting and poaching. Also due to the loss of wild habitat, animals are moving out of forests and into the cities and hence posing a danger to human lives. As a result, it is critical to monitor the movement of wild animals and take appropriate measures, particularly along forest highways or pathways. This project mainly involves capturing the animals in live, detecting them, classifying them and then sending an alert message to the respected authorities. This model detects animals using a Convolution Neural Network (CNN), which is complemented by a real-time item detection method known as You Only Look Once (YOLO).*

Index Terms: *Wild animals, CNN, YOLO.*

I. INTRODUCTION

Human-wildlife conflict has escalated into a pressing global issue, driven by habitat loss, urbanization, and environmental degradation. The Wild Animal Classification and Alert System meets this vital need by using innovative technology to identify, categorize, and monitor wildlife activities in real time, allowing for proactive intervention and promoting coexistence. The primary goal of this system is to reduce human-wildlife conflict by issuing timely alerts to possible hazards. This is accomplished through the sophisticated integration of camera traps, real-time video feeds, and cutting-edge machine learning (ML) techniques such as Convolutional Neural Networks (CNN) and YOLO (You Only Look Once). These technologies enable the system to reliably recognize and classify animal species, differentiate between threat and non-threat circumstances, and swiftly notify appropriate authorities and local residents. This proactive method seeks to avoid potentially risky contacts, so reducing injury to both humans and animals.

Traditional methods of wildlife monitoring often rely on manual observation or tracking, which are labour-intensive, time-consuming, and limited in scope. The Wild Animal Classification and Alert System offers a significant advancement by automating the monitoring process and providing real-time information. Camera traps strategically placed in wildlife corridors and near human settlements capture images and videos of animal activity. These data are then sent into the system's advanced machine learning algorithms for processing. CNNs is a powerful class of deep learning models in which it plays an important role in image recognition and classification. These networks are especially good at extracting complex traits and patterns from visual data, allowing the system to properly identify distinct animal species even in difficult surroundings with changing lighting, weather conditions, and background clutter. The system pre-processes images by converting them to grayscale, segmenting them, and masking irrelevant portions, further enhancing the performance of the CNN. This pre-processing enhances the focus of the model on relevant features and improves accuracy.

The algorithm is employed for object detection. YOLO's unique approach of processing entire images at once makes it exceptionally for real-time applications. It excels in detecting the location and limits of items in an image, displaying bounding boxes around discovered creatures. This combination of CNN for classification and YOLO for detection assures accuracy and speed, which is critical for analysing live video streams and generating timely alerts. This combined approach ensures both accuracy in identifying the species and speed in processing the video feed. The system's alert mechanism is designed detecting a potential threat, the system delivers alerts via multiple communication channels, including SMS messages, email notifications, and integrated mobile applications. This multi-channel approach ensures that alerts reach the intended recipients promptly, enabling swift action to mitigate potential conflicts. These notifications can be delivered to forest departments, local guards, and community people, as well as trigger automated warnings like traffic lights to prevent approaching cars. Researchers can analyse animal migration patterns, habitat utilization, and population dynamics to gain a better knowledge of wildlife ecology. This knowledge can help to shape conservation strategies and management plans, ensuring threatened species Furthermore, the system can assist farmers in protecting their crops from wildlife damage without resorting to harmful methods, such as lethal traps or poisons. Forest officers can also utilize the system to monitor sensitive areas, combat poaching, and prevent other illegal activities.

These notifications can be delivered to forest departments, local guards, and community people, as well as trigger automated warnings like traffic lights to prevent approaching cars. The system's data collection capabilities can be extremely useful for wildlife research and conservation. Researchers can analyse animal migration patterns, habitat utilization, and population dynamics to gain a better knowledge of wildlife ecology. In conclusion, the Wild Animal Classification and Alert System represents a significant advancement in human-wildlife conflict management. By merging cutting-edge technologies with a strong emphasis on ethical and environmental concerns, the system provides a comprehensive solution that emphasizes both human safety and wildlife protection. This approach not only tackles acute conflict situations, but it also provides essential data for long-term study and conservation activities, fostering cooperation between humans and wildlife and helping to preserve our planet's biodiversity.

II. PROBLEM DESCRIPTION

The expansion of human activities into natural areas has resulted in increased human-animal conflicts, posing important issues such as crop devastation, human injuries, and wildlife injury. Traditional methods of conflict mitigation, like fencing and patrolling, are reactive, inefficient, and costly. Current wildlife detection systems often lack accuracy, struggle in varying environmental conditions, and fail to provide real-time alerts. There is an urgent need for an advanced system capable of detecting, classifying, and alerting stakeholders in real time. The wild animal classification and alert system overcome these challenges using Convolutional neural network and YOLO for efficient and accurate monitoring. Human-wildlife conflicts are escalating due to habitat loss, deforestation, and urbanization, which force animals to stray into human-dominated landscapes. These interactions lead to various problems, including the destruction of crops by animals like elephants and wild boars, human injuries caused by predators like tigers and leopards, and harm to wildlife due to retaliatory actions. Additionally, accidents on highways and railways near forests further endanger both humans and animals. Existing measures such as physical barriers, manual patrolling, and deterrents are often inadequate, reactive, and unsustainable. Furthermore, traditional wildlife detection systems frequently suffer from low accuracy, computational inefficiency, and poor performance in challenging environmental conditions like low light or dense vegetation.

The Wild Animal Classification and Alert System provides a proactive solution by leveraging machine learning technologies, specifically CNN and YOLO, for real-time wildlife monitoring. The system captures images or video feeds from camera traps, processes them to detect and classify animals, and sends real-time alerts to stakeholders about potential threats. This ensures timely interventions to prevent conflicts while minimizing harm to wildlife. By combining superior detection capabilities with real-time response mechanisms, the system efficiently handles the critical issues of safety, conservation, and sustainability.

III. LITERATURE REVIEW

Camera traps, which include a motion sensor and a camera, are commonly employed to detect animals. When motion is detected, the sensor prompts the camera to take pictures. By analysing these images, species and behaviours of wild animals can be identified. Trail cameras, a common type of camera trap, come in various models with different features. The sensitivity of motion sensors, image resolution, and transmission frequency can all be adjusted to improve battery life on these devices. Several studies have recently investigated the use of deep learning (DL) to automate wild animal detection and classification, a procedure that would otherwise require time-consuming manual examination of big datasets from video traps. These works can be divided into two categories: the construction of DL models for detection and classification [1]-[7], and the creation of devices that use DL for wild animal detection [8]-[12]. Below are relevant studies that techniques and methods that are directly or indirectly aligned with the objectives of this project at the study "Comparison of Deep Learning Techniques for Classification of Insects in Order Level with Mobile Software Application" [1], researchers assessed four deep learning models faster R-CNN, Inception V3, SSD MobileNet, and YOLOv4—for insect classification at the order level. The study stressed the significance of feature extraction and preprocessing in difficult classification problems. The findings emphasize the need of adopting the appropriate architecture based on task-specific requirements, with broad implications for animal classification systems.

"An Automated Vertebrate Animals Classification Using Deep Convolution Neural Networks" [2], goes into detail about CNN layers for vertebrate classification, including convolutional, pooling, ReLU, and fully connected layers. This foundational study establishes how CNNs extract image features and supports the proposed project's reliance on CNN-based architectures. In "Deep Learning and Computer Vision-Based Framework for Himalayan Bear, Marco Polo Sheep, and Snow Leopard Detection" [3], a comparative evaluation of AlexNet, ResNet-50, VGG-19, and Inception V3 demonstrated that Inception V3 with kNN achieved high accuracy (98.3%). This validates Inception V3's utility in wildlife classification and reinforces its potential application in this project. The study in "Animal Recognition System Based on Convolutional Neural Network" [4], highlights the impact of training data size on classification accuracy. It emphasizes the necessity for diverse datasets and explores the effectiveness of CNNs over traditional methods like PCA, LDA, and LBPH, supporting the choice of CNNs for this project.

Although focusing on fence removal, the method in "Single-Image Fence Removal Using Deep Convolutional Neural Network" [5], demonstrates combining DCNNs with classical image processing to enhance image quality. This approach suggests possibilities for improving camera-trap images under challenging conditions. In "Deep Learning for Inexpensive Image Classification of animals on the Raspberry Pi" [6], researchers used CNNs on Raspberry Pi devices to detect animals. This study highlights the viability of adopting lightweight and cost-effective DL solutions in distant places, which is consistent with the project's deployment plan. The work in "Integrated Animal Recognition and Detection Using Deep Convolutional Neural Network" [7], addresses the issues presented by cluttered camera-trap imagery. By leveraging multilevel graph cuts and integrating machine learning techniques, this study highlights methods to enhance detection accuracy in complex environments. In "Real-time Marine Animal Image Classification by Embedded System Based on MobileNet and Transfer Learning" [8], researchers employed MobileNetV2 with transfer learning for real-time marine animal classification. This lightweight architecture's effectiveness in constrained environments is relevant to the project, particularly for edge device deployment. The paper "IoT-Based Animal Classification System Using Convolutional Neural Network" [9], investigated the use of both audio and visual data for animal classification. The findings suggest a potential to enhance the proposed system's robustness by integrating multimodal data. The work "Animal Detection from Highly Cluttered Natural Situations Using Spatiotemporal Object Region Proposals and Patch Verification" [10], addressed the problems of cluttered situations with spatiotemporal proposals and cross-frame patch verification. Combining traditional methods like HOG with Fisher Vectors and deep learning significantly improved detection in complex environments, demonstrating a hybrid approach's efficacy.

IV. METHODOLOGY

In this, we present the wildlife detection uses Deep convolutional neural networks (DCNNs) to classify and detect wild animals, enabling real-time alerts when animals are spotted. Below, we will break down the technical aspects and mathematical notations, providing a detailed explanation of how the system works.

Deep Convolution Neural Networks

Deep Convolutional Neural Networks (DCNNs) are a type of deep learning model used to identify images, find objects, and analyse videos. They are inspired by how the human visual system works, processing images by analysing the red, green, and blue components simultaneously. DCNNs use a mathematical procedure known as "convolution" to process picture data. The typical architecture of a DCNN has four layers: convolution, pooling, activation, and fully connected layers. These networks are widely used in areas such as image classification, recommendation systems, and natural language processing.

Convolution Layer

The convolution layer is the most fundamental component of a Convolutional Neural Network (CNN). This layer achieves this through a mathematical operation known as convolution. n the convolution layer, small matrices of weights, called filters or kernels, slide across the input data (like an image). These filters apply element-wise multiplications to the input data and total the results to generate an activation map, also known as a feature map. This map highlights certain features in the input data.

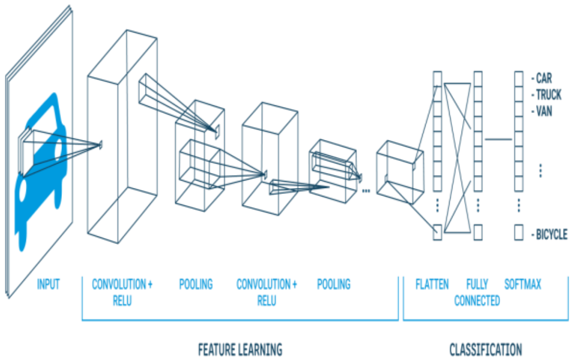

Fig. 4.1 Architecture of CNN

Mathematical Representation of Convolution

The convolution operation can be mathematically expressed as:

$$(I * K)(i,j) = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} I(i+m, j+n) \cdot K(m,n)$$

Where:

- I: input image.
- K: Kernel
- i, j: Coordinates
- Kh, Kw: Kernel's height and breadth.

This operation involves sliding the kernel over the image, performing multiplications with overlapping pixel values, and summing the results to produce the output. Use convolution n filter to the image to detect its features .

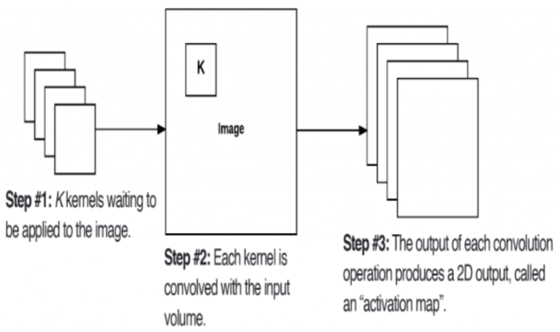

Fig. 4.2: Convolutional Filters in Feature Extraction

Convolution is the process of multiplying weights using input from a neural network. During the multiplication phase, a kernel (for 2D weight arrays) or a filter (for 3D structures) passes over an image multiple time. To cover the entire image, use the filter from right to left and top to bottom. A mathematical process used during convolution. Each filter multiplies the weights using distinct input values. The entire input is added, producing a unique value for each filter position.

ReLU Activation Layer:

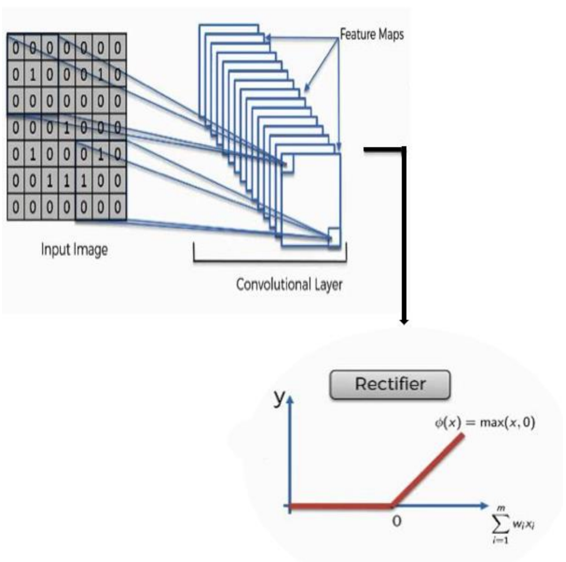

Fig. 4.3: Working of ReLU Activation Function

In a neural network, the activation function converts the node's summed weighted input into activation or output. The rectified linear activation function, or ReLU for short, is a piecewise linear function that returns the input directly if it is positive; otherwise, it returns 0. It has become the default activation function for many types of neural networks due to its ease of training and consistency in producing better outcomes.

The convolution maps are processed by a nonlinear activation layer, such as the Rectified Linear Unit (ReLu), which replaces negative integers in the filtered images with zeros.

$$f(x) = \max(0, x)$$

Where:

- f(x): Activation function
- x: Activation function accepts x as an input.

Pooling Layer:

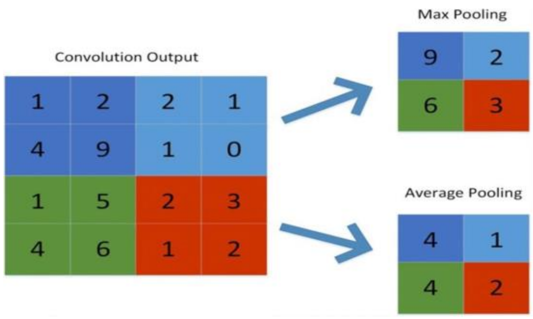

Fig. 4.4: Pooling Mechanism in CNN

The pooling layers gradually reduce the image size, leaving only the most crucial information. For example, for each group of four pixels, either the pixel with the highest value is maintained (known as max pooling) or only the average is retained. Pooling layers reduce overfitting by reducing the number of calculations and parameters in the network.

After multiple repetitions of convolution and pooling layers (in certain deep convolutional neural network topologies, this could happen hundreds of times), and the network terminates with a normal multi-layer perceptron or "fully connected" neural network.

$$f(x) = \max(x_1, x_2, x_3, x_4)$$

Where:

- x1, x2, x3, x4: Values in the 2×2 grid.

Fully Connected Layer:

Many CNN topologies have multiple fully linked layers, with activation and pooling layers in between. Fully connected layers receive an input vector containing the image's flattened pixels, which have been filtered, corrected, and reduced by convolution and pooling layers. Finally, the softmax function is used to the outputs of the fully connected layers to calculate the likelihood that the image belongs to a class, such as elephant, lion, or tiger. Following numerous layers of convolution and pooling, the image is flattened and routed via fully linked layers for final classification. In this step, the model predicts which class the input image belongs to, based on the features extracted. The softmax function converts the output into probabilities

$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Where:

- P(yi): Probability that belongs to class i.
- Zi: Raw output from the network (logits) for class i.
- The denominator sums the exponentials of all logits to normalize the output.

Bounding Box Prediction: YOLO (You Only Look Once)

Object detection relies heavily on anticipating the bounding box around an object. YOLOv5, a prominent real-time object identification model, forecasts the bounding box's centre coordinates, width, and height.

Centre Coordinates:

$$b_x = \sigma(t_x) + c_x, \quad b_y = \sigma(t_y) + c_y$$

Where:

- bx, by: predicted centre coordinates of the bounding box.
- σ(tx), σ(ty): scaled offset predictions.
- cx, cy are the grid cell coordinates.

Width and Height:

$$b_w = p_w e^{t_w}, \quad b_h = p_h e^{t_h}$$

Where:

- bw, bh: predicted width and height of the bounding box.
- pw, ph: box dimensions.
- tw, th: predicted offsets.

Loss Functions: Model Optimization

To optimize the model during training, loss functions are used to penalize incorrect predictions. These include:

CIoU Loss (Localization Loss):

$$L_{\text{CIoU}} = 1 - \text{CIoU}(b, b^*)$$

Where:

- b is the predicted bounding box.
- b∗ is the ground truth bounding box.
- CIoU measures the overlap between the predicted and ground truth boxes.

Confidence Loss:

$$L_{\text{conf}} = \text{BCE}(p_o, p_o^*)$$

Where:

- po is the predicted objectness score.
- po∗ is the ground truth objectness score.
- BCE is Binary Cross-Entropy.

Classification Loss:

$$L_{\text{cls}} = \text{BCE}(p_c, p_c^*)$$

Where:

- pc is the predicted class probability.
- pc∗ is the ground truth class label.

Intersection over Union (IoU):

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Evaluating Bounding Box Overlap. This is used during Non-Maximum Suppression (NMS) to remove redundant bounding boxes that overlap significantly.

Non-Maximum Suppression (NMS):

NMS is used to reduce redundant bounding boxes following object detection.

1. Sort boxes by confidence score.
2. Remove boxes that overlap significantly with the highest-scoring box (based on IoU threshold).
3. Repeat until no boxes remain.

Detection Metrics

Precision: Measures how many detected objects are correct.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: Determines how many genuine items were detected.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 Score: the harmonic means of precision and recall.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Mean Average Precision (mAP): Measures the average of precision at different recall levels.

System Design Architecture

This involves planning and structuring the architecture, components, and functionality of the Wild Animal Classification and Alert System. The design incorporates various technologies, including camera traps, machine learning models, and alert mechanisms, to achieve real-time wildlife monitoring and classification. The core aim of the system is to ensure scalability, reliability, and efficiency in detecting and responding to potential wildlife threats. This system automatically spots and warns people about wild animals. Cameras take pictures or videos, which are then cleaned up and prepared for analysis. A smart program then looks at the pictures, identifies any animals, and draws boxes around them. If the program thinks an animal might be a threat (based on what it is, where it is, and when it was seen), it sends alerts to people's phones or computers. All the information is saved so the system can be improved and used to study animal activity.

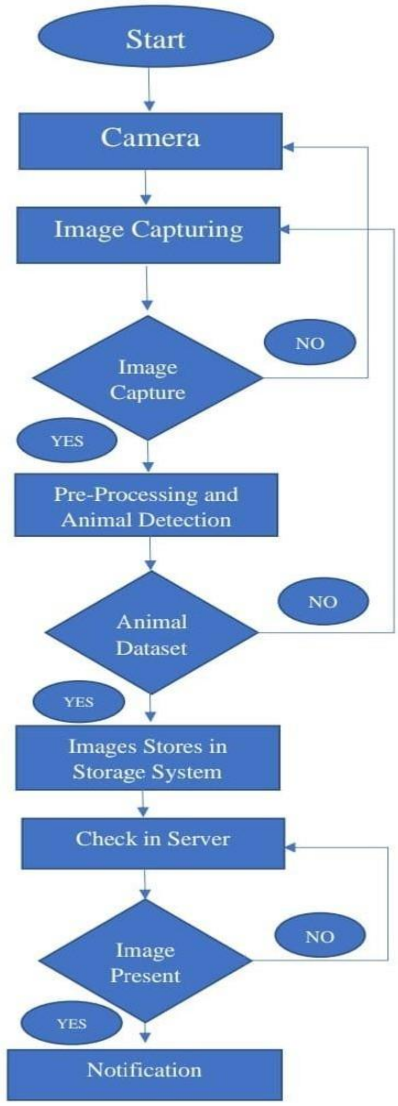

Fig. 4.5: Flowchart of the Wild Animal Classification System

In the above Flow chart, the steps involved are capturing of the image using a camera, image preprocessing and animal detection, storing image in the storage system, sending notification to the respected authorities. When a picture of a wild animal is taken, the photo is sent for pre-processing and animal recognition, whether or not it matches the information that has been collected. Animal detection is a most important part of animal- human life coexistence machine. Once the animal has been identified, the digital camera is tricked into taking particular images that must be processed in order to locate the animals based on their unique talents. The processed information must then be transferred to the server, where it is also stored. Once the animal has been identified, it is processed to generate a notification. Otherwise, it is processed again to determine whether or not to generate a notification.

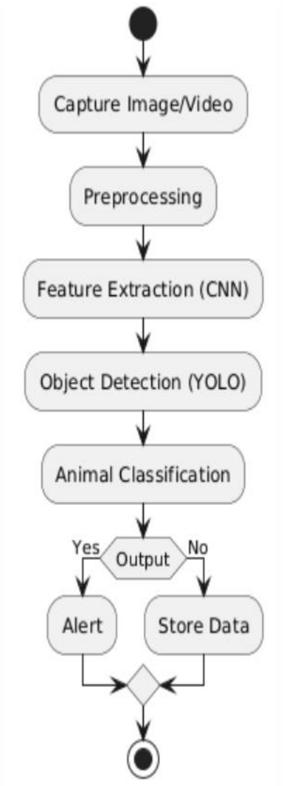

Figure 4.6: Design of the system

In above system, as shown the camera acts as the first protective sphere in which the movement of the animal is captured and then the image of the animal is captured through the media. The photo is transferred from the digital camera to the PC within the control centre, where the photo and animal class are processed.

1. Camera

The camera is set up on a post along the road side adjacent to forest area. As soon as an animal is detected with help of camera, it sends the signal to the camera, which captures the image of the specific area. The camera sends the image for processing.

2. Dataset

The dataset for the proposed project is taken from Kaggle. The images of one species of animal are stored in a folder and many such folders are used. The various ways in which an animal can be captured such as standing, sleeping, hunting, being hunted, climbing a tree, eating food, being in a group etc. are used.

3. Data Flow

Fig. 4.7: Images of elephant in the dataset

A data flow diagram depicts the flow of data within a procedure or process. It offers information about each object's inputs and outputs, as well as the way itself. DFD does not have a control flow, and no loops or judgment rules are executed. A flowchart can describe certain actions that vary according to the type of data. The DFD is a structured-analysis modelling tool. Data flow diagrams are particularly popular because they allow us to visualize the primary activities and data involved in software-systems procedures.

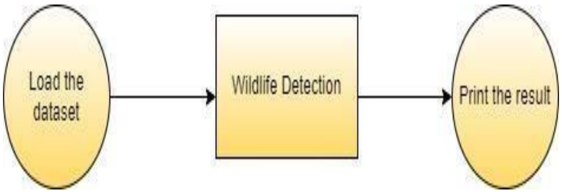

Fig. 4.8: System Dataflow for Training the Model

4. Image Preprocessing

Image pre-processing is an important stage in wild animal classification systems since it prepares the images for analysis and enhances the classification results. In a wild animal classification system, image pre-processing involves several steps as explained below. As it is known, when reading a colour image file, OpenCV reads it as BGR. The only difference between the RGB and BGR image is that, the red, green, and blue channels are read in the reverse order. Once a coloured image is read, it is converted into a grayscale image as it reduces the algorithmic complexities related to computation. In contrary to 3 channels in RGB images, grayscale images consist of only a single channel. The image is then resized as models can be easily trained on smaller sized images and aids in easier computation. Image preprocessing is an important stage in computer vision and machine learning workflows. It entails processing raw image data to increase its quality, enhance its features, and make it acceptable for downstream applications such as object detection, classification, or segmentation. It entails a sequence of processes designed to rectify, filter, normalize, or enhance images in order to increase their appropriateness for future study. These pre-processing steps include noise reduction, which removes unwanted artifacts or random variations from the image; image resizing, which adjusts the image dimensions to meet specific requirements; contrast enhancement, which adjusts the image to improve the visibility of objects or features; and image normalization, which ensures that the pixel values are scaled consistently.

5.Feature Extraction

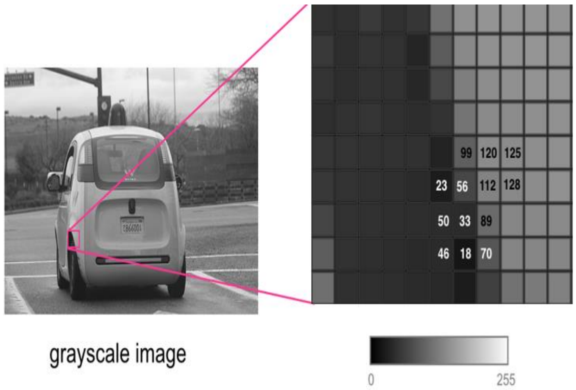

Fig. 4.9: Visual Representation of Extracted Features

Feature extraction is the process of discovering and extracting the most relevant and discriminative features from raw image data so that they can be classified and manipulated. These identify and extract different types of features from the images, such as colour, texture, and shape. The proposed form incorporates 5 convolutional layers using deep learning each went with through a maximum pooling layer. After max-pooling is carried out, the result is given as an enter for the one convolutional layer with 64 portions of length 4x4. The excess convolutional layer has 128 portions of length 3x3 went with through a totally connected layer of 512 neurons. SoftMax and RELU are used for Feature Extraction and Classification and Adam is used for Optimization.

6. Feature Matching

The most critical processes in classifying animals are feature extraction and matching. The animals are classified based on the features recognized in the collected photos. The most prevalent method for feature extraction and matching is called YOLO. It divides the image into S x S grids. If an object's canter falls within a grid cell, that grid cell is in charge of detecting it. Each grid cell predicts bounding boxes and their respective confidence scores.

7. Training a Model

A model is constructed and trained on 15 distinct types of animals. Import all of the major libraries necessary to train the model, including OS, TensorFlow, and NumPy. The YOLOv5 medium model contains 25 blocks in total (ranging from 0 to 24). Each block is a stack of various layers. The main goal is to teach the network to recognize traits that distinguish one class from others. The model was trained on 15 different kinds of wild animals. When the acquired image is analysed again, the animal will be classed according to the features that match.

8. Testing a Model

After training, the model is tested for several different images to ensure whether the animal is classified correctly based on its class. Accuracy is calculated to determine the model's efficiency. Finally, the trained network is utilized to classify animals by processing the input images.

9.Output

Once the animal is classified, notification sent to respected authorities.

V. RESULT AND ANALYSIS

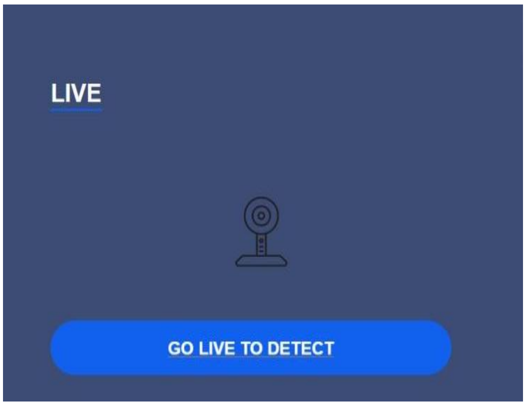

Fig 5.1: Web Interface of Proposed System

Figure 5.1 represents this demonstrates the user interface of the Wild Animal Alert System. It includes a button labelled "GO LIVE TO DETECT," which initiates the system to access the connected camera. Upon clicking this button, the camera feed starts, enabling real-time detection of animals. This interface is user-friendly and ensures the system is accessible even to individuals with minimal technical expertise.

Fig 5.2: Detection of a Deer with Bounding Box and Confidence Score

Figure5.2 represents the result of the detection and categorizing. The system has identified a deer with an accuracy score of 90%. The bounding box surrounding the animal, as well as the confidence score (0.90), indicating that the YOLO (You Only Look Once) object detection technique was applied successfully. This feature ensures accurate and efficient identification of wildlife in real-time.

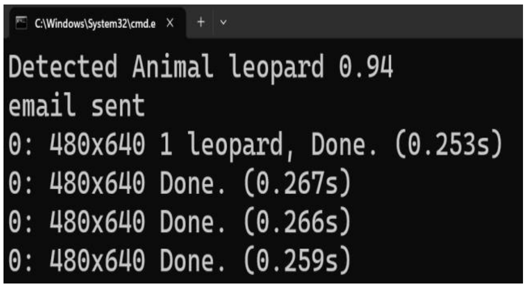

Fig. 5.3: Backend Detection Output on the Command Prompt

Figure 5.3 represents the backend output from the command prompt. It shows the detection details, such as the identified animal (leopard) with a confidence score of 94%. Additionally, the system indicates that an email notification has been successfully sent to stakeholders, informing them of the detection. The timestamps and processing times for each frame demonstrate the system's efficiency, with detection taking less than 0.3 seconds per frame.

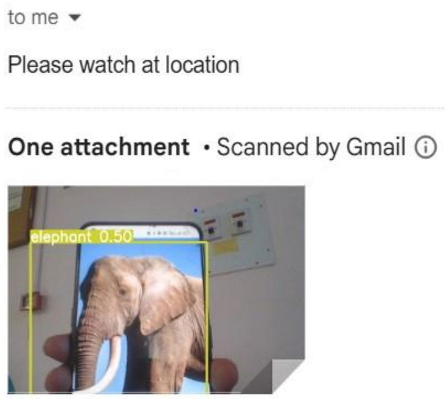

Fig.5.4: Notification Sent via Email Upon Animal Detection

Figure 5.4 represents appears to be designed to scan emails from a specific address for images. Once an image is found, the project extracts the image and attempts to identify objects within it using image recognition. If a match is found (e.g. "leopard" with a high confidence score of 0.94), the project triggers an email notification to the designated authorities. This notification likely includes details about the image and the identified object. In essence, your project automates a process of monitoring emails for images and then classifying those images using object detection. This could be useful for applications where image recognition can streamline tasks or improve efficiency.

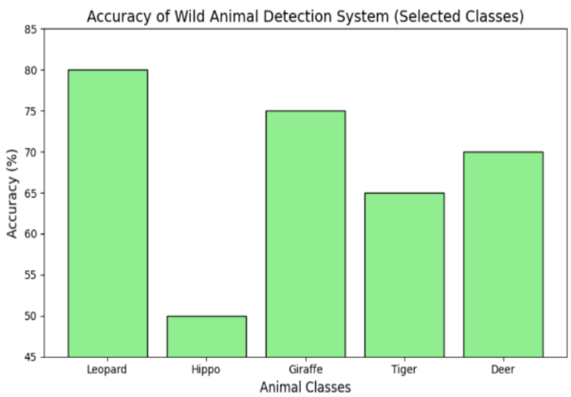

Fig. 5.5: Performance Analysis of Wild Animal Detection System

This bar graph represents the accuracy of the Wild Animal Detection System for five selected animal classes, with accuracy percentages ranging from 50% to 80%. Among these classes, the system performs best for Leopard, achieving the highest accuracy of 80%, followed closely by Giraffe at 75% and Deer at 70%. These results indicate reliable detection for these animal types. However, the system demonstrates lower performance for Tiger and Hippo, with accuracies of 65% and 50%, respectively. This shows that the algorithm may have difficulty detecting these species, maybe due to fewer distinguishing traits or low training data for these classes. Overall, the graph demonstrates the system's capabilities while also identifying areas for improvement to increase detection uniformity across all classes.

VI. CONCLUSION

The Wild Animal Alert System is a solution designed to minimize conflict between humans and wildlife using advanced technology. It employs a fast and accurate object detection system (YOLO) to identify animals in real-time using camera feeds. When an animal is detected with high confidence, the system automatically sends an email alert to relevant personnel, enabling quick responses and preventative measures. This system is designed to be user-friendly and accessible, even for non-technical users such as farmers, forest officials, and wildlife enthusiasts, making it practical for diverse field deployments. The system's real-time detection capabilities are crucial for handling dynamic and unpredictable situations, offering a proactive approach to wildlife management and safety.

This project demonstrates the power of technology in wildlife conservation and community safety. Its automated email alerts and user-friendly design make it an effective tool for avoiding potentially harmful interactions between humans and animals. The system’s scalability and adaptability mean it can be improved with new technology and changing needs. By addressing the challenges of human-wildlife interaction, the Wild Animal Alert System contributes to a more harmonious coexistence and showcases the potential of AI and machine learning for solving important environmental and social issues.

FUTURE SCOPE

In the future, we can improve by adding more advanced sensors for sensing the animals around us, such as ultrasonic sensors, drone cameras, a more powerful system, animal-specific buzzer sounds to keep the animals away from that area, a more accurate dataset for identifying a large number of animals, sending mails categorized by animal, sending alert calls along with mails via GSM, and many more. It only works with colourful images. Therefore, in future we want to work for the multi label classification.

REFERENCES

[1] Durmus Ozdemir and Musa Selman Kundurachi, “Comparison of Deep Learning Techniques for Classification of the Insects in Order Level with Mobile Software Application”, Creative Commons Attribution, ISSN: 2169-3536, Vol. 10, 2022, 35675-35684.

[2]Nidhal K. El Abbadi and Elham Mohammed Thabit A. ALSAADI, "An Automated Vertebrate Animals Classification Using Deep Convolution Neural Networks", 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Kurdistan Region - Iraq, ISBN: 978-1-7281-5249-3, April 2020, 72-77.

[3] Sonain Jamil, Fawad, Muhammad Sohail Abbas, Faisal Habib, Muhammad Umair and Muhammad Jamil Khan , “Deep Learning and Computer Vision-based a Novel Framework for Himalayan Bear, Marco Polo Sheep and Snow Leopard Detection”, 2020 International Conference on Information Science and Communication Technology, ISBN: 978-1-7281-6899-9, 2020.

[4] Tibor, Richard , Miroslav and Peter, “Animal Recognition System Based On Convolutional Neural Network” advances in electrical and electronic engineering, Vol. 15, Issue 3, 2017.

[5] T. Matsui and M. Ikehara, "Single-Image Fence Removal Using Deep Convolutional Neural Network," Vol. 8, pp. 38846-38854, 2020.

[6] Brian H. Curtin, Suzanne J. Matthews, “Deep Learning for Inexpensive Image Classification of Wildlife on the Raspberry Pi”, 0082-0087.

[7]G.Supriya,Mr.A.S.Loganathan,Mr.K.Balaji, ”Integrated Animal Recognition and Detection Using Deep Convolutional Neural Network”, International Journal of Innovative Research in Science, Engineering and Technology, vol. 8,2019, 4042-4048.

[8] Xuefeng Liu, Zhenqing Jia, Xiaoke Hou, Min Fu, Li Ma, Qiaoqiao Sun, “Real-time Marine Animal Images Classification by Embedded System Based on Mobilenet and Transfer Learning”, Supported by the Nation Natural Science Foundation of China.

[9] L. G. C. Vithakshana, W. G. D. M. Samankula, “IoT based animal classification system using convolutional neural network”, 2020 International Research Conference on Smart Computing and Systems Engineering, 2020, 90-95.

[10] Zhi Zhang, Zhihai He, Wenming Cao, “Animal Detection from Highly Cluttered Natural Scenes Using Spatiotemporal Object Region Proposals and Patch Verification”, a publication of the ieee circuits and systems society.