Received 19 June 2024, accepted 27 July 2024, date of publication 7 August 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439230

RESEARCH ARTICLE

Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction

JOHNWESILY CHAPPIDI AND DIVYA MEENA SUNDARAM
School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh 522237, India
Corresponding author: Divya Meena Sundaram (divyameena.s@vitap.ac.in)

ABSTRACT Leveraging advanced technologies, such as the cascaded YOLOv8-based approach, this research aims to detect wild animals, thereby preventing Wild animal intrusion in residential areas and sudden road crossings. A reliable wildlife animal detection system is essential for monitoring biodiversity, understanding animal behaviour, and supporting global conservation efforts. This paper uses datasets to introduce a cascaded YOLOv8-based approach for wildlife animal detection. Initially, the input dataset undergoes adaptive histogram equalisation for contrast enhancement, followed by super-pixel-based Fast Fuzzy C-Means (FCM) for segmentation. Features are then extracted using ResNet50, DarkNet19, and Local Binary Pattern, and finally, the optimal cascaded YOLOv8 detects the wildlife animals based on these features. The proposed MATLAB-based technique for detecting wildlife animals performs at its best, achieving 97% accuracy along with excellent metrics for kappa, precision, sensitivity, specificity, and F measures. This research contributes to advancing wildlife conservation efforts by providing a robust and efficient method for monitoring and preserving biodiversity. Future research endeavours may explore integrating advanced deep learning models and incorporating diverse datasets to refine further and enhance wildlife animal detection capabilities, ultimately facilitating more effective conservation strategies in natural ecosystems.

INDEX TERMS Cascaded YOLOv8, superpixels based fast fuzzy C-mean, ResNet50, DarkNet19, local binary pattern.

I. INTRODUCTION

One of the fundamental responsibilities of biologists in field research is identifying and tracking wild animals, which is essential for determining the size of the population and researching the behaviour patterns of particular organisms [1]. A crucial ecological task is to observe wild animals in their native habitats. Earth's ecosystems are undergoing rapid, unique, and significant changes due to overexploitation of natural resources brought on by the world's population increase and unrelenting pursuit of economic development [2]. Ecosystems are kept stable and healthy by protecting wildlife and their habitats. Wild animals can encourage the growth and reproduction of plants and other species since they are an integral ecosystem component.

The primary early wild animal detection method is human eye observation, which requires time and labour. People have started recognising animals by analysing visual data because of the increasing use of portable image-collecting devices.

The complicated and constantly shifting environment makes it challenging to gather data about wildlife in the wild. Modern technology has expanded options and opportunities for wildlife detection. Since infrared cameras can operate continuously for 24 hours, have little effect on various habitats or conditions, cause little animal disruption, and require very little field personnel, they are frequently utilised in wildlife surveys [3].

With the help of machine learning and a deep learning algorithm, we can add an enormous collection of animal photos to enable the computer to determine if an animal is there or not as an obstacle [4], [5], [6]. In deep learning models such as CNN and RNN, evaluation metrics and

The associate editor coordinating the review of this manuscript and approving it for publication was Claudia Raibulet.

datasets are used for training. Convolutional Neural Networks (CNNs), a subset of deep learning, have revolutionised the field of animal detection in data. These models do not require manual feature engineering because they can learn hierarchical features from raw pixel data. Because of this, the network can adjust to various animal looks. Recurrent neural networks (RNNs) and 3D convolutional neural networks (3D CNNs) also make it possible to record a temporal dependency, which improves the understanding of motion patterns over time. Given the circumstances, deep learning methods—particularly CNNs—have increased the precision and effectiveness of animal detection [7], [8], [33].

Animals typically have dense, lush backgrounds, which makes segmentation difficult due to varying lighting conditions and occlusion. The issue of animal detection is reduced by animal segmentation [9]. Automated analysis from image sequences requires the segmentation and recognition of animals based on their characteristics from the background [10]. Conventional techniques for segmenting images, such as thresholding and edge detection, are limited to identifying objects whose visual characteristics significantly contrast with their surroundings. These approaches are straightforward and only need a small amount of computation. Still, they cannot be employed for the segmentation task when faced with images of wild animals with unclear grey features, complicated backgrounds, and camouflaged environments. Nevertheless, thresholding cannot completely segment the image [11], [12].

Detection and Classification of animal species is an area that needs good techniques as it reduces the problems of wildlife road accidents leading to deaths and injuries and helps humans understand diversity better. Animal assaults frequently cause most human fatalities and injuries [13], [14], [15]. Wildlife detection and classification can help prevent animal-vehicle accidents, trace animal facilities, and prevent evading deaths, injuries, and property damage. Though attacks cannot be prevented, they can be minimised, and faster actions can be taken [16], [17].

These factors are essential for identifying wildlife animals. Thus, they employed various currently available techniques, including YOLOv7, CNN, and KNN. The existing detection method has many drawbacks due to its inability to identify tiny objects. To tackle this problem, this paper uses Cascaded YOLOv8 to detect wild animals. The significant contributions of this paper are summarised as follows:

1) This paper proposes the cascaded YOLOv8 technique to detect wildlife animals. The databases are gathered from the Kaggle Animal Images Dataset (KAD) dataset. The proposed approach has four stages: pre-processing, segmentation, feature extraction, and Animal detection.
2) Adaptive histogram equalisation contrasts the image in the preprocessing step. The preprocessing image is sent to the segmentation phase using superpixels-based fast FCM. Resnet50, Darknet29, and Local Binary Pat-tern extract features from the segmented output.
3) Finally, the optimal cascaded YOLOv8 is used to identify wildlife animals. The detection system can effectively and accurately identify animals in images using a cascaded technique.

Identified research gaps are discussed in the following. Firstly, while traditional methods such as human eye observation have long been employed, they are time-consuming and labour-intensive, highlighting the need for more efficient technological solutions. Although infrared cameras offer continuous monitoring capabilities with minimal disturbance to wildlife, challenges persist in accurately detecting animals in complex and dynamic environments. Moreover, while machine learning and deep learning algorithms, particularly Convolutional Neural Networks (CNNs), have shown promise in automating animal detection, challenges remain in segmenting animals from dense and occluded backgrounds. Current segmentation techniques, such as thresholding and edge detection, are limited in their applicability to wildlife images with varying lighting conditions and camouflage. Additionally, the importance of animal detection and classification in mitigating wildlife-related accidents underscores the need for robust detection methods that can accurately identify animals of all sizes. Despite advancements in existing techniques like YOLOv7, CNN, and KNN, there is still a gap in effectively identifying small objects, prompting the exploration of novel approaches such as the Cascaded YOLOv8 model. Addressing these gaps is crucial for advancing wildlife conservation efforts and minimising human-wildlife conflicts.

The paper is structured into several sections. It begins with a Literature Review, followed by Methods and Materials in Section III, where the proposed methodology is detailed in Section IV. Section V covers the Experimental Framework, while Section VI presents the Results and Discussion. Finally, Section VII concludes the paper, discussing the implications of the findings and outlining future research directions.

II. GUIDELINES FOR MANUSCRIPT PREPARATION

A dataset of wildlife images from the Northeast Tiger and Leopard National Park (NTLNP dataset) had been developed by Mengyu Tan et al. [18]. Additionally, they investigated the effectiveness of training models on day and night data individually versus together and assessed the recognition performance of three widely used object detection architectures. They used the following models in this experiment: FCOS under feature extractors ResNet50 and ResNet101 (anchor-free one-stage), Cascade R-CNN under feature extractor HRNet32 (anchor-based two-stage), and YOLOv5 series models (anchor-based one-stage). The experimental findings demonstrated the satisfactory performance of the day-night combined training's object detection models. Their models' average performance in animal image detection was 0.98 mAP (mean average precision), and their average performance in animal video categorisation was 88% accuracy. YOLOv5m in one stage had the highest recognition accuracy. Ecologists might save a great deal of time by using

AI technology to extract information swiftly and efficiently from enormous amounts of photos.

Two forms of segmentation were proposed by Rashid et al. [19] using the threshold and channel image separation. Thresholding in and of itself was a kind of regional segmentation. However, thresholding was insufficient to segment the image correctly; further image processing methods like the morphological process had to be used to achieve more precise segmentation. Different segmentation parameters, like the thresholding value and structuring element for the morphological process, were needed for image segmentation depending on the colour diversity of the picture's pixels. The two photos had been effectively segmented using two different techniques.

A method for the automatic detection and identification of animals using Deep CNN with genetic segmentation has been studied by Chandrakar et al. [20]. The current work demonstrates using a convolutional neural network to group input animal photo data. A comparison was made between the suggested work and common recognition techniques such as SU, DS, MDF, LEGS, DRFI, MR, and GC. There was a need for a highly accurate system for animal detection because the current approaches had higher error rates due to high false-positive & negative rate detection. The suggested work states that a 3-layer neural network was utilised for classification, and a genetic algorithm was used for segmentation. A database including 100 unique subjects with two classes and ten images per class was established to train and analyse the proposed work. The experimental findings demonstrated the segmentation utilising genetic algorithms and the originality of the suggested method in terms of f-measurement, MAE, precision, and recall. Therefore, the total results—precision (99.02%), recall (98.79%), F-Measurement (98.9%), and MAE (0.78%) improved by the proposed approach.

Using a manually annotated collection of images as training data, Rančić et al. [21] compared and presented the performance of multiple innovative network architectures and utilised the results to predict the existence of objects in the remaining dataset. They deployed three iterations of the You Only Look Once (YOLO) architecture and a Single Shot Multibox Detector (SSD) to detect deer in a densely forested area. Their effectiveness was evaluated using mean average precision (mAP), precision, recall, and F1 score. Additionally, they assessed the models' performance in real time. According to the findings, the chosen models could identify deer with a confidence score of up to 99% and a mean average precision of up to 70.45%. The fourth iteration of YOLO had the highest recall value of 75% and the highest accuracy of 86%. Although its compressed version showed four times higher real-time performance, it obtained significantly lower results, with 83% mAP In its best scenario, the best-performing models were subjected to the counting function, giving us precise deer distribution throughout all photos. YOLOv4's counting error was 8.3%, whereas YOLOv4-tiny's error was 7.1% due to miscounting 12 deer.

A methodology for detecting animals in images obtained from roadside cameras and the components of a basic animal detection system had been provided to Antônio et al. [22]. With this process, parts of the image could be identified by their features and then classified into animal and non-animal classes using Machine Learning (ML) techniques. Five methodologies were used to traverse the image's pixels to compare two machine learning algorithms. The accuracy of animal identification on highways was higher with the KNN learning model than with Random Forest.

A wild animal detection system was proposed by Verma et al. [23] to monitor wildlife and identify wild animals from extremely congested nature photos. The camera-trap network provided the data, including highly congested landscapes that made it challenging to recognise wild creatures and resulted in poor recognition and false discovery rates. They used a camera trap database, which offers candidate regions using multilayer graph cuts in the spatiotemporal area, to address the problem. The regions were used to create a validation step determining whether animals are present in a scene. The deep Convolutional Neural Network method was utilised to extract these properties from crowded photos (CNN). VGGNet and ResNet, two well-known CNN models, were implemented to develop the system on a typical camera trap database. Lastly, some of the top machine learning approaches for classification were fed the CNN characteristics. Their results showed that their suggested system outperformed other methods documented in the literature. The following Table 1 shows a Comparison with existing methods.

According to Archana et al. [24], it was proposed that wild animals that intrude into human settlements be identified. The Foreground Detector algorithm was used to identify animal movement in the input video. A Gaussian mixture model based on background removal was used to extract the animal from the foreground. Morphological filters were employed to eliminate noise from the binary image by subtracting the background. The backpropagation algorithm was used for both training and recognition. The animal was identified if the test image and the taught images matched. The project benefited the forest department and strived to protect wildlife and other animals.

Based on the literature mentioned above, we have discovered the following problems with existing systems:

1) Some wild animals, such as rodents and small birds, might be difficult to spot because of their small size, especially from a distance or in crowded areas.
2) Unfavorable weather conditions, including rain, fog, snow, or extremely high or low temperatures, could reduce the performance of detection systems.
3) Certain species, such as flying birds or fast-moving mammals like cheetahs, have a high movement rate, making it difficult for detecting systems to identify them precisely.
4) The existing method can reduce the model's ability to learn the variability of the background, which leads

TABLE 1. Comparison with existing methods.

Author	Method	Merits	Limitations
Tan et al. [18]	Object Detection with FCOS, Cascade R-CNN, YOLOv5	Combined training of day and night data improves object detection performance. YOLOv5m achieved the highest recognition accuracy.	Correctly identifying rare or similar-looking species can be difficult. Misclassifications may occur due to visual similarities.
Rashid et al. [19]	Segmentation using thresholding and channel image separation	Effective segmentation through a combination of thresholding and morphological processing. Different segmentation parameters were needed based on color diversity.	Thresholding alone isn't enough to properly segment the image. Additional image processing techniques, like the morphological process, are necessary to achieve more precise segmentation.
Chandrakar et al. [20]	Deep CNN with genetic segmentation	Utilised CNN for animal photo data grouping. Achieved improved segmentation and classification accuracy using genetic algorithms.	higher error rates due to high false-positive and false-negative rate detection in existing methodologies
Rancic et al. [21]	YOLO and SSD for deer detection	Successfully detected deer with high confidence and mean average precision. YOLOv4 had the highest recall, while YOLOv4-tiny showed better real-time performance.	In dense environments, objects may overlap in images, making detection and accurate counting difficult.
Antônio et al. [22]	Machine Learning for animal identification	Applied ML techniques for identifying animals in roadside camera images. KNN outperformed Random Forest in accuracy.	The ability of the model to generalise to new, unseen data can be a challenge, especially if the training data is not sufficiently diverse.
Verma et al. [23]	Wildlife detection with CNN models	Proposed a system using CNN models (VGGNet, ResNet) for wildlife detection in congested photos. Outperformed other documented methods.	The cluttered images from camera-trap networks make it difficult to detect wild animals, leading to low detection rates and high false discovery rates.
H. et al. [24]	Foreground Detector and Backpropagation for wildlife identification	Used Foreground Detector algorithm for animal movement identification. Achieved wildlife identification through background removal and morphological filters.	The techniques might not work consistently in different environments with varying lighting, outdoor environments, and noise.

the model to misidentify everything that looks even remotely like an animal [25].

These are the main challenges which motivate us to do this research on wildlife animal detection.

III. METHODS AND MATERIALS

Animal identification and recognition in their natural environments, utilising a variety of technological instruments and techniques, is known as wildlife animal detection. Wildlife animal detection aims to use various technologies to track and monitor animal populations in their natural environments.

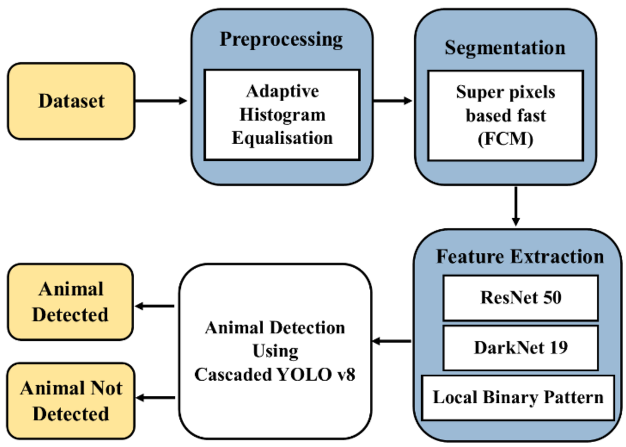

FIGURE 1. Block diagram of the proposed method.

The complicated issue is accurately identifying and tracking wildlife animals in their natural habitat. To overcome those problems, optimal feature-based two-cascaded YOLOv8 algorithms are proposed here. The suggested method consists of four steps: 1) preprocessing, 2) segmentation, 3) feature extraction, and 4) animal detection. In the preprocessing step, Adaptive histogram equalisation is used to im-prove the input image's contrast. The following method is segmentation. The preprocessed output image is segmented using Super pixels-based Fast FCM (Fuzzy C-Means). After that, the feature is extracted from the segmented output using Res-net50, Darknet29, and Local Binary Pattern.

Lastly, cascaded YOLOv8 is proposed as a method to detect wildlife animals.

A. PRE-PROCESSING

Pre-processing is crucial in wildlife animal detection to ensure the data is appropriately formatted, cleaned, and enhanced for practical analysis. This pre-processing technique uses Adaptive Histogram Equalization to improve the contrast in images.

1) ADAPTIVE HISTOGRAM EQUALIZATION

A digital image processing method called adaptive histogram equalisation improves image contrast. By enhancing the contrast locally, the adaptive technique deviates from standard histogram equalisation. It separates the image into discrete blocks and then calculates the equalisation of the histogram for each block.

The above-described approach is used to pre-process the input dataset. The segmentation method then receives the preprocessing output.

B. SEGMENTATION

Animals are separated from their background in images or videos by a segmentation method to detect wildlife animals. This segmentation method uses Super pixels-based Fast Fuzzy C-Means (FCM) to segment the preprocessed output images.

1) SUPERPIXELS-BASED FAST FUZZY C-MEANS (FCM)

A pixel can be fixed in one or more clusters using the data clustering technique known as FCM. Similarly, a cluster comprises some degree of each data point and is typically identified by a membership degree. Moreover, for producing a superpixel-based FCM method, a finite collection of super-pixels is partitioned into several fuzzy cluster collections c. In consideration of some given criteria. Further, the objective function of a superpixel-based FCM method obtained by dividing a superpixel dataset $f_{j=1}^{n}$ into several clusters c, Expressed as follows:

$$J(u, v) = \sum\nolimits_{i=1}^{c} \sum\nolimits_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} \tag{1}$$

Here, the cluster centroids are denoted as $v = [\![(\{v\}]\!]_1, v_2, \ldots, v_c\}$ and also between i^{th} centroids and the j^{th} superpixel; the Euclidean distance computed is indicated as d_{ij}; The membership among 0 and 1 is denoted as u_{ij}. The cluster fuzziness level is determined with $m\epsilon\,[1, \infty]$. However, the goal function is repeatedly optimised using a fuzzy partition of the well-known super-pixel sample [28].

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}}, v_i = \frac{\sum_{j=1}^{n} u_{ij}^{n} f_j}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{2}$$

In particular, most noise-free images had good segmentation using the conventional FCM method's functions. Nevertheless, noise-corrupted photos will produce non-robust results, making it difficult to segment these images. Fundamentally, the problems above are caused mainly by non-robust Euclidean distance and disregard for spatially relevant information in the image. To overcome these problems, we thus suggest robust distance measures. On the other hand, this distance measure is indicated by improving the conventional FCM by incorporating the neighbouring and similar super-pixel into its objective function. The output that is produced is then fed into the following process.

C. FEATURE EXTRACTION

Feature extraction is a crucial task in wildlife animal detection. Three models extract the features from the segmentation output images: ResNet50, DarkNet19, and Local Binary Pattern (LBP). Res-Net50 excels at learning hierarchical features from images; DarkNet19 is efficient for real-time object detection tasks; and LBP effectively captures local texture patterns in images with low computational cost. The following provides a thorough explanation of each feature extraction technique.

1) RESNET50

ResNet50 is the name for residual networks that have fifty layers. ResNet50 has more capacity for identity mapping. The ResNet50 model, which has fifty layers for feature extraction, resolves the overfitting issue in the training data. Convolution-al neural networks like ResNet50 have been extensively utilised for various computer vision applications, such as feature extraction, object identification, and image classification.

A fixed-dimension input image is used to start the procedure. This image is usually square and has pixels as its dimensions for ResNet-50. Res-Net-50 comprises multiple convolutional layers layered on top of one another. These layers perform feature extraction by identifying patterns and features at various levels of abstraction. The deeper levels capture more intricate details. The usage of residual blocks is one of ResNet's primary innovations. For the network to learn residual functions, skip connections, also known as shortcuts, are present in every residual block. This makes it possible to train very deep networks by assisting in the fight against the vanishing gradient issue during training. Pooling layers, such as max pooling, are periodically employed to minimise the feature maps' spatial dimensions, lowering computing complexity and preventing overfitting. The convolutional layers extract high-level features mapped to the required output classes or features by fully connected layers at the end of the network. The output is obtained directly from the convolutional layers when employing ResNet for feature extraction, although these fully linked layers are frequently eliminated [29]. Extracted features are received from the convolutional layers' output. It is possible to detect animals using these features. Figure 2 displays the ResNet50 network structure.

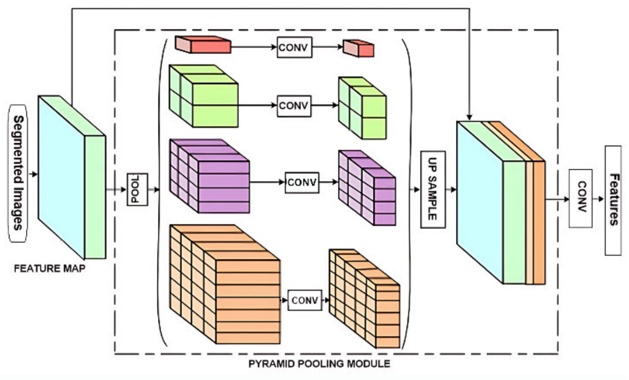

FIGURE 2. ResNet50 network structure.

2) DARKNET19

One popular deep-learning model for object detection is called DarkNet19. A convolutional neural network (CNN) with 19 layers brings up Dark-Net-19.

DarkNet19's initial input is a fixed-size input image. Convolutional layers are applied to the input image. Each convolutional layer adds filters to the input image to detect features like edges, textures, and patterns. A non-linear activation function is applied element-by-element after each convolutional layer to add non-linearity to the network. Pooling layers, typically max pooling, minimise the spatial dimensions of the feature maps while preserving the most crucial data. This facilitates the achievement of translation invariance and lowers computing complexity. The feature maps become flat into a one-dimensional vector near the conclusion of the network. After that, one or more fully connected layers are routed through these flattened features. To create a High-level representation of the input image, these layers first conduct a weighted sum of the input characteristics and then an activation function. The network's last layer generates the output predictions. It could entail estimating bounding boxes, animal classes, and confidence scores in the case of animal detection. Figure 3 depicts the network structure of DarkNet19.

DarkNet19's initial input is a fixed-size input image. Convolutional layers are applied to the input image. Each convolutional layer adds filters to the input image to detect features like edges, textures, and patterns. A non-linear activation function is applied element-by-element after each convolutional layer to add non-linearity to the network. Pooling layers, typically max pooling, minimise the spatial dimensions of the feature maps while preserving the most crucial data. This facilitates the achievement of translation invariance and lowers computing complexity. The feature maps become flat into a one-dimensional vector near the conclusion of the network. After that, one or more fully connected layers are routed through these flattened features. To create a High-level representation of the input image, these layers first conduct a weighted sum of the input characteristics and then an activation function. The network's last layer generates the output predictions. It could entail estimating bounding boxes, animal classes, and confidence scores in the case of animal detection. Figure 3 depicts the network structure of DarkNet19.

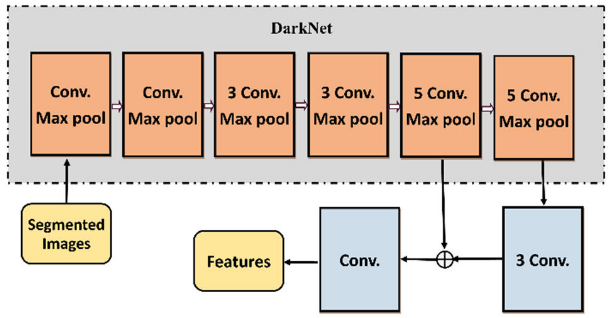

FIGURE 3. DarkNet19 network structure.

3) LOCAL BINARY PATTERN (LBP)

A texture descriptor called Local Binary Pattern (LBP) is used in computer vision and image processing to extract features from images. It works exceptionally well for applications like object detection, face recognition, and texture categorisation. LBP shows the magnitude relationship between the center pixel and surrounding pixels in a micro pattern. The LBP value is assigned to one or zero depending on whether the next pixel is higher or equal. The descriptor uses a binary integer (binary pattern) to represent the outcome over the neighbourhood. The following is the mathematical expression of LBP for a pixel,

$$t = g\left(x_i\right) - g(x) \tag{3}$$

$$LBP\left(x\right) = \sum_{i=0}^{p} s\left(t\right) 2^i$$

$$\text{Where, } s\left(t\right) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{4}$$

where, x is the location of the centre pixel, x_i is the location of the i^{th} neighbouring pixel and $g(\cdot)$ is the pixel intensity value [28].

The features are extracted from the segmentation output based on the above technique. The identification of the animal detection is then done using the resulting features.

IV. PROPOSED METHODOLOGY

A. ANIMAL DETECTION SYSTEM

Animal detection involves finding wildlife using the Cascaded YOLOv8 method. This method effectively accomplishes animal identification and localisation in photos or video streams. Cascaded YOLOv8 is an enhanced version of the YOLO (You Only Look Once) object detection model, specifically YOLOv8, which aims to improve the accuracy and robustness of the original model. The following section provides a thorough discussion of the optimal cascaded YOLOv8.

1) OPTIMAL CASCADED YOLOV8 METHOD

The proposed algorithm uses a state-of-the-art object detection algorithm called YOLOv8 to detect wildlife animals. In contrast to earlier, highly effective models in the YOLO series (such as YOLOv5 and YOLOv7), YOLOv8 is a sophisticated and state-of-the-art model that provides faster and more accurate detection. YOLOv8, which builds on the innovations of earlier YOLO versions, adds additional features and optimisations that make it the best option for various object identification tasks across various applications. The deep convolutional neural network (CNN) architecture of YOLOv8 is built on that of its predecessors.

YOLOv8 comprises three primary parts: the backbone, neck, and head.

1) A new backbone design is known as CSPNet that outperforms earlier backbones in accuracy and efficiency.
2) A new neck design known as FPN + PAN that more effectively aggregates features from various backbone levels.
3) A new head architecture known as PANet that is more resilient to scale changes and occlusion. The overall YOLOv8 network architecture is visually depicted in Figure 4.

2) OPTIMAL CASCADED YOLOV8 METHOD

CSPDarknet53 serves as the basis for the construction of YOLOv8 and comes in five different sizes: nano, small, medium, giant, and extra-large. It has fifty-three convolutional layers. Five down-sampled iterations of the input features yield five distinct scale features, designated P1 – P5. Figure 4 depicts the backbone network's structure. The C2f module replaces the original backbone networks Cross Stage Partial (CSP) module. To improve the information flow of the feature extraction network while still being lightweight, the C2f module uses a gradient shunt connection. The backbone

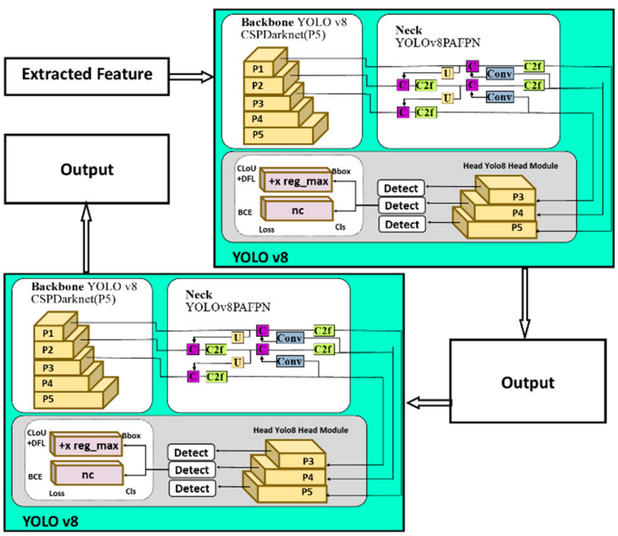

FIGURE 4. Shows the cascaded YOLOv8 network structure.

network pools the input feature maps to a fixed-size map for adaptive size output using the spatial pyramid pooling fast (SPPF) module.

3) NECK

Figure 4 illustrates the PAN-FPN (Path Aggregation Network–Feature Pyramid Network) architecture used to create YOLOv8. With channel fusion via up-sampling, this feature pyramid network incorporates three down-sampled inputs. In contrast to the neck structure of the YOLOv5 and YOLOv7 models, the YOLOv8 model preserves original performance while attaining a lightweight design by eliminating the convolution operation in the PAN structure following up-sampling. In the end, three branches get the output, fed toward the decoupling head.

4) HEAD

As illustrated in Figure 4, YOLOv8's detecting part has a decoupled head structure. For predicted bounding box regression and object classification, the decoupled head structure employs two distinct branches, each using a different loss function. Bi-nary cross-entropy loss, or BCE Loss, is applied to the classification task. Distribution focal loss (DFL) and CIoU, this detection structure is used for the projected box bounding regression challenges. It can accelerate model convergence and increase identification precision [31]. DFL uses cross-entropy to model the target detection frame's position as a global distribution. This optimisation strategy increases the likelihood that the position is near the label. As a result, the network can quickly focus on the target position, as indicated by Equation (5).

$$DFL\,(s_i, s_{i+1}) = -\left((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_{i+1})\right) \tag{5}$$

whereas DIOU calculates the Euclidean distance between the centers of the two detection frames, *CIoU* quantifies the difference between the actual and predicted frames. As seen in Equations (6) to (8), *CIoU* expands on *DIoU* by considering the aspect ratio of the detection frames. α represents the weight function ν used to measure the similarity of the aspect ratio; *IoU* is the intersection ratio between the actual frame and the predicted frame; ρ is the Euclidean distance between the centers of the predicted frame and the actual frame; a and a^{gt} represent the centres of the predicted and real frames; c is the diagonal distance of the smallest enclosing region that can contain predicted and real frames.

$$CIOU = \frac{\rho^2\left(a, a^{gt}\right)}{b^2}\alpha\nu \tag{6}$$

$$\nu = \frac{4}{\pi^2}\left((arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})\right)^2 \tag{7}$$

$$\alpha = \frac{\nu}{(1 - IOU) + \nu} \tag{8}$$

5) CASCADE IMPLEMENTATION

Cascading involves using one model's output to refine another's predictions. In the context of YOLO (You Only Look Once) models like YOLOv8, cascading typically means using multiple stages of animal detection to improve accuracy and robustness. Unlike the traditional single-stage YOLOv8, which directly predicts bounding boxes and class probabilities in one pass, cascaded YOLOv8 breaks down the detection process into several iterative stages, allowing for progressive refinement of predictions. A technique for merging each stage's outputs to obtain the final detections must be designed.

The Cascaded YOLOv8 model starts by coarsely detecting the animals in the image using a YOLOv8 model. This stage aims to quickly identify regions of interest (ROIs) where animals might be present. These ROIs are then subjected to a more refined detection process in the second stage, utilising another YOLOv8 model. This subsequent stage operates with either a smaller input size or higher resolution, allowing for finer localisation and classification of animals within the identified ROIs. Following the second detection stage, post-processing steps such as non-maximum suppression (NMS) help refine the final bounding boxes by eliminating redundant detections. Ensemble methods can combine outputs from both stages, improving detection accuracy. More modifications to training data and techniques could be necessary to train the cascaded model effectively.

Using a cascaded approach, the detection system can efficiently identify animals in images while maintaining high accuracy. The primary advantage of cascaded YOLOv8 lies in its improved accuracy and robustness. The use of cascaded techniques in YOLOv8 offers significant advantages for animal detection. By breaking down the detection process into multiple stages, each refining the results of the previous one, cascaded approaches improve accuracy by reducing false positives and false negatives and enhance performance in complex or high-precision applications by refining animal predictions through multiple stages. They also reduce the computational load by filtering out unlikely candidates early, leading to faster processing times without compromising detection quality. This is especially beneficial in complex environments with cluttered backgrounds, where cascaded approaches can progressively focus on relevant regions, distinguishing animals from background noise. This modularity makes it easier to update the model for specific detection tasks or to incorporate new data, enhancing its ability to detect animals in diverse and dynamic environments.

Wildlife detection is accomplished using the Optimal Cascaded YOLOv8 method. The next part explains the experimental investigation for the suggested method [30].

The model training parameters for the Missouri Camera Traps dataset are as follows: The learning rate is set to 0.001 for GPU and 0.0005 for CPU, with 50 epochs for GPU and 30 for CPU. The batch size is 32 for GPU and 16 for CPU, using the *ReLU* activation function. The architecture consists of 4 convolutional blocks and 4 deconvolutional blocks, with L2 regularisation applied. Training involves 10,000 iterations for GPU and 7,000 for CPU. The genetic algorithm's population size is 50 for GPU and 30 for CPU.

V. EXPERIMENTAL FRAMEWORK

This section discusses the dataset description with sample images, performance metrics, system setup, and implementation details.

A. DATASET DESCRIPTION

The experiments that are conducted make use of a KAD dataset that consists of images that have been annotated for animal detection. The annotations were done using the YOLO COCO model, which is well-known for its ability to detect animals in various environments. The Missouri Camera Traps dataset has around 25,000 camera trap images having 20 species, including prominent labels. All sequences have congested scenes with different spatial resolutions, marking the same recognition of species and challenging circumstances. The Wildlife Image and Localization Dataset (WILD) comprises 5,784 photos and 12,007 labelled annotations covering 28 species.

TABLE 2. Comparison and datasets description.

Description	Missouri Camera Traps	WILD	KAD
No. of Images	55,000	5,784	28,000
No. of Classes	20	28	10
Image Resolution	vary from 1920 × 1080 to 2048 × 1536	Varied	256×256
Data Format	Annotations in COCO Camera traps .Json format and whitespace-delimited text format	Bounding box in Pascal VOC format	COCO format for compatibility with YOLO models
Total size	10GB	1.4 GB	3.4 GB

This dataset provides valuable insights into real-world detection scenarios for wildlife-related activities. The table.3 discusses more about these datasets, fig.5. shows a Camera-trap image sample, fig.6. shows sample images from the KAD dataset. The dataset description and comparison are mentioned in Table 2.

B. PERFORMANCE METRICS

The performance of our proposed model was assessed with a few usual metrics like Accuracy, F-measure, Kappa, Precision, Sensitivity and Specificity. Accuracy is the proportion to which the model is correct or perfect. F-measure evaluates the performance of a classification model, particularly in binary classification tasks. Cohen's Kappa, often called Kappa, is a statistical measure that assesses inter-rater agreement for categorical data, accounting for agreement occurring by chance. Precision is the model's Positive Predictive Value (PPV), i.e., being accurate. Sensitivity is the ability of the model

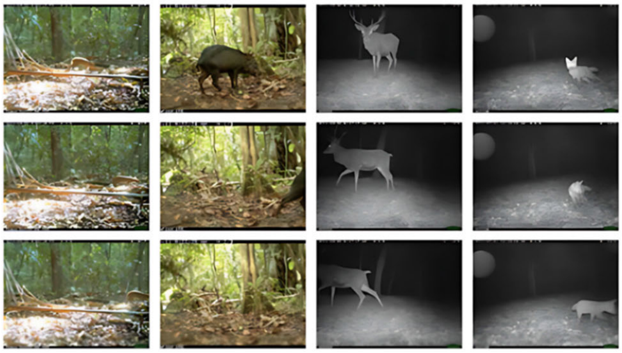

FIGURE 5. Camera-trap image samples, each column representing a sequence triggered by animal motions.

FIGURE 6. Sample image from the KAD dataset.

to recall. Specificity is the percentage to which the model is exact. Equations from (9) to (14) determine the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values to calculate these six components. Use these values to estimate the model's accuracy, F-measure, Kappa, precision, specificity, and sensitivity. A list of each is provided below.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{Fmeasure} = \frac{2 \times \text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}} \tag{10}$$

$$kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP+FP) \times (FP+TN) + (TP+FN) \times (FN+TN)} \tag{11}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{14}$$

C. SYSTEM SETUP

MATLAB 2020b software, deep learning and parallel computing toolboxes were used to conduct our experiment, which was installed on a laptop with a Core i7-12750H processor, NVIDIA GeForce RTX 2070 graphics accelerator, 16 GB of RAM, and running a Windows 11 Professional x64 operating system.

VI. RESULT AND DISCUSSION

This part evaluates wildlife animal detection using the optimal cascaded YOLOv8 technique. Here, the suggested model is implemented using the MATLAB platform. Numerous performance metrics are computed to assess the proposed animal detection, such as accuracy, F score, kappa, precision, sensitivity, and specificity. The research compares the training percentage and values of the proposed model to those of several other algorithms that are presently in use, including You Only Look Once v7 (YOLOv7), Convolutional Neural Network (CNN), Deep Neural Networks (DNN), Deep Belief Networks (DBN) and Recurrent Neural Network (RNN). In fig.7. suggested method's overall output is displayed below.

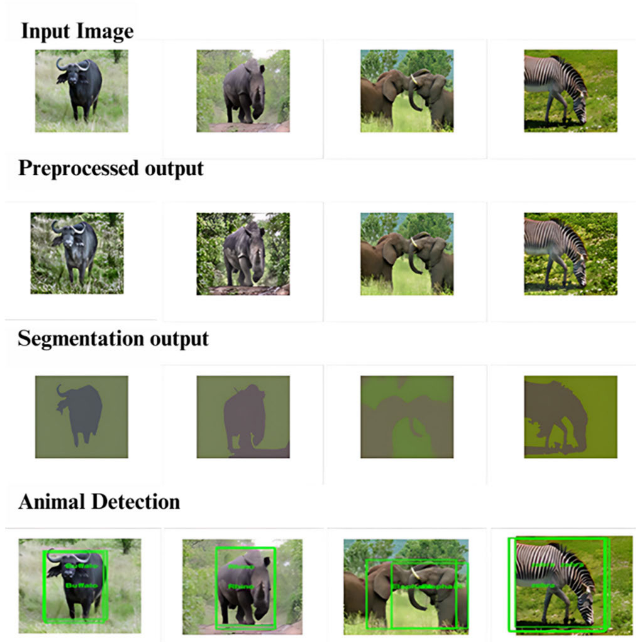

FIGURE 7. The output of the proposed method.

Table 3 shows that a KAD dataset achieves an Accuracy of 97%, a Missouri Camera Traps dataset achieves 98%, and a WILD dataset achieves 96.6%. The system is robust, stable, and suitable for dealing with images captured from the wild. However, in the KAD dataset, using the Cascaded YOLOv8 enhances its ability to detect animals in diverse and dynamic environments.

TABLE 3. Performance evaluation.

Dataset	Name	Accuracy
Dataset 1	KAD	97%
Dataset 2	Missouri Camera Traps	98%
Dataset 3	WILD	96.6%

Graphs showing accuracy and loss versus epoch provide an efficient way of illustrating this research during training and testing. The results are shown in the graph below.

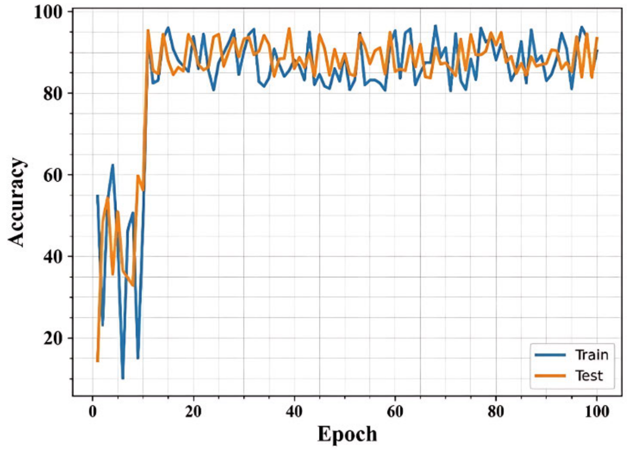

FIGURE 8. Graph depicting the training and testing accuracy with epoch.

Fig. 8. displays a graphical depiction of the accuracy at the epoch. The plot indicates that accuracy values rise across epochs. It shows that the accuracy of both the training and testing sets increases as the number of epochs increases.

Fig. 9. displays a graphical depiction of the loss at epoch. The graphic indicates that the loss values drop off throughout the course of epochs. It shows that the loss of both the training and testing sets decreases as the number of epochs decreases.

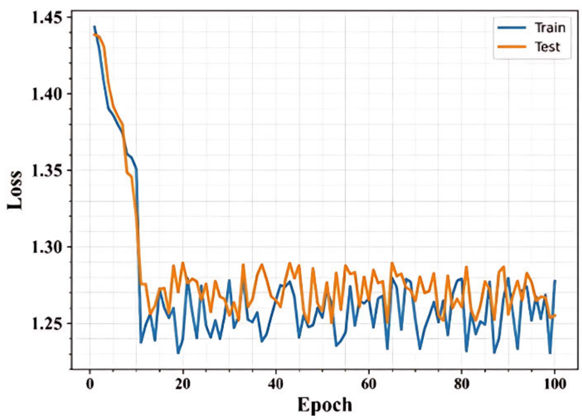

FIGURE 9. Graph depicting the training and testing accuracy with epoch.

Several techniques are analysed, and the training percentage and values of You Only Look Once v7 (YOLOv7), Convolutional Neural Network (CNN), Deep Neural Network (DNN), Deep Belief Networks (DBN) and Recurrent Neural Network (RNN) are compared with the proposed model. The comparison results of the proposed model are plotted in the graphs below.

A. COMPARATIVE ANALYSIS

The proposed method's training percentage and values are compared with the algorithms explained in this section. The results are shown in the graph below. The performances of the several quality metrics are compared with the suggested technique in more detail below.

Fig. 10. displays a graphic representation of the projected approach's accuracy measure. It is re-viewed with different training percentages; during 50% of the training percentage, the proposed technique achieves an accuracy of 0.95, whereas the present method achieves 0.89 for YOLOv7, 0.86 for CNN, 0.8 for DNN, 0.81 for DBN and 0.79 for RNN. During 80% of the training percentage, the proposed technique achieves an accuracy of 0.96, whereas the present method achieves 0.93 for YOLOv7, 0.9 for CNN, 0.85 for DNN, 0.84 for DBN and 0.82 for RNN. This evaluation shows that the proposed method obtains better accuracy than the current methodologies.

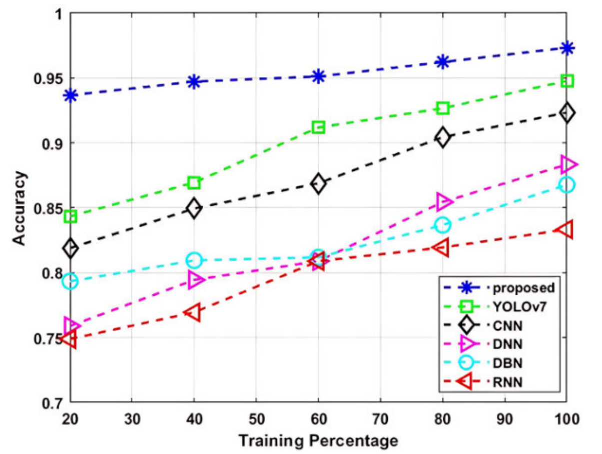

FIGURE 10. Graph depicting accuracy measure.

A graphic illustration of the suggested approach's F-measure is shown in Figure 11. It is examined with different training percentages; during 50% of the training percentage, the proposed technique achieves an F-measure of 0.91, whereas the present method achieves 0.89 for YOLOv7, 0.88 for CNN, 0.84 for DNN, 0.78 for DBN and 0.75 for RNN. During 80% of the training percentage, the proposed technique obtains an F-measure of 0.94, whereas the existing method achieves 0.92 for YOLOv7, 0.9 for CNN, 0.85 for DNN, 0.81 for DBN and 0.79 for RNN. Based on this evaluation, the proposed approach achieves the highest F-measure level compared to the existing methods.

The kappa of the suggested approach is shown in Fig. 12. It is examined with different training percentages; during

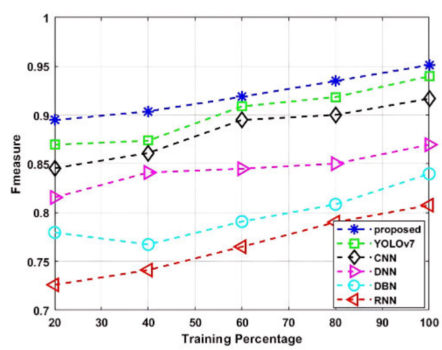

FIGURE 11. Graph depicting the F-measure measure.

50% of the training, the proposed technique achieves a kappa of 0.88, whereas the present method achieves 0.86 for YOLOv7, 0.83 for CNN, 0.81 for DNN, 0.78 for DBN and 0.75 for RNN. During 80% of the training, the proposed method achieved a kappa of 0.92, whereas the present method achieves 0.9 for YOLOv7, 0.89 for CNN, 0.85 for DNN, 0.84 for DBN and 0.8 for RNN. According to this evaluation, the suggested approach achieves the highest kappa level compared to the current methodologies.

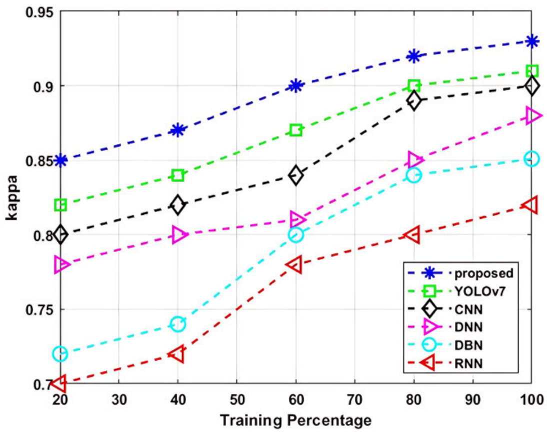

FIGURE 12. Graph depicting the Kappa measure.

The projected approach's precision measure is examined and shown in Fig. 13. It is reviewed with different training percentages; during 50% of the training percentage, the projected technique attains a precision of 0.86, whereas the present method achieves 0.83 for YOLOv7, 0.8 for CNN, 0.75 for DNN, 0.65 for DBN and 0.64 for RNN. During 80% of the training percentage, the projected technique obtained a precision of 0.92, whereas the present method achieves 0.87 for YOLOv7, 0.86 for CNN, 0.84 for DNN, 0.75 for DBN and 0.75 for RNN. Based on this evaluation, the proposed approach achieves the highest precision level compared to the traditional methods.

Fig. 14. provides a graphical representation of the projected approach's Sensitivity. It is examined with different training percentages; during 50% of the training, the proposed technique achieves a Sensitivity of 0.88, whereas the present method achieves 0.87 for YOLOv7, 0.84 for CNN, 0.79 for DNN, 0.73 for DBN and 0.56 for RNN. During 80% of the, the proposed technique obtains a Sensitivity of 0.93, whereas the present method achieves 0.9 for YOLOv7, 0.87 for CNN, 0.86 for DNN, 0.8 for DBN and 0.76 for RNN. Based on this evaluation, the proposed approach achieves the highest Sensitivity level compared to the existing methods.

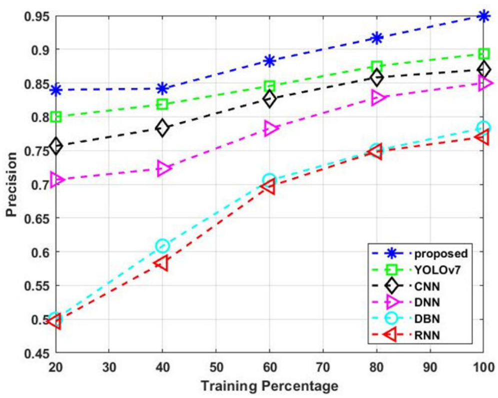

FIGURE 13. Graph depicting precision measure.

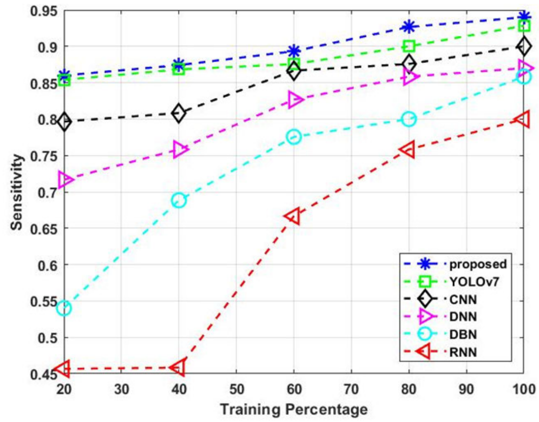

FIGURE 14. Graph depicting sensitivity measure.

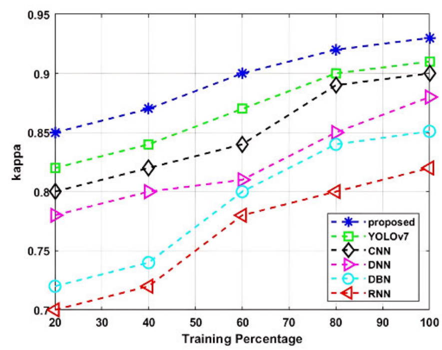

FIGURE 15. Graph depicting specificity measure.

Fig. 15. analyses and displays a graphic representation of the predicted approach's Specificity measure. It is examined with different training; during 50% of the training, the projected technique attains a specificity of 0.94, whereas the present method achieves 0.91 for YOLOv7, 0.89 for CNN, 0.85 for DNN, 0.88 for DBN and 0.8 for RNN. During 80% of the training, the projected technique obtained a specificity of 0.95, whereas the present method achieves 0.93 for YOLOv7, 0.89 for CNN, 0.87 for DNN, 0.86 for DBN and 0.82 for RNN. This investigation shows that, compared to the current methodologies, the suggested method delivers the best level of specificity.

Table 4 provides a comparison analysis of the method's performance metrics. When we compare the proposed method with the existing method across all performance indicators, the results clearly show that the proposed performance metrics have the highest values. The table values where *A* is accuracy, is F-measure, k is kappa, P is Precision, SE is Sensitivity, and Sp is Specificity values.

TABLE 4. Comparison analysis.

Method	A	F	K	P	SE	SP
Proposed	0.97	0.95	0.93	0.95	0.94	0.96
YOLOv7	0.95	0.94	0.91	0.89	0.93	0.95
CNN	0.92	0.92	0.90	0.87	0.9	0.93
DNN	0.88	0.87	0.88	0.85	0.87	0.88
DBN	0.87	0.84	0.85	0.78	0.86	0.92
RNN	0.83	0.81	0.82	0.77	0.8	0.86

The experiment's results show that the suggested strategy detects animals more accurately than the existing methods. The following is a list of the results of comparing the proposed method with several research articles.

Table 5 compares the accuracy of the proposed Cascaded YOLOv8 model with earlier attempts. The results show that, compared to the other different research methodologies, the proposed method achieves 97% accuracy. The recommended methodology yielded superior results since it used the Cascaded YOLOv8.

TABLE 5. Proposed performance comparison with state-of-art techniques.

Source	Object Detection Network	Accuracy
Mengyu Tan et al.,[16]	YOLOv5, FCOS and Cascade R-CNN	88%
M N Rithvik et al.,[14]	DNN	93%
Sanjay S et al.,[4]	CNN	91%
R. Kavitha et al.,[2]	CNN	96.6%
Aibin Abraham et al.,[23]	YOLO	94%
Thirupathi Battu and D. Sreenivasa Reddy Lakshmi [24]	DNN	90%
Lukasz Popek et al.,[25]	RCNN-YOLO	94%
Proposed	Cascaded YOLOv8	97%

VII. CONCLUSION

This paper proposes an optimal cascaded YOLOv8 method to identify wildlife animal detection. The input dataset has been gathered from the KAD dataset. The proposed method is scheduled in four stages: pre-processing, segmentation, feature extraction, and animal detection. The input image is initially given to the preprocessing stage, where adaptive histogram equalisation contrasts the image. Next, superpixel-based Fast Fuzzy C-Means (FCM) segment the pre-processing output. After that, ResNet50, DarkNet19, and Local Binary Pattern extract features from the segmented output. Finally, based on the features, the optimal cascaded YOLOv8 is used to identify wildlife animals. The MATLAB programming language is employed here to carry out the recommended model. The proposed approach's effectiveness is assessed through several performance metrics, including accuracy, precision, sensitivity, specificity, kappa, and F measures. The accuracy of the suggested model for detecting wildlife animals is 97%. Based on this evaluation, the proposed approach has achieved optimal results. The experiment outcomes demonstrate that the suggested approach detects wildlife animals more accurately than the current techniques. Cascaded YOLOv8 achieves high accuracy in wildlife animal detection but faces challenges like environmental variability, camouflage, and occlusion. So, we intend to enhance the research for upcoming projects using more advanced deep learning models, such as various YOLO iterations and datasets for detecting wildlife animals.

A. FUTURE SCOPE

For those interested in advancing wildlife animal detection research, there are several promising avenues for future work. Firstly, exploring and implementing more advanced deep learning models beyond YOLOv8, such as YOLOv9 or other cutting-edge architectures, could enhance detection accuracy and efficiency. Additionally, focusing on refining preprocessing techniques to improve image quality and clarity could further optimise detection performance. Moreover, there's potential for refining segmentation and feature extraction methods to better distinguish animals from their backgrounds and capture relevant features.

Expanding the scope of datasets used for evaluation to encompass a wider variety of habitats, species, and environmental conditions would increase the robustness and applicability of detection methods. Integrating multimodal data sources, including audio and environmental sensor data, holds promise for enhancing detection capabilities and providing more comprehensive insights into wildlife behaviour. Addressing real-world challenges, such as varying lighting conditions and occlusions, through developing adaptive and robust detection algorithms is crucial for practical applications in the field.

Overall, future research in wildlife animal detection should aim to push the boundaries of detection accuracy, efficiency, and versatility while addressing real-world challenges encountered in wildlife monitoring and conservation efforts. By pursuing these avenues, researchers can contribute to developing more effective and reliable tools for protecting biodiversity and understanding ecosystems.

REFERENCES

[1] W. Yang, T. Liu, P. Jiang, A. Qi, L. Deng, Z. Liu, and Y. He, "A forest wildlife detection algorithm based on improved YOLOv5s," *Animals*, vol. 13, no. 19, p. 3134, Oct. 2023.

[2] R. Kavitha, N. S. Devi, N. Amirthakarthiga, and R. Srivarshini, "Wild animal detection using CNN," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 5, pp. 678–683, May 2023.

[3] S. Li, H. Zhang, and F. Xu, "Intelligent detection method for wildlife based on deep learning," *Sensors*, vol. 23, no. 24, p. 9669, Dec. 2023.

[4] S. Sajithra Varun and G. Nagarajan, "DeepAID: A design of smart animal intrusion detection and classification using deep hybrid neural networks," *Soft Comput.*, pp. 1–12, May 2023.

[5] J. Chappidi and D. M. Sundaram, ''Enhanced animal detection in complex outdoor environments using modified YOLO V7,'' *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 19, pp. 375–382, 2024.

[6] N. Sundaram and S. D. Meena, ''Integrated animal monitoring system with animal detection and classification capabilities: A review on image modality, techniques, applications, and challenges,'' *Artif. Intell. Rev.*, vol. 56, no. 1, pp. 1–51, Oct. 2023.

[7] P. Kumar, S. Luo, and K. Shaukat, ''A comprehensive review of deep learning approaches for animal detection on video data,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 11, pp. 1420–1437, 2023.

[8] C. Carl, F. Schönfeld, I. Profft, A. Klamm, and D. Landgraf, ''Automated detection of European wild mammal species in camera trap images with an existing and pre-trained computer vision model,'' *Eur. J. Wildlife Res.*, vol. 66, no. 4, p. 62, Aug. 2020.

[9] Y. Pulimi, S. R. Koppula, R. Katta, and R. Suryaneni, ''Wild animal detetction and alert system using YOLOV8,'' *Int. Res. J. Modernization Eng., Technol. Sci.*, vol. 6, no. 2, Feb. 2024. [Online]. Available: https://www.irjmets.com/

[10] N. Banupriya, S. Saranya, R. Swaminathan, S. Harikumar, and S. Palanisamy, ''Animal detection using a deep learning algorithm,'' *J. Crit. Rev.*, vol. 7, no. 1, pp. 434–439, 2020.

[11] J. Tang, Y. Zhao, L. Feng, and W. Zhao, ''Contour-based wild animal instance segmentation using a few-shot detector,'' *Animals*, vol. 12, no. 15, p. 1980, Aug. 2022.

[12] F. Schindler and V. Steinhage, ''Instance segmentation and tracking of animals in wildlife videos: SWIFT—Segmentation with filtering of tracklets,'' *Ecol. Informat.*, vol. 71, Nov. 2022, Art. no. 101794.

[13] C. C. Ukwuoma, Z. Qin, S. B. Yussif, M. N. Happy, G. U. Nneji, G. C. Urama, C. D. Ukwuoma, N. B. Darkwa, and H. Agobah, ''Animal species detection and classification framework based on modified multi-scale attention mechanism and feature pyramid network,'' *Sci. Afr.*, vol. 16, Jul. 2022, Art. no. e01151.

[14] B. Dave, M. Mori, A. Bathani, and P. Goel, ''Wild animal detection using YOLOv8,'' *Proc. Comput. Sci.*, vol. 230, pp. 100–111, Jan. 2023.

[15] Z. Zhang, Z. He, G. Cao, and W. Cao, ''Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification,'' *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 2079–2092, Oct. 2016.

[16] M. N. Rithvik, M. N. Agrawal, and P. Mona, ''Wildlife detection system using deep neural networks,'' *Int. Res. J. Eng. Technol. (IRJET)*, vol. 9, no. 7, pp. 729–734, Jul. 2022. [Online]. Available: https://www.irjet.net/archives/V9/i7/IRJET-V9I7139.pdf

[17] R. Gayathri, P. H. Princy, and P. Krishandhini, ''Wild animal intrusion detection system using YOLOv8,'' *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 5, no. 5, pp. 4583–4586, May 2023. [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2023/39491/final/fin_irjmets1684472145.pdf

[18] M. Tan, W. Chao, J.-K. Cheng, M. Zhou, Y. Ma, X. Jiang, J. Ge, L. Yu, and L. Feng, ''Animal detection and classification from camera trap images using different mainstream object detection architectures,'' *Animals*, vol. 12, no. 15, p. 1976, Aug. 2022.

[19] T. Rashid, M. Bin Hamzah, M. Rasheed, A. Jaber, M. Sarhan, M. Aldaraji, and T. Saidani, ''Image segmentation for animals in the wild using scilab software,'' *Al-Salam J. Eng. Technol.*, vol. 2, no. 2, pp. 72–77, Mar. 2023.

[20] R. Chandrakar, R. Raja, and R. Miri, ''Animal detection is based on deep convolutional neural networks with genetic segmentation,'' *Multimedia Tools Appl.*, vol. 81, pp. 42149–42162, Dec. 2022.

[21] K. Rancic, B. Blagojevic, A. Bezdan, B. Ivosevic, B. Tubic, M. Vranesevic, B. Pejak, V. Crnojevic, and O. Marko, ''Animal detection and counting from UAV images using convolutional neural networks,'' *Drones*, vol. 7, no. 3, p. 179, Mar. 2023.

[22] W. H. S. Antǒnio, M. Da Silva, R. S. Miani, and J. R. Souza, ''A proposal for an animal detection system using machine learning,'' *Appl. Artif. Intell.*, vol. 33, no. 13, pp. 1093–1106, 2019.

[23] G. K. Verma and P. Gupta, ''Wild animal detection from highly cluttered images using deep convolutional neural network,'' *Int. J. Comput. Intell. Appl.*, vol. 17, no. 4, Dec. 2018, Art. no. 1850021.

[24] H. Archana, K. Poornima, V. Raksha, H. R. Shwetha, and H. S. G. Supreeth, ''Artificial neural network based image processing for wild animal detection and monitoring,'' *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 5, pp. 687–691, May 2017. [Online]. Available: https://www.researchgate.net/profile/Supreeth-Hsg/publication/318931171_Artificial_Neural_Network_Based_Image_Processing_for_Wild_Animal_Detection_and_Monitoring/links/5ad1aa76aca272fdaf77ace1/Artificial-Neural-Network-Based-Image-Processing-for-Wild-Animal-Detection-and-Monitoring.pdf

[25] A. Abraham, B. Mathew, D. Panikkar, and J. John, ''Wild animal intrusion detection system using YOLO,'' *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 5, pp. 2165–2456, 2023. [Online]. Available: https://ijisrt.com/wild-animal-intrusion-detection-system-usingyolo

[26] T. Battu and D. S. Reddy Lakshmi, ''Animal image identification and classification using deep neural networks techniques,'' *Meas., Sensors*, vol. 25, Feb. 2023, Art. no. 100611.

[27] Ł. Popek, R. Perz, and G. Galiński, ''Comparison of different methods of animal detection and recognition on thermal camera images,'' *Electronics*, vol. 12, no. 2, p. 270, Jan. 2023.

[28] S. N. Kumar, A. L. Fred, and P. S. Varghese, ''Suspicious lesion segmentation on brain, mammograms and breast MR images using new optimized spatial feature based super-pixel fuzzy C-means clustering,'' *J. Digit. Imag.*, vol. 32, no. 2, pp. 322–335, Apr. 2019.

[29] S. Ray, ''Disease classification within dermascopic images using features extracted by ResNet50 and classification through deep forest,'' 2018, *arXiv:1807.05711*.

[30] Y. Kaya, M. Uyar, R. Tekin, and S. Yildirim, ''1D-local binary pattern based feature extraction for classification of epileptic EEG signals,'' *Appl. Math. Comput.*, vol. 243, pp. 209–219, Sep. 2014.

[31] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, ''UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios,'' *Sensors*, vol. 23, no. 16, p. 7190, Aug. 2023.

[32] Q. Liu, H. Ye, S. Wang, and Z. Xu, ''YOLOv8-CB: Dense pedestrian detection algorithm based on in-vehicle camera,'' *Electronics*, vol. 13, no. 1, p. 236, Jan. 2024.

[33] J. Chappidi and D. M. Sundaram, ''A comparative study of animal detection and classification algorithms, applications and challenges,'' in *Proc. AIP Conf.*, 2024, Art. no. 020020.

JOHNWESILY CHAPPIDI received the Diploma degree in computer science from the Dr. BR Ambedkar Government Model Residential Polytechnic, Rajahmundry, the Bachelor of Technology degree (Hons.) in computer science and engineering from the Lingayas Institute of Management and Technology, Vijayawada, affiliated with Jawaharlal Nehru Technological University, in 2013, and the Master of Technology degree (Hons.) in computer science and engineering with a specialization in digital image processing from the College of Engineering and Technology, Acharya Nagarjuna University, Guntur, in 2015. He was an Assistant Professor with the Department of Computer Science and Engineering, Paladugu Parvathi Devi College of Engineering and Technology, Vijayawada, from July 2015 to March 2018. He was the Senior Officer–IT and an OE with The Evangelical Fellowship of India Commission on Relief, Delhi, from October 2020 to 2021. Before that, he was an Assistant Professor with the Department of Computer Science and Engineering, Sasi Institute of Technology and Engineering, Tadepalligudem, from May 2018 to 2020. He has been a Research Scholar with VIT-AP University, Andhra Pradesh, India, since 2021.

DIVYA MEENA SUNDARAM received the B.Tech. degree in information technology from Vellore Institute of Technology (VIT), Vellore, India, in 2014, the M.E. degree in computer science and engineering from Anna University, Chennai, in 2016, and the Ph.D. degree from VIT University, in 2020. She was an Assistant Professor with the Jansons Institute of Technology, for a year. She was an Assistant Professor with Jain University, Bengaluru, before moving to VIT-AP University, Amaravati, in 2021. She is currently working as an Assistant Professor Sr. Grade 2 at VIT-AP University, Amaravati. In a span of three years' experience, she has published more than 45 research articles in SCOPUS and SCI. She has around 11 patents and one seed grant of 3.5 lakh to her credit. She is also guiding four Ph.D. scholars, of which two are international students. Her research interests include artificial intelligence, image processing, deep learning, thermal imaging, cloud computing, and remote sensing. She received the Best Researcher Award from 2017 to 2023.