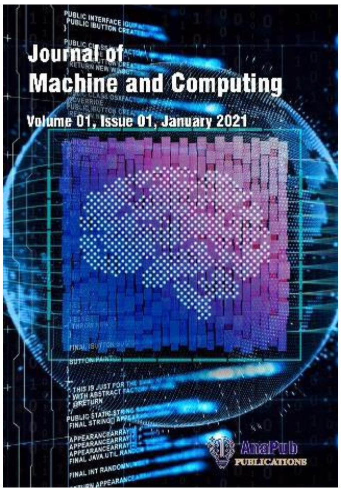
Improving Agricultural Safety Through Deep Neural Networks for Intrusion Monitoring



Thirupathi Battu and Lakshmi Sreenivasa Reddy D

DOI: 10.53759/7669/jmc202505131
Reference: JMC202505131
Journal: Journal of Machine and Computing.

Received 29 March 2025
Revised from 14 May 2025
Accepted 14 June 2025

Improving Agricultural Safety Through Deep Neural Networks for Intrusion Monitoring

Thirupathi Battu[1], Dr. D. Lakshmi Sreenivasa Reddy[2]

[1]Department of Computer Science & Engineering, University College of Engineering (A),
Osmania University Hyderabad, Telangana, India

[2]Department of Information Technology, Chaitanya Bharathi Institute of Technology (A),
Affiliated to Osmania University Hyderabad, Telangana, India

battuthirupathi2@gmail.com[1], dlsrinivasareddy_it@cbit.ac.in[2]

Orchid Id: 0000-0002-8380-9264

Abstract

The intersection of agricultural activities and natural habitats presents several implications, one of the most pressing being the intrusion of various species into crop fields. To address the continued and growing challenge of discovering methods for successful mitigation and protection of crops, a new method must be introduced. Current practices, such as using pheromones and monitoring by humans, yield suboptimal results due to inefficiency, labour-intensive processes, and environmental damage. This paper proposes a novel method based on deep learning technologies, namely UNet and EfficientNet-B7 architectures, for semantic segmentation of crop-damaging species in agricultural habitats. The method provides a reliable means to detect species intrusion by integrating it into automated monitoring operations. Whether identifying the segmentation point of target species or utilizing feature-based detection with the proposed model, the application of advanced convolutional neural networks ensures the accuracy of these systems. This method demonstrates higher performance in species identification and promotes newer and ecologically friendly methods of implementing non-invasive monitoring for conservation alongside agriculture. The experimental results confirm that the proposed UNet with EfficientNet-B7 achieves a high accuracy ratio of 95.3%, an F1-score ratio of 92.4%, and an efficiency ratio of 97.6% compared to other existing models.

Keywords: Intrusion, Deep Learning, Semantic Segmentation, Convolutional Neural Networks, EfficientNet-B7, Unet

1. Introduction

Agriculturalists worldwide face various difficult problems that result from the simultaneous existence of farming and wildlife. One of which is when wild animals invade fields; as a result, they damage crops and become a source of monetary loss [1]. As previously stated, this relationship between people and animals can grow into a full-fledged conflict that negatively impacts farming production and wildlife conservation efforts [2]. Implementing successful animal identification and monitoring approaches on farms is critical to combat this issue.

There are many challenges in detecting and watching wild animals in agro landscapes [3]. Traditional methods, such as human observation or physical obstacles, are often time-consuming, expensive, and fragile, as they do not efficiently deter animals from restricted areas due to the highly dynamic behaviour of the animals and the huge areas of agricultural lands

[4]. Additionally, identifying and treating invaders must be done swiftly and with non-invasive and eco-friendly practices.

Furthermore, aside from the issues discussed above, there has also been a rise in novel awareness of the sustainability requirement. The latter is crucial for eliminating conflicts between animals and people in farming domains [5]. Commonly used approaches to deter animals, including fencing and using chemical repellents, harm wildlife and the environment [6]. As a result, innovative, environmentally friendly, innovative, innovative technological solutions that can efficiently solve the abovementioned issues are required.

The use of deep machines helps to alleviate the problem of identifying and monitoring wild animals in farms. It involves the application of deep machine, a form of machine learning that employs neural networks comprising multiple layers to decode complex patterns and associations from data. Scientists can use such models to model sophisticated prediction methods that can accurately and efficiently detect and monitor wild animals within farms [7].

One of the applications of deep learning in identifying wild animals is the analysis of remote sensing data; it may include satellite imagery or drone recordings. Indeed, this bird's-eye view of agricultural fields can easily detect animal intrusions [8]. Subsequent analysis of recorded data by deep learning may identify animals as separate objects from other elements, thus detecting wildlife presence automatically.

Another solution is to use sensor networks with cameras and various environmental sensors across agricultural areas. In particular, these sensors allow for monitoring animal activity, such as movement or behaviour. Thus, the data is obtained much faster, and deep learning algorithms can rapidly analyze it to immediately identify and track any potential threats posed by wild animals to agriculture and act accordingly by farmers or wildlife managers. In conclusion, combining deep learning with remote sensing and sensor networks addresses the problem of wildlife detection and monitoring in agricultural areas and ensures a balance between agriculture and wildlife conservation.

The main contribution of the paper is:

- Designing the UNet and EfficientNet-B7 architectures for semantic segmentation of a wild boar in an agricultural habitat.
- Identifying the segmentation point of a wild boar or utilizing feature-based detection with the proposed model, the application of advanced convolutional neural networks ensures the accuracy of these systems.
- The experimental outcomes were performed, and the suggested UNet-EfficientNet-B7 model increased the tracking accuracy ratio, efficiency ratio, F1-score ratio, and training and testing loss compared to other existing models.

The rest of the paper is prearranged as follows: section 2 discusses the literature survey, section 3 proposes the UNet and EfficientNet-B7 model, section 4 deliberates the results and discussion, and section 5 concludes the research paper.

2. Literature Survey

Bijuphukan Bhagabati et al. [9] highlighted creating a real-time detection and classification system of wild animals in video feeds and notifications to prevent avoidable interactions between humans and animals. The system detects wild animals in real time by implementing deep learning models and YOLOV5 with the SENet attention layer. To shorten the time used for manual labelling process creation and adequate trial and training data, it was conducted by blending open source and bespoke datasets from various animal species. A cloud-based artificial intelligence system was implemented in the cameras to take photos from various KNP areas and validate the model's efficacy.

Kristina Rancic et al. [10] presented many state-of-the-art network architectures trained on a well-annotated picture collection to predict the existence of objects in the dataset. Three permutations of the You Only Look Once, and a Single Shot Multi boxDetector architecture identify deer in heavily wooded environments. The efficacy of these models' effector information, including mean average precision precession, recall, and F1 score, was reviewed. These models are also assessed based on their proficiency in real time.

Axiu Mao et al. [11] presented the research series concerning Animal Activity Recognition (AAR) using wearable sensors and deep learning algorithms. The text outlines sensor kinds often used and animal species and behaviours commonly examined. Moreover, the text gives an exhaustive examination of deep learning technologies utilized in wearable sensor-enhanced AARs. The division of technologies is made according to the classification of deep learning algorithms. The following paragraphs give a wide range of publicly available datasets produced for the past five years using a wearable and augmented reality sensor. This knowledge would be beneficial for potential researchers. Further, the article discusses the potential difficulties of deploying a deep learning model in the subject area. Additionally, the article presents the alternatives and suggests follow-up research. The analysis herein provides feasible outcomes in enhancing the current AAR systems' efficiency through wearable and wearable sensors. The automatic system, which gathers measurable results combined with the veterinarian professional's subject view, shows high potential in enhancing the species' health and well-being.

W. P. S Fernando et al. [12] suggested a comprehensive approach to deter animals by considering many characteristics such as color, coat pattern, morphology, and diurnal and nocturnal vocalizations. Moreover, it can detect the number of animals approaching crops and monitor animal behaviour to prevent inaccurate alerts. For these objectives, the research uses several methodologies, such as image processing and deep learning, to examine aural, visual, and pictorial data obtained from particular animal populations.

B. Natarajan et al. [13] presented a novel model that integrates the VGG-19 architecture with a Bi-LSTM network. The model is designed for animal identification systems and alert generation based on detected activities. The systems ensure that Short Message Service (SMS) are sent to the nearest forest office for rapid responses. The proposed model has shown more progress in general performance.

S. Sajithra Varun et al. [14] presented a new method to identify animals by employing new advancements in Deep Learning algorithms. The proposed solution is implemented by utilizing deep learning technology to spot animals early on and take security measures to eliminate harm due to animals. This work integrates image processing and artificial intelligence techniques to spot animals, recognize their species, and identify them automatically. CNNs are utilized for image recognition purposes. It also includes an alert module and animal-repellant circuitry. The proposed methodology experimented with datasets meticulously created for animal validation, including Amur Tiger Re-identification in the Wild, Animals Detection Images Datasets, and Google Open Images V6+. Deep CNNs are utilized to extract data from intricate optical pictures. Typical camera trap databases are used to create hybrid CNN models. After the classification process, the retrieved CNN features are input into highly efficient deep learning algorithms. As a result of this process, it was observed that the performance was of better quality than several other publications.

Ashwini V. Sayagavi et al. [15] developed a novel, efficient, and reliable computer vision method for automatically identifying wild animals. The technique utilizes the YOLO object detection paradigm to detect the existence of untamed creatures in photos. More precisely, the approach is designed to identify six specific categories: humans and five separate species of animals (elephant, zebra, giraffe, lion, and cheetah). Once animals are detected, their movements are tracked using CSRT to determine their intent. Based on this information, alerts are sent to notify the appropriate authorities. In addition, the article describes the creation of a prototype for this suggested approach, using Raspberry Pi devices that are outfitted with cameras.

Aby K Thomas et al. [16] suggested a solution by combining the Internet of Things (IoT) with Machine Learning (ML) approaches. Image processing and IoT sensor networks have led to significant sensor advances. The problem of animal-human conflicts in agricultural areas and forest zones is substantial, endangering human lives and resulting in substantial financial losses. A wireless sensor-based animal incursion detection system may be created by analyzing video clips from a given dataset. The current study presents the utilization of the SlowFast architecture for Invariant Feature Extraction. Video annotation is performed, and spatial data is extracted using the IFE model. A novel type of monitoring may be conducted if the animals are classified by their characteristics based on the visual analysis. It is feasible to lower animal-vehicle collisions, improve animal monitoring, and prevent theft via detection and classification methods. The usage of deep learning methods is vital for effective implementation.

Sibusiso Reuben Bakana et al. [17] suggested the lightweight and efficient wild animal recognition model (WildARe-YOLOv5). The model uses Mobile Bottleneck Block modules and an enhanced StemBlock to reduce the computational cost of model parameters and backbone FLOPs. The author utilizes Focal-EIoU as a loss function and a BiFPN-based neck to evaluate predicted bounding box accuracy during inference. The author tried on Wild Animal Facing Extinction, Fishmarket, and MS COCO 2017. The baseline model showed a 17.65% gain in FPS, 28.55% model parameter reduction, and 50.92% FLOP reduction compared to state-of-the-art deep learning models.

Subraja Rajaretnam and Varthamanan Yesodharan [18] recommended the deep batch normalized exponential linear unit AlexNet (DbneAlexnet) and the Gazelle Hunting Optimization Algorithm (GHOA) for wild animal recognition. Simulate the IoT-Multimedia Sensor Networks (WMSN) network, with IoT nodes gathering wild animal identification photos. The recommended GHOA routes photos to the base station. Image pre-processing at the BS uses the Weiner filter (WF) to eliminate noise from the raw wild animal picture. Salient map extraction uses the denoised output to identify evident areas in the range of view and guide placement selection. The saliency map is then sent to DbneAlexnet, who is trained using the suggested GHOA to identify wild animals. Combining the Gazelle Optimization Algorithm (GOA) with the deer hunting optimization algorithm yields the GHOA algorithm. Precision, recall, and f1-score show 90.2%, 89.1%, and 89.6% for detection.

Diego Bárbulo Barrios et al. [19] introduced the convolutional neural networks implemented on thermal UAV imagery for monitoring mammalian herbivores. Thermal Multi-Object Tracking and Segmentation (MOTS) in UAV images tracks mammalian herbivores in this work. Our study developed and assessed a cutting-edge MOTS algorithm (Track R-CNN) for dairy cow segmentation, identification, and tracking. A UAV with a thermal camera collected data in two farms at various angles and heights in overcast/sunny and 16.5 °C circumstances. We found that dataset variety and balance, particularly considering the circumstances under which the data was acquired, might improve tracking efficiency in certain cases. Transfer learning was utilized to migrate knowledge for algorithm training. Our best model (68.5 sMOTSA, 79.6 MOTSA, 41 IDS, 100% counting accuracy, and 87.2 MOTSP) using 3D convolutions and an association head shows that Track R-CNN can detect, track, and count herbivores in UAV thermal imagery under heterogeneous conditions.

Fei Chen et al. [20] presented ConservationBots as autonomous aerial robots for fast, robust wildlife tracking in complex terrains. While avoiding possible animal disruptions, the aerial robot demonstrates strong localization performance and completes tasks quickly, which is important for airborne systems with limited energy. Our method addresses the practical and technical issues that arise when using a lightweight sensor in conjunction with new ideas, such as: (i) using an information-theoretic objective to plan trajectory and measurement actions, (ii) developing a bearing detector that is more resistant to noise, and (iii) formulating a tracking algorithm that is resilient to missed and false detections encountered in real-world scenarios. This enables the robot to strategically choose near-instantaneous range-only measurements for faster localization and time-consuming sensor rotational actions to acquire bearing measurements and achieve robust tracking performance.

Akanksha Mishra and Kamlesh Kumar Yadav [21] suggested the Smart Animal Repelling Device (SARD) for Anti-Adaptive Harmful Animal Deterrence. Resource management in Edge or Fog environments is presented in this paper using a complete distributed system. Using Docker containers, the SARD framework deploys Internet of Things (IoT) apps as microservices, capitalizing on the containerization concept. Including several Internet of Things (IoT) applications, resources, and power management tactics for fog and Edge computing systems in the proposed software system is possible. The experiment results show that the AI system can successfully recognize animals using computational approaches that are efficient with power. The system's ability to satisfy the needs of anti-adaptive hazardous animal deterrence in real-time is guaranteed by this implementation, which also keeps the mean average accuracy high at 93.25%.

Based on the survey, there are several issues with existing models in attaining high tracking accuracy, efficiency ratio, and F-score ratio. Hence, this study proposes the UNet and EfficientNet-B7 architectures for semantic segmentation of a wild boar in an agricultural habitat.

3. Proposed Method

Wild animals' invasion of agricultural fields is a serious problem that endangers farmers' crops, their livelihoods, and their families. Manual surveillance or simple sensor-based systems are conventional but labour-intensive, inaccurate, and wasteful, particularly on a big scale. Issues such as fluctuating light levels, thick foliage, fog, or severe rain make it difficult for these conventional methods to provide reliable and precise detection. Even more challenging for detection and tracking are the habits and motions of animals, including posture changes, occlusions, and quick movements. Simple automated solutions that use machine learning or image processing have a long way to go. In real-time situations, especially in settings with limited resources, many models may not work because they need too much computing power. Moreover, they don't always have the resilience to deal with different kinds of animals, their unexpected movements, and environmental obstacles, which may lead to inaccurate tracking, missed detections, or false positives. This study proposes the integration of the Unit with EfficientNet-B7 for wild boar semantic segmentation, a major innovation in computer vision. The integration of the strong architecture of the Unit with the better feature extraction ability of efficientNet-B7 can offer high accuracy and efficiency in wild boar recognition and segmentation.

3.1 UNet

The U-Net architecture is a convolutional neural network specifically developed for semantic segmentation tasks but is better suited to applications within the medical imaging analysis field. It consists of a contracting path, the encoder, while the other is the expanding path or decoder. The primary function of the name encoder is to obtain the contextual information and features from a given input image. The function of the decoder is to provide the spatial detail and the segmentation map. For this reason, the network can calculate the exact positional prominence of the items within the image at a given time.

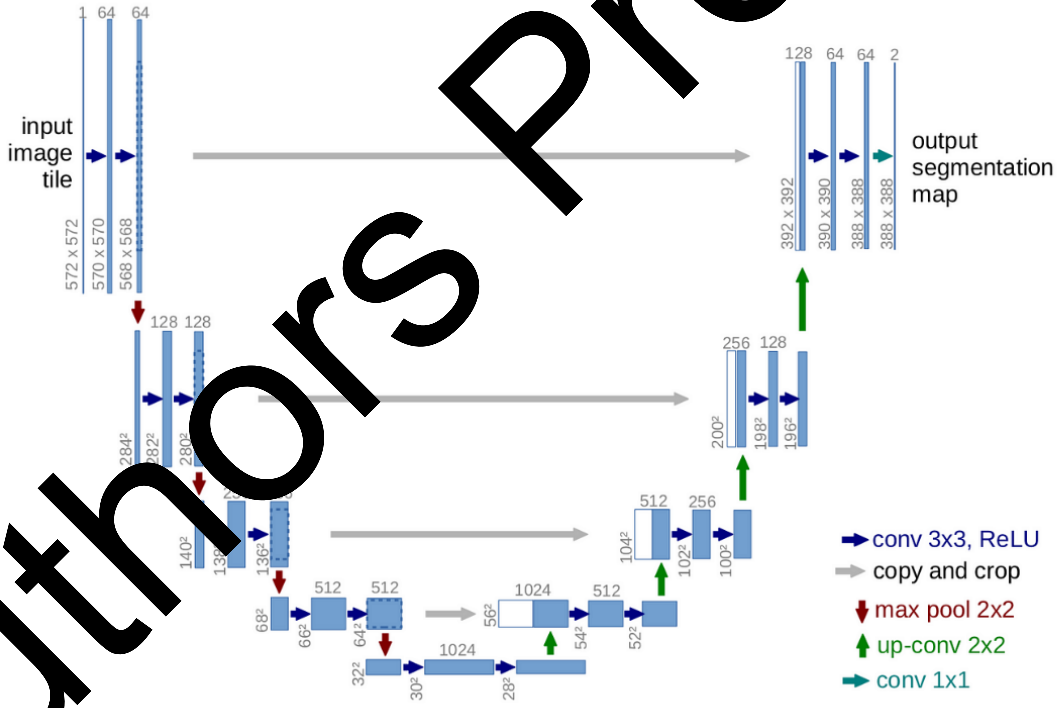


Figure 1: UNet Architecture

The specifics of the layers incorporated in the UNet architecture:

- **Encoder Layers**

The first layer of the UNet architecture is the encoder, which consists of a series of convolutional layers and pooling layers. They are made to draw highly complex features from the input picture while reducing its spatial size. The convolutional layers utilize filters to draw features from the digitized input image, whereas the pooling layers decrease the spatial size while maintaining important information about the input feature. The downsampling procedure aids in capturing contextual information for broader receptive fields.

- **Bridge Layer**

After several rounds of downsampling during the encoder route, the most common practice is adding a bridge layer that connects the encoder and the decoder pathways. The bridge layer is created to maintain a high-detailed representation of the features captured by the encoder as it moves to the decoder and helps localize accurately. Normally, it has additional convolutional layers that refine the captured features.

- **Decoder Layers**

The decoder of the UNet is designed to increase the resolution of the encoder's obtained features to produce the final segmentation map. The decoder is designed as a set of upsampling layers that are then followed by layers of convolution. The dimensions of the feature maps in terms of space are increased gradually in these layers where the spatial information is not lost. The upsampling layers would increase the original input image's resolution feature map match. This allows for better localization of the objects and borders within the segmented image.

- **Skip Connections**

The critical innovation from the UNet architecture is the introduction of skip connections that join the corresponding levels of the encoder and decoder paths. Skip connections allow the exact spatial information to be passed from the encoder to the decoder, allowing the fully connected layer to recover finely detailed features lost in the downsampling phase. The UNet architecture accurately localizes objects in the segmented image by selecting features of various resolutions, using the encoder context as a reference for matching.

- **Output Layer**

The output layer is the termination of the output route and is positioned at the finish of the decoder. It is a convolutional layer programmed with a convolutional operation and activates with a sigmoid or a softmax function as its primary nature. The segmentation map is delivered as a pixel label for the entire input picture, enabling pixel labelling to recognize regions of interest in the input like pictures, medical-building instruments, and more.

A UNet comprises the Encoder, Bridge, Decoder, Skip connections, and output layers. Each is critical in feature extraction, downsampling, upsampling, and localization, which is valid for any segmentation work. As a result, it remains to be the best model in different works, especially in biomedical imaging. In addition, in corporate activities that demand the perfect anatomy and tissue demarcations to be generated to help process interventions, such an architecture is heavily used.

Advantages of Unet

- **Architecture in the form of U:** The U-Net design consists of a contracting route, which is composed of downsampling, followed by an expanding path, which is composed of upsampling. The architectural design portrays the shape of the letter "U" . This enables the accurate identification of certain characteristics, thus creating detailed divisions or segments.

- **Skip connections:** Within U-Net, equivalent layers in the contracting and expanding routes are linked through skip connections. Skip connections enable the network to retain high-resolution information generated by the contracted route, facilitating the reconstruction of spatial information discarded during downsampling.
- **Reduced vanishing gradient problem:** The U-Net's skip connections help reduce the vanishing gradient problem, a common problem faced by deep neural network models during training. These skip connections establish direct links between the initial and ending layers, which improves the flow of gradients across the network layers, encouraging quicker convergence and easier training.
- **Efficient use of training data:** The U-Net optimally utilizes the training data through data augmentation techniques and transfer learning, which decreases the number of annotated training samples demanded compared to other segmentation architectures. It is especially useful when the labelled data is scarce or financially expensive.
- **Versatility in picture size:** U-Net is flexible and can be trained with images of different sizes. Based on this, it is critical to work with medical data, as the images might contain significant disparities in quality and size.
- **Cutting-edge performance:** U-Net has achieved state-of-the-art results on a variety of medical image segmentation projects through training on a limited number of samples, including cell segmentation, organ segmentation, tumour identification, as well as lesion contours.
- **Rapid inference:** The U-Net architecture allows for very fast inference times, making it a good choice for any application that requires real-time or near real-time processing, e.g., image guidance for surgery or when rapid decisions are needed, such as for diagnostics.
- **Versatility across domains:** Originally, the U-Net was developed to work on medical image segmentation, but it found applications in other areas, such as satellite image analysis, industrial inspection, and even autonomous driving. This serves as an indication of its effectiveness and potential for diverse applications.

In conclusion, the U-Net architecture is a powerful technique for various segmentation operations in which the precision of object localization and segmentation in the picture is crucial. This is accomplished via an effective implementation of skip connections, one of the few designed to be U-shaped, and rapid training methods.

3.2 Efficientnet-b7

EfficientNet is a family of convolutional neural network designs designed to provide optimal performance with minimal computational cost. The label "EfficientNet-B7" is used to identify an individual member of this family by its exact depth, width, and resolution scaling parameter values.

3.2.1 MBConv Block

The MBConv block is a central building block of the EfficientNetB7 architecture, a state-of-the-art convolutional neural network designed for rapid and effective image classification use cases. Let's look at and scrutinize every part of the MBConv block:

Expand: This is the first step of the MBConv block operation. The number of channels or the existing feature mapping in the input tensor is increased. Normally, a 1×1 convolutional layer with many output channels is used to achieve this compared to the output. The main aim of this expansion is to allow the model to capture a wide variety of diverse features from the input.

Depthwise Convolution: Following the expansion stage, the expanded tensor was subjected to a depthwise convolution, in which each port retained only one convolutional filter. During a standard convolutional layer, all filters must engage with all input ports on the way out. On the contrary, depthwise convolution employs just one filter for each port. This method substantially lowers the computation expense while effectively obtaining spatial characteristics.

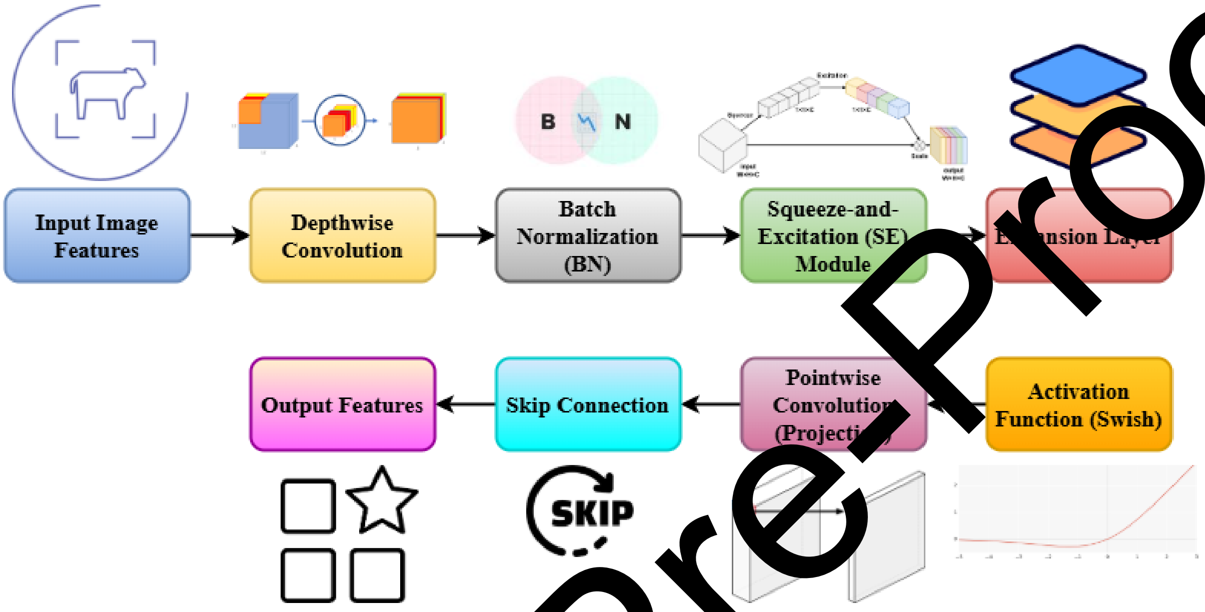


Figure 2: MBConv block

Squeeze: A squeeze operation is applied after the depthwise convolution is executed. Usually, it operates through global average pooling, which covers the spatial information over all channels by computing the resulting average for each channel. The result is the tensor of reduced spatial dimensions and preserved per-channel knowledge.

Reshape & Reduce: After the squeeze stage, the output is then reshaped and reduced at this stage. The primary task during the reshaping process is to transform the tensor into a smaller size or reshape it into a new shape if possible. Following the reshaping process, the channels need to be reduced further. Typically, the reduction process involves 1x1 convolutions. This reduction enables the next stage to reduce the information while maintaining its high representational capacity.

Excite: The last operation in the MBConv block, the excitation procedure aims to boost significant characteristics while reducing less critical ones. This is done by frequently recalculating the feature maps to reweight them during inference dynamically. The process often necessitates the collection of attention weights for each channel and their application separately to the feature maps to raise the importance of the dominant channels and minimize the importance of less essential channels. However, this operation's stimulation aids the model in focusing more on essential characteristics and improves its separation capability.

In short, the MBConv block executes these operations in a well-designed order that effectively collects and cultivates unique features inside the input tensor. At the same time, it minimizes the computational expense and model parameters. Even if it has already become known as a highly successful design pattern in many picture classification undertakings, it hangs on to a high level of efficacy.

3.2.2 Architecture of EfficientNetB7

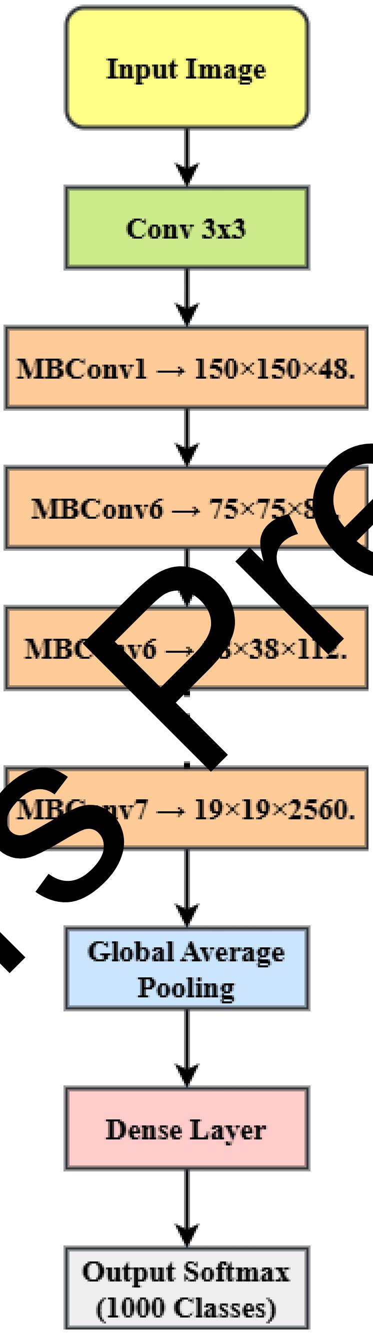


Figure 3: Architecture of EfficientNetB7

- **Input Layer:**

The EfficientNet-B7 neural network’s input layer is the first data layer, and the model’s lead layer is concerned with how the raw input data, usually in the form of pictures, is fed into the ensuing layers of the model. The input layer is critical because it defines the format and configuration of the input data the network expects to receive, whether in learning or performance. In contrast to other CNN designs, EfficientNet-B7 has a flexible input layer that takes pictures with disparate dimensions and aspect ratios.

In particular, EfficientNet-B7 uses AutoML, which is a method that automatically searches for the best-performing input picture dimension based on available machine computation resources for similar models. This process involves resizing the incoming photographs into various dimensions and checking the MPAs produced by the resulting models. The final best-balanced accuracy/efficiency scale image resolution is chosen as the input picture dimension the model expects to receive. Once decided, the input layer awaits pictures of that size, frequently expressed in height, width, and colour channel number. For instance, if the dimensions for the input layer are set at 600 x 600p, the layer expects a 600p high and 600p wide picture with pixels at three different colour coordinates: red, green, and blue. During learning and performance, the input layer pre-processes the input pictures.

This process gets input pictures to fit most models' height, width, and color pixel integration settings. The process also includes scaling, normalization, and augmentation and is designed to help the model learn more robust and consistent features towards more accurate predictions. Therefore, the efficient net-b7 input layer is instrumental in enforcing the input data format, determining the ideal input picture dimension, and making the input pictures ready for easy learning by the rest of the network. Compared with traditional CNN designs, EfficientNet-B7 is highly adaptable and scalable, responsive to varied and asymmetrical input resolution, and runs more diverse picture sizes. Thus, it is ideal for most computer vision loads.

- **Stem Convolutional Layers:**

The Stem Convolutional Layers of the EfficientNet-B7 architecture play a critical role in the initial pre-processing of input images. It marks as a gateway for the neural network, and its primary task is to extract the underlying patterns in input images while compressing them to simplify the analysis in the subsequent levels. The stem convolution layers comprise several processes designed to identify the rudimentary elements such as edges, patterns, and colours. It was necessary to lay the foundation for the network to progressively learn more complex and abstract patterns as the data travelled through multiple levels.

The input image is subjected to convolution with filters at the initial stages. Present as small matrices, the filters run over the image and perform element-wise multiplication and summation. Since the process is done across all spatial positions, it helps identify patterns occurring at various places. Initially, convolutions have small design dimensions to detect complex elements present in input images. After the initial convolution processes, additional layers, such as pooling or striding, can be applied to reduce the feature maps’ size in terms of spatial dimensions.

The purpose of downsampling is to reduce the computational load for the network, additionally boosting its receptive field, enabling it to capture more of the correlation and global trends in data. The design of stem layers on EfficientNet-B7, thus, is very well thought out, appropriately balancing computational efficiency and representation capacity. Most convolutional layers used in stem layers have reduced computational costs without much compromise on performance. Methods such as depthwise separable convolutions are used to factorize typical convolutions into independent operations for spatial and channel-wise dimensions.

- **EfficientNet Blocks (MBConv Blocks):**

The EfficientNet Blocks, also known as MBConv Blocks, are crucial parts of the EfficientNet design intending to generate a balanced model that combines efficiency and effectiveness. These blocks are crucial for achieving state-of-the-art levels with far fewer parameters and computation costs than traditional convolutional neural networks.

The MBConv block is achieved by using depthwise separable convolution. Depthwise separable convolution originates from the idea of separating the traditional convolution into two – the depthwise and the pointwise convolution layers. This separation makes it far more computationally efficient by reducing the number of parameters in the parameters and computing the spending pool while preserving training capacity. Depthwise convolution is a stage in which convolution is separately conducted utilizing single filters supplied to each input channel of the feature map. It results in spatial data. Find the information from each solitary filter and each of the input channels. The computation and operations manage to be far cheaper than traditional input controls. In the subsequent phase, termed pointwise convolution, the channel count of the feature maps is increased, followed by the utilization of 1x1 convolutional filters on the entire input volume.

The final step is the ReLU activation function, a linear activation used in contemporary neural networks. Increasing channels thus allows the network to understand harder and interpretative features while maintaining computational efficiency. Additionally, MBConv Blocks frequently have extra modifications to aid feature representation further and improve the model file. A common addition to MBConv Blocks is using a squeeze-and-excitation block in each MBConv Block. This block assists by dynamically recalibrating the feature responses by learning to focus on informative feature channels. This is achieved by allowing the module to emphasize different feature planes, hence learning how to talk to different kernels and layers with given weights.

- **Global Average Pooling Layer:**

The Global Average Pooling Layer is a component in many CNN architectures, including EfficientNet-B7. The main purpose of this step is to convert the spatial data in the feature maps produced by the last convolutional layers into a compact form suitable for classification or regression. The feature maps produced by the latest convolutional layer contain a lot of information regarding the distinct spatial patterns of the input image after many convolutions and activations. It is necessary to reduce the spatial dimensions of the feature maps before they can be piped into fully connected layers for classification while maintaining the essential information. This drop can reduce the number of parameters in the succeeding layers, preventing overtraining and cutting the computational cost.

The mean value of each feature map over its spatial dimensions is calculated to reduce the dimensionality of the feature maps in the Global Average Pooling Layer. Global average pooling is implemented differently from ordinary pooling layers, even though it computes both the max and average pooling over a predefined kernel size and stride. For each channel of the feature map, the global average pooling layer calculates the mean activation value by adding the activation values to all the spatial places within the channel and dividing by the total number of spatial locations.

The outcome is a feature vector in which each element is equivalent to the average activation value of specific feature map channels. Global average pooling calculates the mean value of its feature map's overall spatial dimensions. Such a pooling operation will include only critical information and rarely consider inconsequential geographical data. The pooling method suggested in this concept is extremely beneficial because it maintains the geographical context while significantly diminishing the feature representation’s dimensionality. These properties allow for better computational efficiency and less overfitting.

Furthermore, enhancing the network’s ability to remain invariant to translations through the operation of a global average pooling layer reduces the dependence of the output of this layer on the exact position of the features in the feature maps. This addresses the network’s ability to translate, rotate, and distort the input image in its ability to learn and improve its performance.

- **Fully Connected (Dense) Layer:**

Fully Connected Layer, known as the Dense layer in neural network topologies. Its central role is to capture intricate patterns from the information aggregated by its predecessors. This is the high-level reasoning component of the network; it is critical for classification, regression tasks, and image recognition. A fully Connected Layer is one where every neuron in the current layer is connected to every neuron in the next layer, producing the dense matrix. It is so densely connected that it allows capturing nuanced, effective dependencies between features. Therefore, the network can better understand complex patterns.

In the EfficientNet-B7 design, the Fully Connected Layer is often fed by the Global Average Pooling Layer. After global average pooling, the feature maps are flattened to a one-dimensional array or list, used as input for the dense layer. Every neuron in the densely connected layer makes an individual computation to the layer in mathematics. Every neuron in the dense layer multiplies its inputs by its corresponding weight, sums them all together, and then all are passed through an activation function to bring the non-linearity into the play. The weight and bias used for each neuron are tuned during the training by backpropagation. Backpropagation is an optimization technique for a given parameter. It prepares the network, which helps the network modify them so that the discrepancies between the expected and actual output are minimized. Our network uses this phase to modify its parameters to understand the underlying patterns in the training dataset.

In the case of a classification task, the Fully Connected Layer output is frequently passed through a softmax activation function. A softmax transforms the raw output values into probability distributions over the given number of classes. The network can generate predictions by selecting the class with the highest probability.

- **Output Layer**

The last layer in a convolutional neural network is the Output Layer. The model offers predictions or outputs through these channels. The design and structure of the model should be used to achieve a specific purpose, such as classification or regression.

For classification, and since EfficientNet-B7 is good at image classification, the output layer generally has a dense layer followed by a softmax activation function. They are mainly used to convert the features extracted by the following layers to the class probabilities of the different classes. Every neuron in the output layer is associated with a given class. The softmax function is later used to convert the raw output scores for all classes into a standardized probability distribution, where the probability of each class is represented. The model can then relate the probability of input forms part of the class.

For example, if the model learns to classify pictures into 1000 distinct classes in the image data, such as imageNet, the output layer contains 1000 neurons in which each forms a class. Softmax activation ensures that the sum of class probabilities equals 1, which makes it possible to understand how confident the model is in making the prediction. In a regression task, where the goal is to predict a continuous value, the output layer can contain a single neuron with a linear activation function. For example, if the task is to predict the house price based on the house features, the output layer provides a single output representing the price.

It is also crucial to note that the output layer acts only to generate predictions. Still, the final results are obtained by compiling the information extracted from the rest of the previous layers, notably the convolutional layers, pooling layers, and fully connected layers. The output layer merely gathers these features and passes them through another transformation to make final predictions.

3.3 Proposed combination of Unet Efficientnet-b7

Integrating Unet and EfficientNet-B7 proposed for wild boar semantic segmentation is a new method in computer vision and image processing. Unet is a well-established architecture for semantic segmentation. It is popular because it creates accurate segmentation of images by semantically determining pixels. Structurally, it is a composition of an encoder-decoder, in which the original object sample extracts the local features, and the other one receives the global features. Such structure is beneficial in the recognition and segmentation of objects. EfficientNet-B7 is a convolutional neural network state-of-the-art architecture. This model achieved substantial results in various image classification fields. Moreover, it is characterized as a model that can utilize the computational power most effectively while having a small model size. This is achieved by the uniform scaling method, which scales the network's depth, width, and resolution equally.

Combining the above two modules, Unet and EfficientNet-B7, can bridge the strengths of both architectures. The model proposed in this paper is expected to provide a more accurate and fast approach to semantic segmentation for wild boar. The EfficientNet-B7 was utilized to obtain the high-level features of the input images. In contrast, Unet was applied through the encoder-decoder path to enhance the features across the upper layers, resulting in an extraordinary semantic segmentation mask.

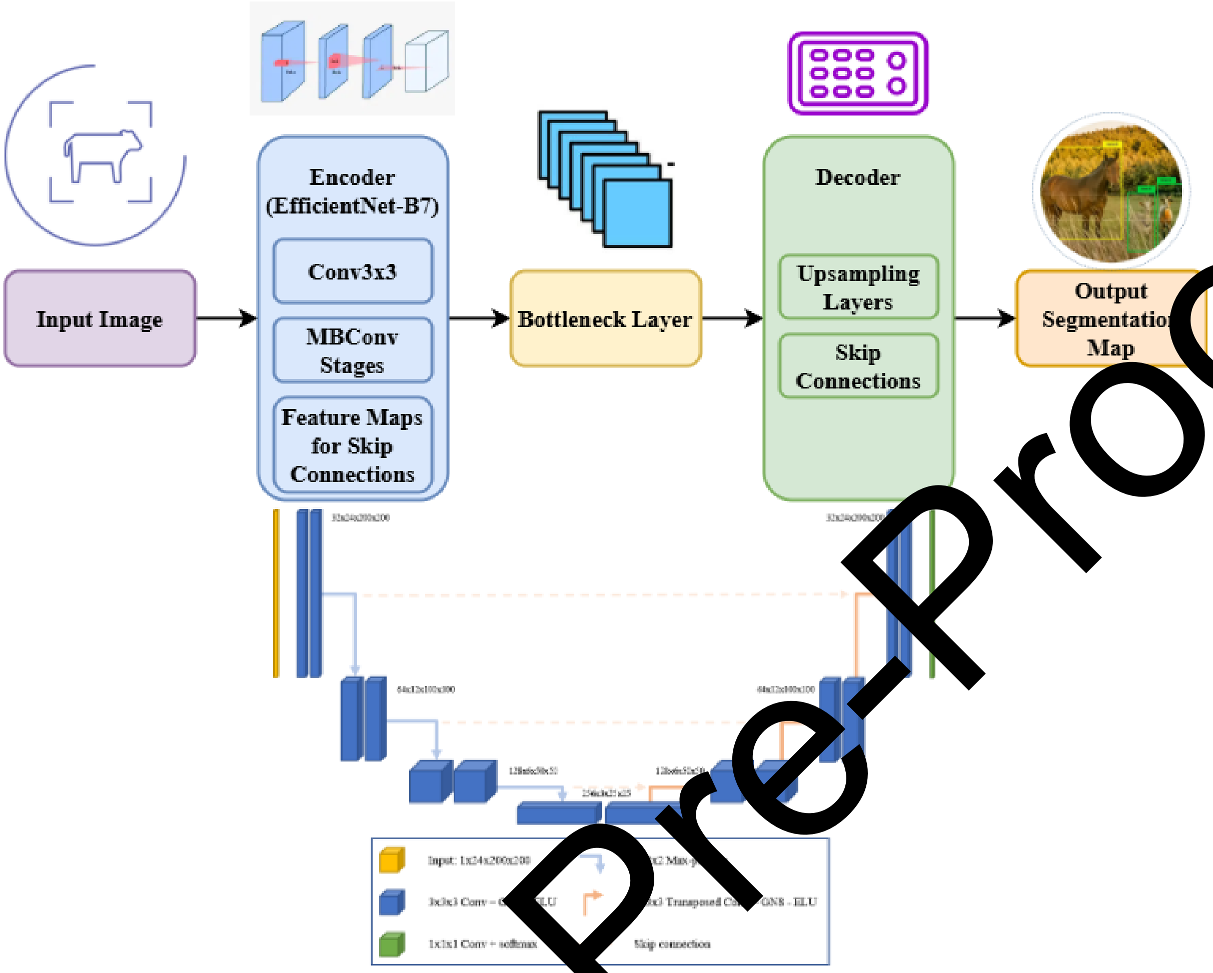


Figure 4: Architecture of proposed UNet with EfficientNetB7

- **Conv 2D**

Convolutional layers are central to the Convolutional Neural Networks (CNNs) design, as they are specifically developed for feature extraction. To perform this function, they create filters processed with incoming data to reveal features through convolution. Filters shifted horizontally and vertically over small items of the inputs produce feature maps. I assume that "Conv 2D" suggests regular 2D convolutional layers that extract only the necessary feature data from the input image.

- **MBConv 3x3**

MBConv, short for Mobile Inverted Bottleneck Convolutions, first emerged in the EfficientNet family. Core components of MBConv may be divided into three distinct stages: depthwise convolution, expansion convolution, and pointwise convolution. This MBConv 3×3 means the depthwise convolution layer used a 3×3 kernel for the feature convolutions. They have constructed the convolutional layers as a kernel that is either extremely efficient in computing or very successful at learning complex patterns from data.

- **MBConv 5x5**

MBConv with a 5x5 kernel utilizes a larger kernel size (5x5) for the depthwise convolution process, similar to MBConv with a 3x3 kernel. This modification enables the model to include broader spatial features in contrast to its 3x3 counterpart, perhaps enhancing its ability to detect patterns over greater regions in the input data.

- **Up Conv 2D**

Up-convolutional layers, also referred to as transposed convolutional or deconvolutional layers, are architectural components that have been specifically developed for semantic segmentation. Up-convolution is the fundamental method of increasing the resolution of feature maps, thus allowing the network to predict the label for every pixel densely. According to this architecture, the operation is called "Up Conv 2D" and is meant to upsample. This is done to maintain the model's ability to appropriately determine the exact borders of a segmentation.

- **Concatenation**

Concatenation is the process that is considered to be crucial for neural network topologies where models are created based on skip connections, such as for U-Net and its variations. The process is combining feature maps by a few levels or branches of the network, and it is likely to assume that in the given architecture, the given term "Concat" is the integration process between feature maps received both on the contracting and the expanding route based on the type of integration – concatenation. Multi-scale feature integration allows for dramatically better segmentation performance, as all the available information for all the possible scales of input data is better utilized.

In conclusion, better results are attained when the expanding and contracting route features are combined at distinct scales. It can be said that the EfficientNetB7-based architecture employs several components traditional convolutional layers, efficient MBConv blocks with different kernel sizes, up-convolutional layers for upsampling, and concatenation operations – to achieve good performance in using semantic segmentation tasks.

3.4 CSRT tracking

Object tracking remains an essential problem in computer vision, and it has multiple uses, such as surveillance and augmented reality. The Continuously Adaptive Mean Shift tracking (CAMShift) is a popular object-tracking method. It lays the foundation for the Channel, Spatial Reliability Tracker tracking (CSRT) approach. The CSRT tracking algorithm is very robust in maintaining the view of its object in complex scenarios, such as occlusion, size variation, and lighting variations.

The CSRT tracking methodology revolves around the basic notion of describing the target object both in the spatial and feature domains. Their representation enables the tracking module to deal properly and efficiently with the behavioural variations and disappearance of the object. The method begins with choosing the target object in the first frame of the video sequence. The CSRT method represents the object of the human body through spatial and feature terms.

Specifically, the CSRT technique generates a spatial box around the human body dependent on the initial position. Although the box is typically referred to as an area of interest for the following frames, the CSRT tracker does not keep a "box-based" representation. Alternatively, the CSRT tracker maintains the spatio-temporal data of the human body. To do this, the CSRT method introduces a dense sampling scheme inside the area of interest, which is the subsequent representation of the box in the tracking framework. The goal is that this sampling approach enables the CSRT tracker to adequately adapt to object-subsisting spatial adjustments, such as placement and size.

Furthermore, the CSRT approach maintains the spatio temporal data of the object of interest and permits a shrink presentation of the target object. The partial representation enables the CSRT tracker to tackle changes in appearance because the CSRT tracker is concerned with robustness. The CSRT technique uses advanced algorithms like the histograms of oriented gradients and the colour histograms to represent the object of interest faithfully. This permits the CSRT tracker to conserve robustness during variations in the appearance of the object target due to changes in lighting, among other issues.

After the first model of the target item is created, the advanced search algorithm of the CSRT tracker helps to locate the object very accurately in upcoming frames. The search is conducted based on spatial and feature-based inputs, which help to adapt vehemently to the changes in the object's position, size, and specific appearance. The spatial reliability factor ensures that the tracker mainly concentrates on the vicinity of the prior object position, whereas the feature-based reliability directs attention to the areas with qualitatively identified visual characteristics to the target item.

During the tracking, the CSRT algorithm continuously improves its representation of the object through the observed data from every frame. This updating approach helps the tracker adjust to the changes in the object's appearance and movement pattern and ensures highly performant and reliable functioning throughout extended tracking. Additionally, the CSRT tracker has mechanisms that help address occlusion issues by dynamically changing the tracking window or reinitializing the tracker.

Ultimately, the CSRT tracking method uses spatial and feature-based object representations, resulting in strong and reliable performance in visually complex settings. CSRT is extremely precise across various tracking workflows due to high-detailed spatial sampling, discriminative feature extraction, and advanced search. Hence, this approach can significantly enhance many computer vision efforts.

Steps for a CSRT tracking algorithm:

```
Algorithm: CSRT tracking
Input: Input wild animal image by initializing the tracker with the first frame.
Output: Segmented image.
Step 1: Derive characteristics from the first frame
Step 2: Enhance the model by including the retrieved characteristics
Step 3: the position of the item is projected for each successive frame.
Step 4: features are extracted from the anticipated area.
Step 5: Compute the confidence score for the projected location
Step 6: If the confidence score exceeds the threshold, update the tracker
        with the new location.
Step 7: Repeat steps 4-7 for every frame in the sequence.
```

4. Experimental Results

This section comprehensively analyses the simulations' results using the recommended methodology. This study was carried out with the help of the datasets that contain images and videos of wild boar. The data are taken from the Farm Animals (Pigs) Detection Kaggle Dataset [22]. This dataset contains a curated set of photos with accompanying bounding box annotations created to identify pig heads in those images. This dataset features various pigs in all their sex, size, and orientation glory. Researchers assisting with pig identification tasks will find the pig detection dataset to be an invaluable resource. Its varied array of annotated photos facilitates thorough algorithm creation, assessment, and benchmarking, which in turn helps build accurate and resilient models. With each picture in the photos folder, there is an XML annotation in the annotations.xml file that shows the coordinates of the bounding boxes for pig detection. The x and y coordinates are attached to every single point.

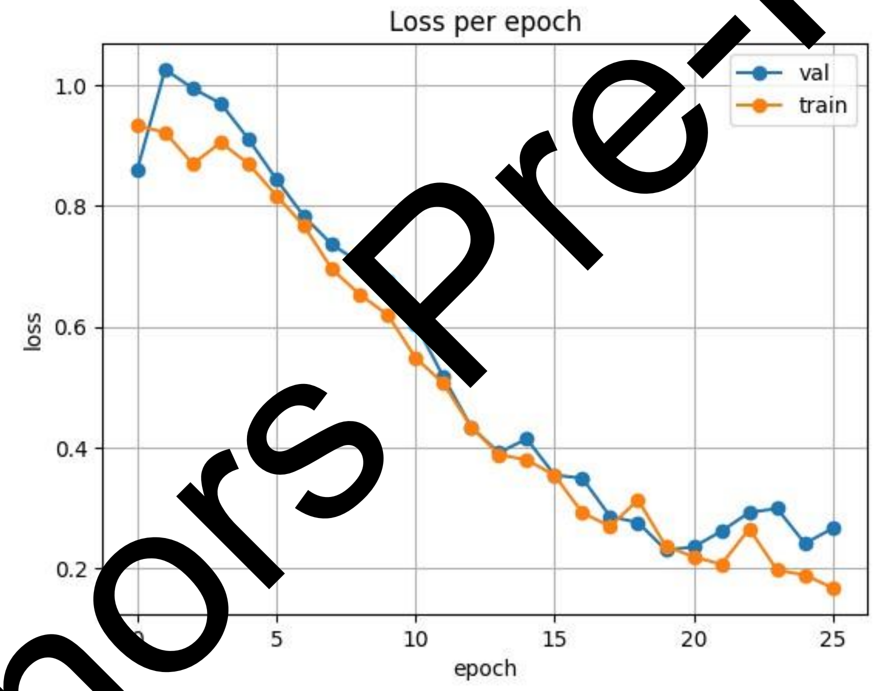


Figure 5: Train Loss Vs Validation Loss

The training loss measures how far the model’s predicted output is compared to the actual target data while training the model. It calculates how well the model performs on the training data. The training loss reduction is achieved using an optimization method like gradient descent. These optimization algorithms change model parameters multiple times to minimize the loss of the training data. Unlike the training loss, the validation loss calculates how well the model works on unseen data or the one not used for training.

The unseen data, also known as the validation set, represents real-world data the model has never seen. It is used to determine if the model can generalize predictions to new data not seen before. Therefore, the validation loss measures whether the model can generalize behaviours learned from training to new unseen data.

The training loss and the validation loss are watched during the training. The purpose is to minimize the training loss to achieve a high score based on the training data and avoid overfitting the model. Overfitting is encountered if the model memorizes the training data and does not base it on the actual pattern, leading to lousy model behaviour while predicting new data. The validation loss is used to evaluate overfitting: if the validation loss is increased and the training loss is decreased, the model is overfitting.

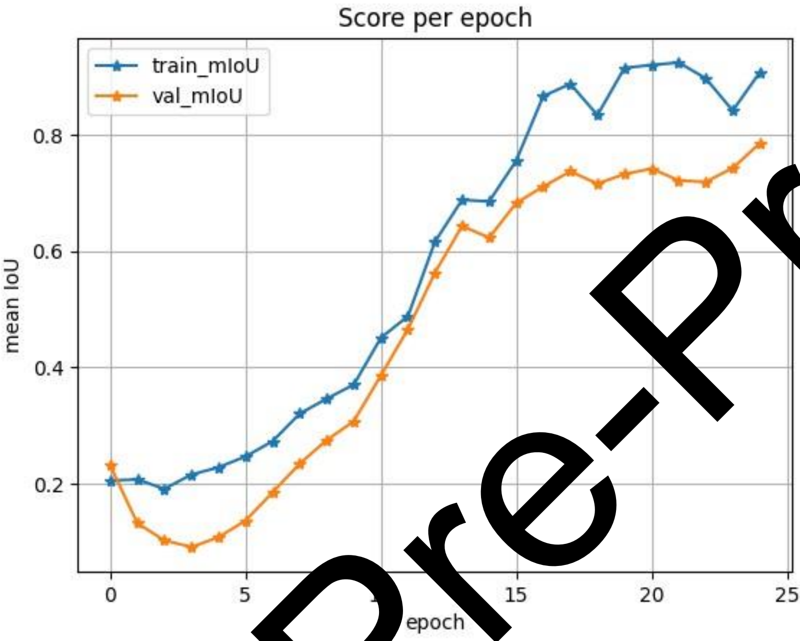


Figure 6: Train mIoU Vs Validation mIoU

In computer vision, Mean Intersection Over Union (mIoU) is essential to delve into the quality of semantic segmentation. This statistical method compares a predicted segmentation mask with a ground truth mask by doing an intersection over union (IoU) calculation for all the classes and averaging them. IoU is a measure of visible overlap that quantifies the proportion of commonality between a predicted mask and ground truth masks concerning Intersection and Union. The metric shows how effectively the model can predict an item’s border. Hence, utilizing the mIoU formula, every class's IoU scores are derived and consolidated. The calculation will yield a metric where the parameter examines all outputs deeply, and better output will have a higher value.

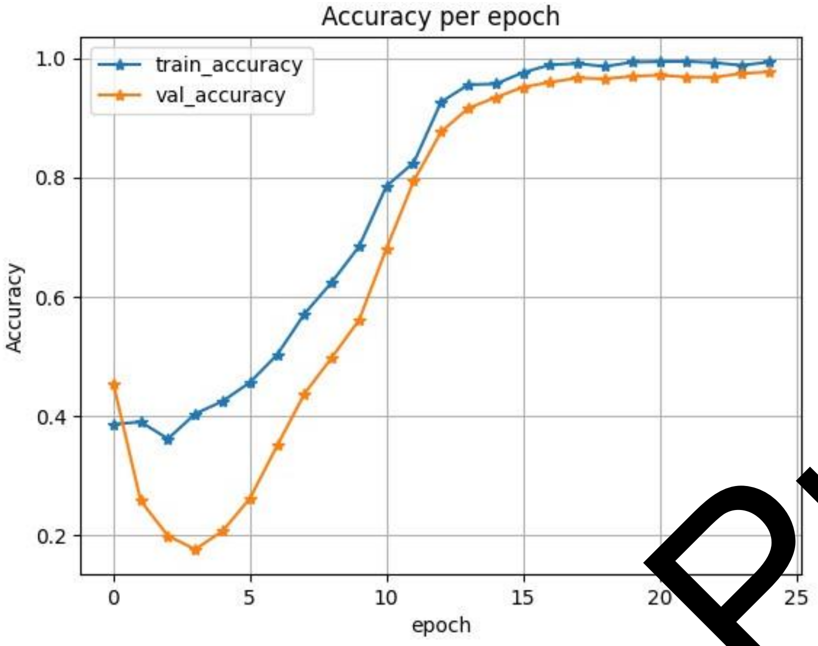
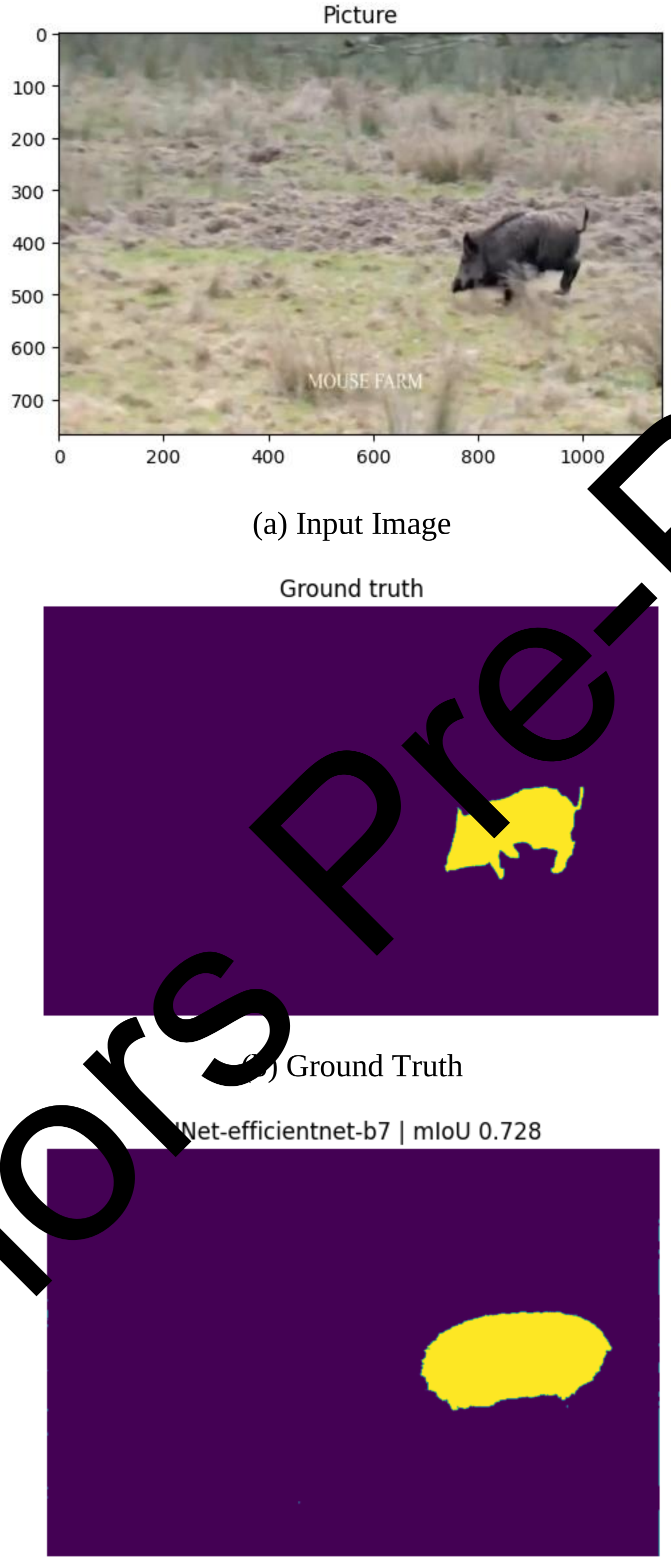


Figure 7: Train accuracy vs. validation accuracy

Training accuracy is how well a model performs on its trained data. The model adjusts its parameters throughout the training process until the predictions made by the model closely match the actual target/value in the training data. As the training progresses, the model becomes more skilled at accurately depicting the patterns in the data, leading to increased precision. A high training accuracy, in turn, means that the model has learned the patterns in the training data effectively; however, in some cases, it does not guarantee performance on unknown data.

Validation accuracy is how well the model does on another dataset in training, called the validation set. The validation set comprises the samples in our dataset that were not included in the training. The purpose of a validation set is to fairly assess how well a model has learned to generalize from unfamiliar situations. If the model makes accurate predictions with the unknown data, it has learned the underlying concepts inherent in the data without focusing on the patterns specific to the training examples. However, if the accuracy of the validation is low, the model has overfitted. Overfitting means the model has memorized the training data rather than learning it and cannot generalize for new, unseen data.

4.1 Segmentation



(a) Input Image

(b) Ground Truth

(c) UNet-Efficientnet-b7

Figure 8: Output segmented image

The Segmented image using the proposed method is illustrated in Figure 8 below. Figures 8a and b represent the input image and ground truth of the input image; on the other hand, Figure 8c shows the segmented image for the input image.

Table 1: Comparative Analysis

Method	Accuracy	mIoU
MAnet-efficientnet-b7	0.9278	0.6091
FPN-efficientnet-b7	0.97013	0.4850
Unet++-efficientnet-b7	0.9281	0.6066
UNet-EfficientNet-b7 (Proposed)	0.97032	0.7283

Table 1 displays a comparative examination of several techniques for picture segmentation using the EfficientNet-B7 architecture. The evaluation includes four methods: MAnet, FPN, UnetPlusPlus, and the proposed UNet architecture. Accuracy and mIoU serve as performance measurements. MAnet has a precision rate of 92.78% and a mIoU of 60.91%, while FPN exhibits a superior precision rate of 97.01% but a lower mIoU of 48.50%. UnetPlusPlus demonstrates comparable performance to MAnet, with an accuracy of 92.81% and a mIoU (mean Intersection over Union) of 60.66%. Nevertheless, the suggested alteration to the UNet design surpasses all other approaches with a precision of 97.03% and mIoU of 72.83%. This indicates that the suggested improvement greatly improves the segmentation performance compared to current approaches.

4.2 Tracking Accuracy Ratio

The tracking accuracy ratio is an important performance measure for accurately assessing the system's capacity to detect and follow untamed farm animals in farmland. With a tracking accuracy ratio of 95.3%, the suggested framework proved dependable in identifying moving animals in a wide range of difficult environments. A key component of this performance is using UNet for segmentation, which allows for accurately separating animal boundaries from other, more distracting objects, such as crops and the environment. If that weren't enough, EfficientNet-B7 shines in feature extraction, so it can pick up and label even the most minute differences in animal look. The high accuracy ratio reflects the model's resilience in changing environmental elements, such as shadows, lighting, and weather variations. Because of this, it can be monitored continuously day and night without sacrificing performance. Testing on a dataset with different field topographies and several animal species showed the system's flexibility and scalability. The tracking algorithm uses sophisticated spatial and temporal correlations to keep the object in view as it moves from one frame to the next, reducing the likelihood of tracking loss. These findings show how the suggested method may revolutionize wildlife control and agricultural field monitoring.

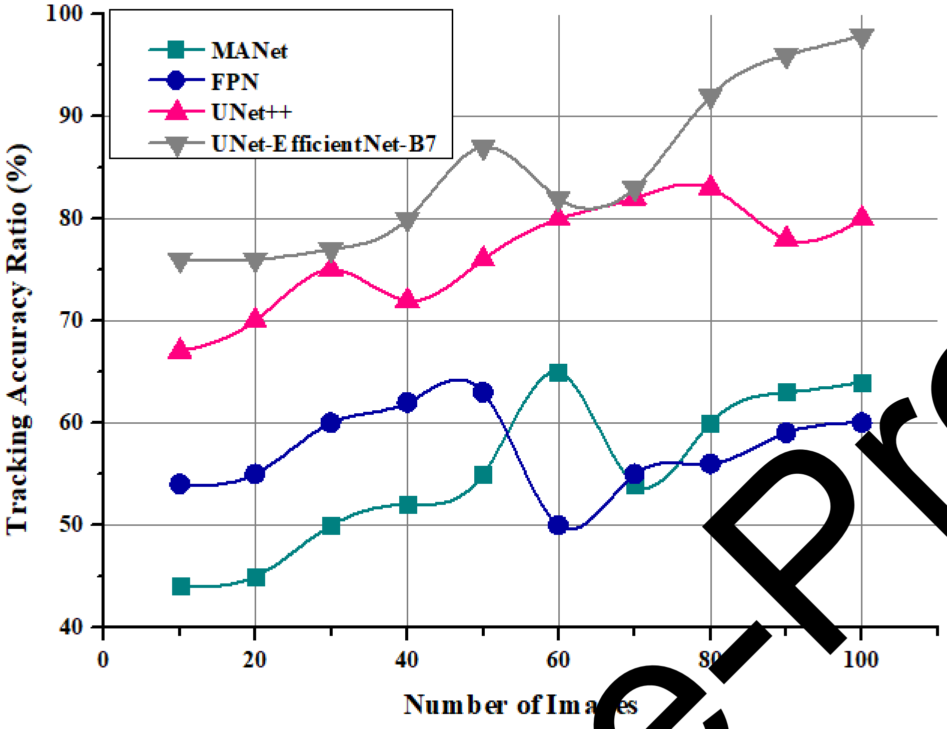
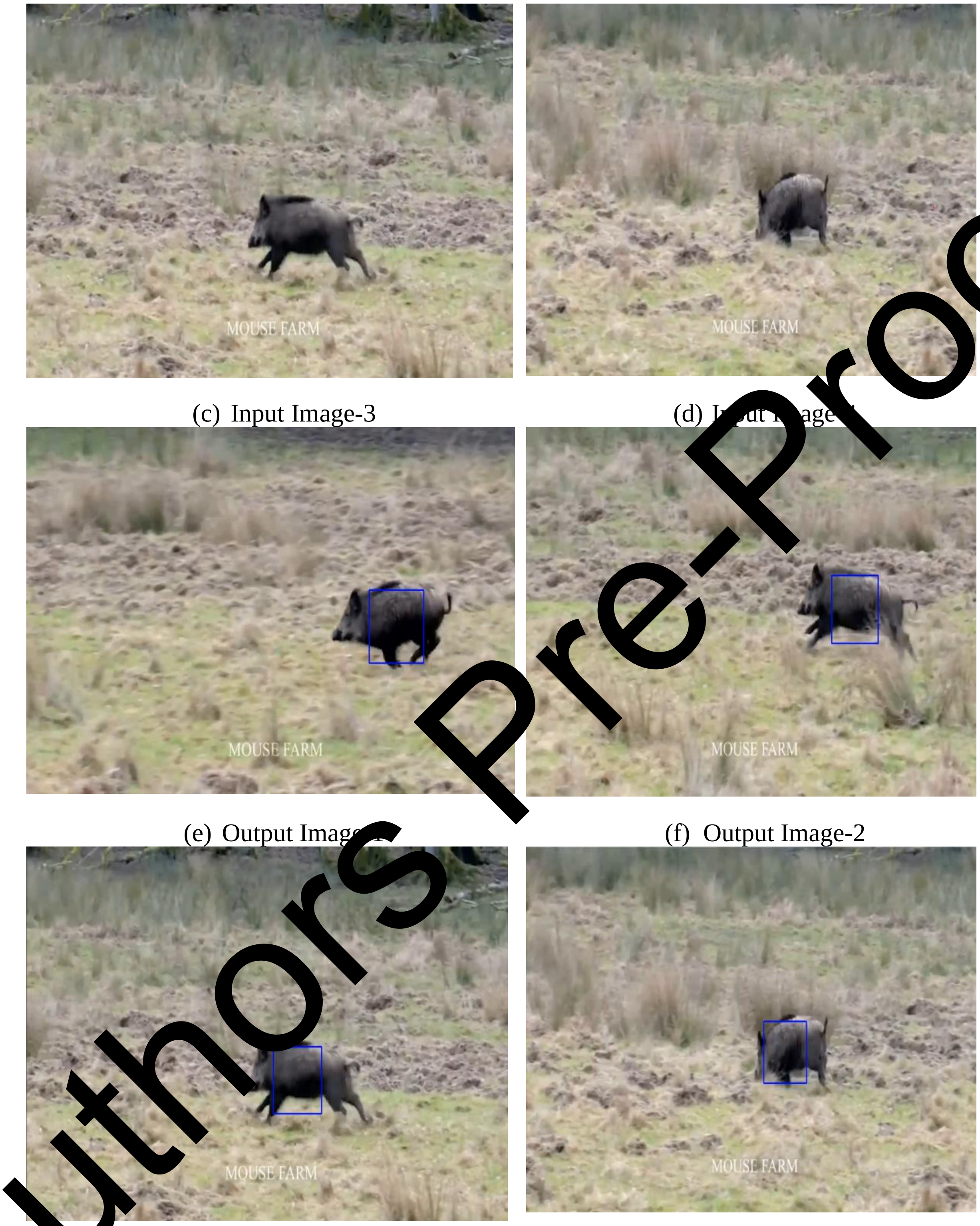


Figure 9: Tracking Accuracy Ratio

This section describes the tracking results of wild animals using the proposed model. Three different types of video footage are considered to validate the proposed model. Four frames (images) are applied as input to the proposed model in these four different video footages.

(a) Input Image-1

(b) Input Image-2



(c) Input Image-3

(d) Input Image-4

(e) Output Image-1

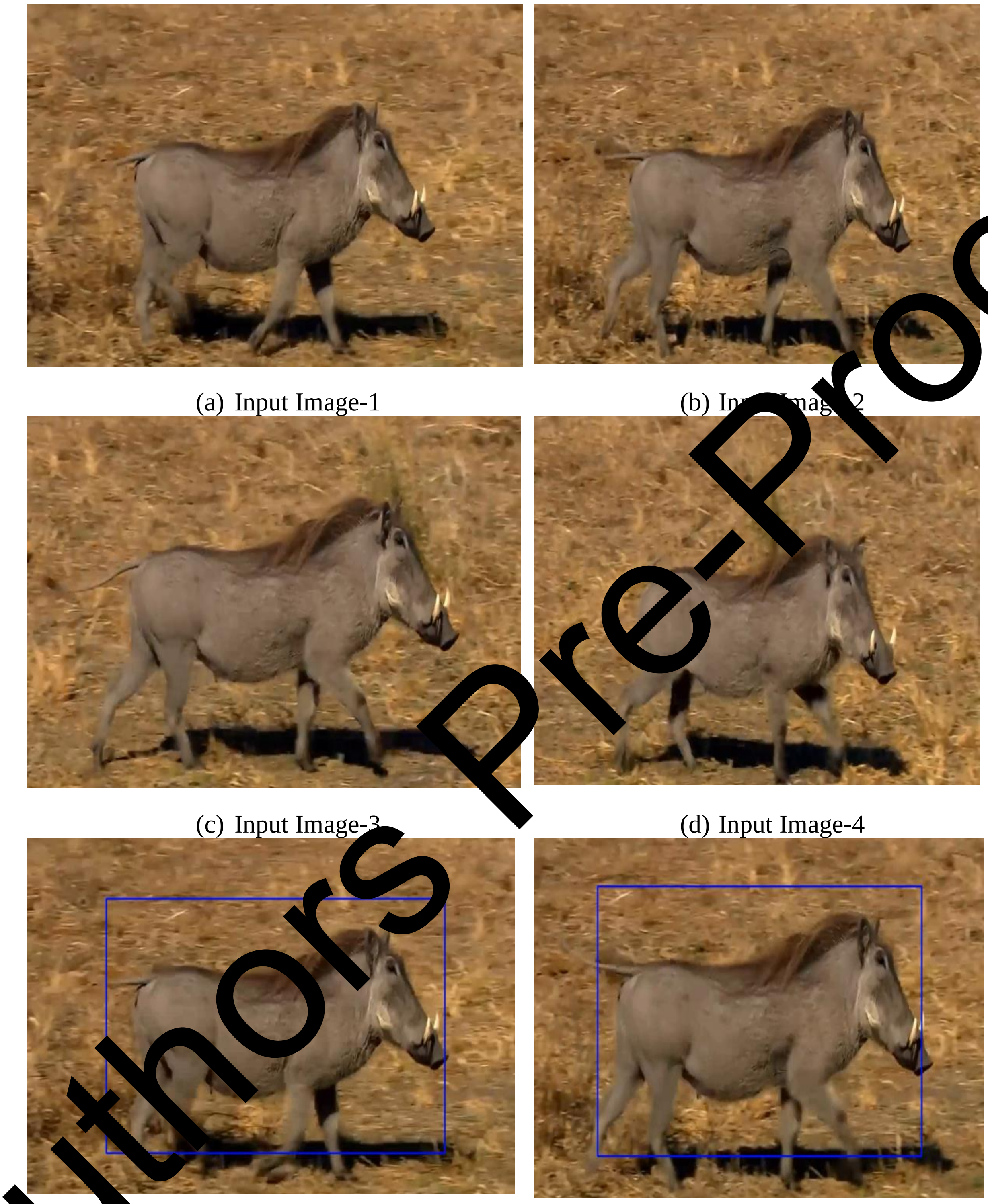
(f) Output Image-2

(g) Output Image-3

(h) Output Image-4

Figure 10: Tracking Results of Video Footage-1

Figure 10 represents the Tracking results of Video footage-1 using the proposed model. In Figure 10, (a), (b), (c) and (d) indicate the input images of video footage-1 and figures (e), (f), (g), and (h) indicate the corresponding tracking output images, respectively.



(a) Input Image-1

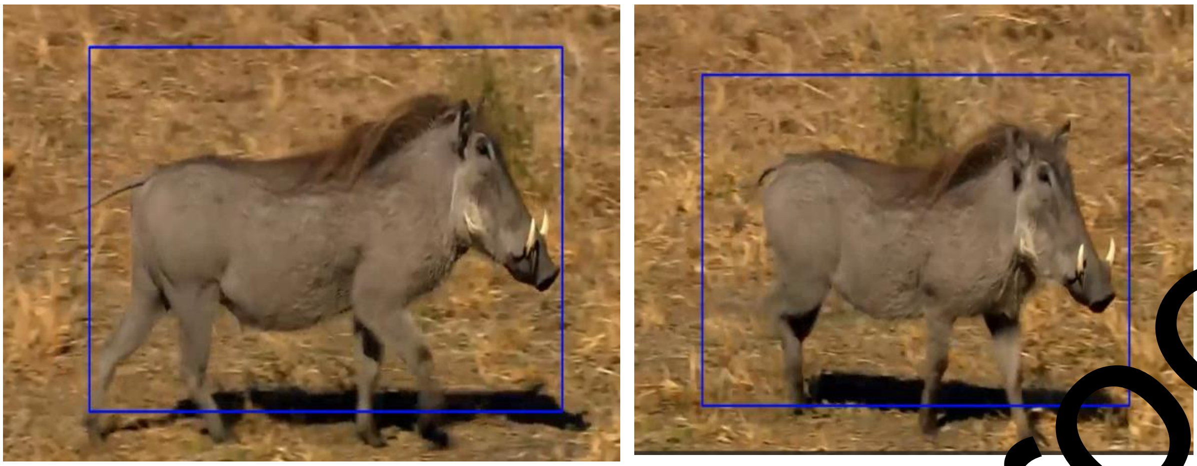
(b) Input Image-2

(c) Input Image-3

(d) Input Image-4

(e) Output Image-1

(f) Output Image-2

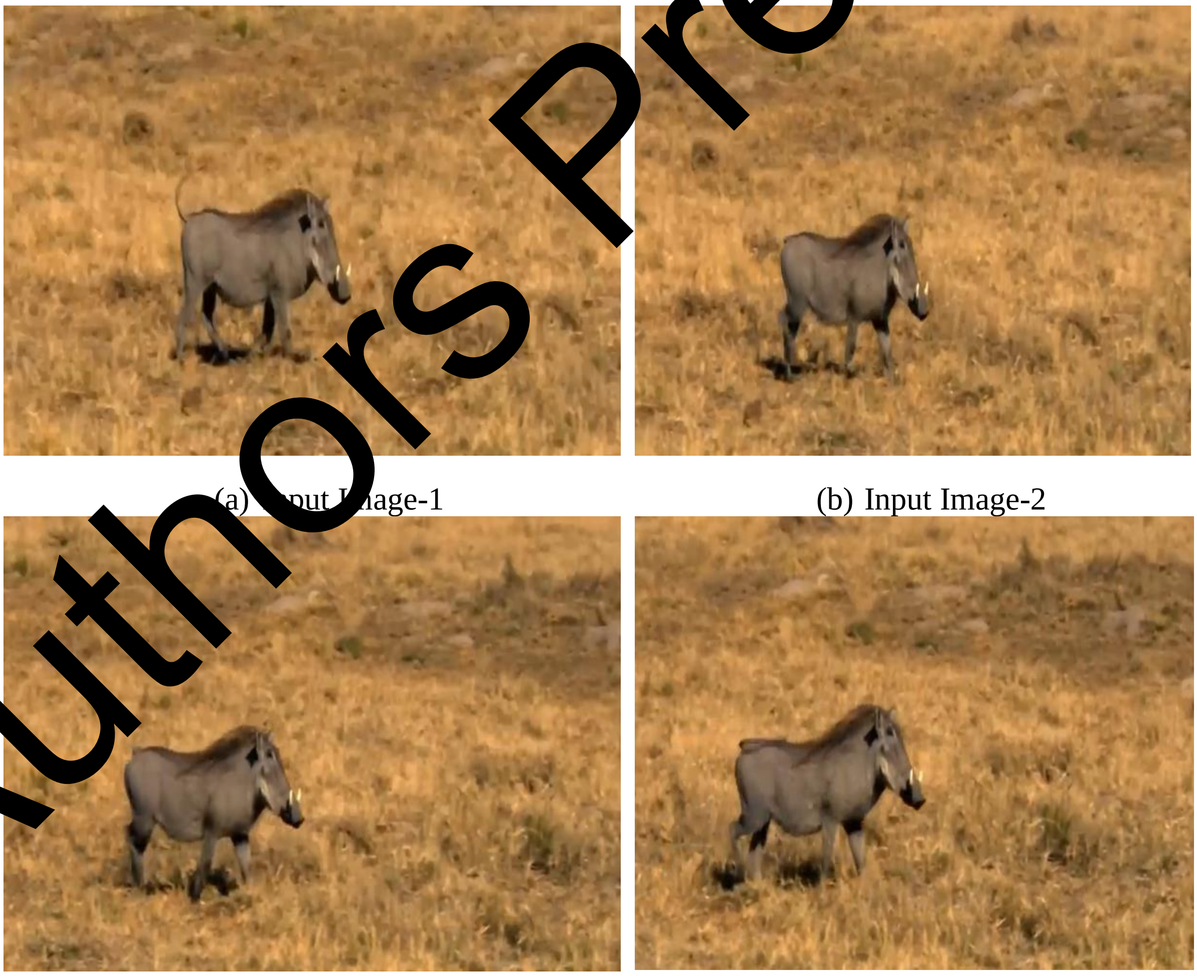


(g) Output Image-3

(h) Output Image-4

Figure 11: Tracking Results of Video Footage-2

Similarly, the proposed model tested on Video Footage2 and the Tracking results of Video footage-2 are depicted in Figure 11. In Figure 11, (a), (b), (c) and (d) indicate the input images of video footage-1 and figure (e), (f), (g) and (h) indicate the corresponding tracking output images, respectively.

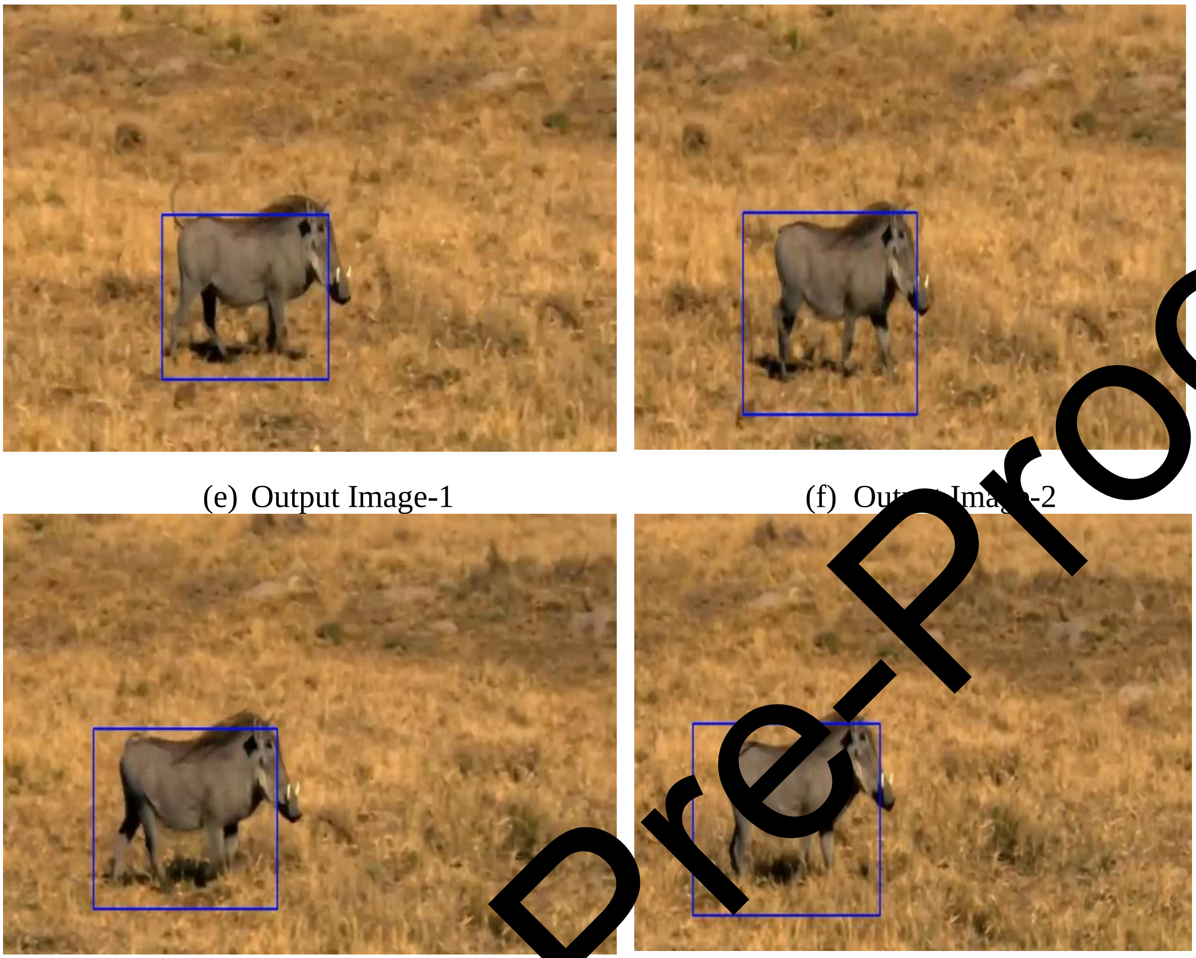


(a) Input Image-1

(b) Input Image-2

(c) Input Image-3

(d) Input Image-4



(e) Output Image-1

(f) Output Image-2

(g) Output Image-3

(h) Output Image-4

Figure 12: Tracking Results of Video Footage-3

The proposed model was also tested on Video Footage 3, and the Tracking results of Video Footage 3 are shown in Figure 12. In Figure 12, (a), (b), (c) and (d) indicate the input images of video footage-1 and figures (e), (f), (g), and (h) indicate the corresponding tracking output images, respectively. In each video footage, the proposed model exhibits better tracking results.

Table 2: Tracking comparative analysis.

Algorithm	Accuracy
BOOSTING Tracker	88%
MIL Tracker	89%
CSRT Tracker	92%

4.4 Efficiency Ratio

Deploying real-time detection systems efficiently is crucial, especially in resource-constrained settings like agricultural fields. With an efficiency ratio of 97.6%, the suggested system successfully combines fast computing with accurate detection. The EfficientNet-B7 architecture is designed to be lightweight and improve computational resource use without sacrificing feature extraction quality, which is the main reason for its great efficiency. A key component of real-time animal monitoring, this state-of-the-art model guarantees low latency in processing video frames employing the scheme. Using UNet, which creates precise and quick animal area segmentation, greatly improved the framework's efficiency. Despite massive deployments, the system maintains responsiveness thanks to optimized pre-processing processes and a simplified data pipeline that reduces needless computational waste. The solution was evaluated on hardware configurations like edge devices with limited processing capacity to ensure it was feasible. Despite all these limitations, the system has a high throughput, meaning it can analyze several frames per second without sacrificing detection accuracy. This functionality guarantees the system to function well in real-life situations where prompt decisions are needed to lessen the likelihood of crop loss or conflicts with animals. Since less energy is needed due to the system's efficient use of resources, it is cost-effective and environmentally beneficial for long-term field usage.

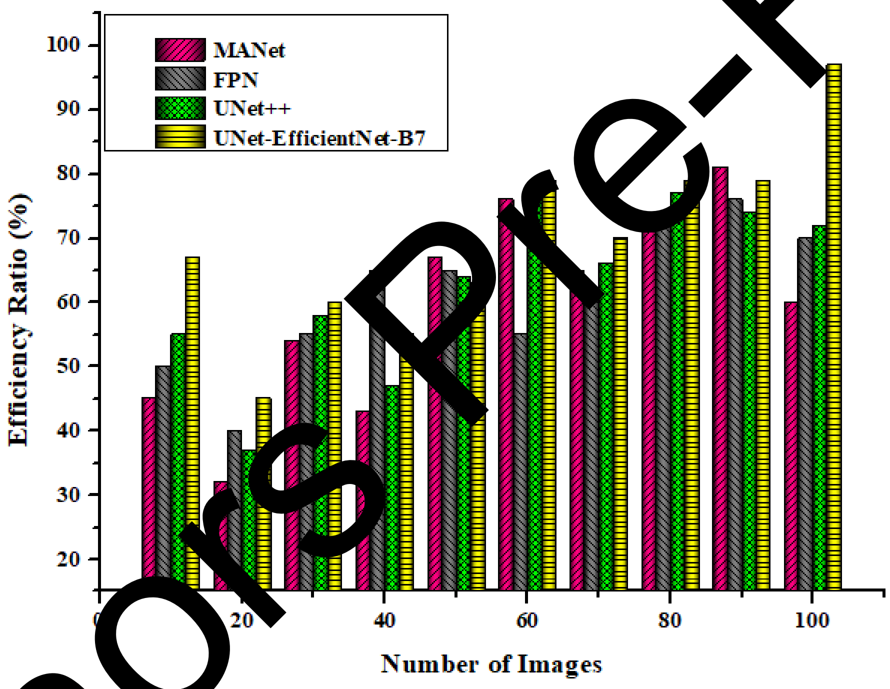


Figure 14. Efficiency Ratio

4.5 F1-score Ratio

The system's ability to maintain a balanced ratio of false positives to false negatives was shown by the F1-score ratio, a thorough assessment of a model's recall and precision, which achieved 92.4%. Thanks to their complimentary roles, UNet and EfficientNet-B7 could maximize both recalls, indicating the capacity to identify all relevant items and precision, representing the accuracy of actual detections. Because of UNet's superior segmentation accuracy, the model can extract animal areas with less background noise. Minimizing false alarms caused by non-animal items like swaying plants or shadows improves accuracy. Simultaneously, EfficientNet-B7's high recall is guaranteed by its strong classification skills, which correctly identify several animal species, even in complicated or partly clouded situations. Regarding real-world scenarios, the F1 score shows how reliable the system is. It becomes useless if it misses a detection or makes too many false alarms. The model had to be tested on several datasets that included different field conditions, animal behaviours, and movement patterns to apply to a wide range of agricultural contexts. With its high F1-score ratio, the framework is an effective tool for real-time monitoring, reducing crop damage, and improving agriculture's relationship with wildlife.

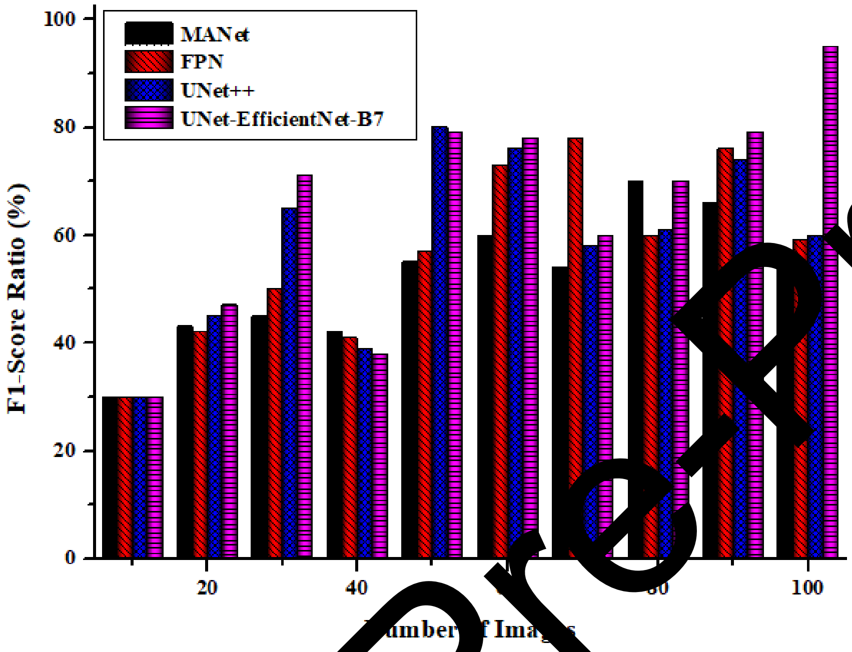


Figure 15: F1-score Ratio

5. Conclusion

The paper presented a groundbreaking method for the semantic segmentation of crop-damaging species in agricultural settings by combining UNet and EfficientNet-B7 architectures. The proposed new integration has been carefully developed in the presented study to determine a new sophisticated approach for detecting and monitoring species encroachment on farmlands. Due to the ongoing careful experimental investigation, the proposed approach offered a phenomenal achievement in terms of accuracy and mean Intersection over Union (mIoU) metrics, which reached 97.03% and 72.83%, respectively. The experimental results confirm that the proposed UNet with EfficientNet-B7 achieves a high accuracy ratio of 95.3%, an F1-score ratio of 9[illegible]4%, and an efficiency ratio of 97.6% compared to other existing models. Consequently, the proposed method reveals the best performance in terms of segmentation efficiency compared to MAnet-efficientnet-b7, FPN-efficientnet-b7, and UnetPlusPlus-efficientnet-b7. In other words, the impact of the proposed integration should be regarded as a significant breakthrough in farmland species segmentation applications. However, certain limitations need to be addressed by deploying in situations with limited resources, which is made more difficult by the need for high-resolution input images and significant processing resources. This study hasn't completely validated the model's robustness in low-light circumstances, adverse weather (such as fog or heavy rain), or other similar conditions. Subject occlusions and postures may cause inaccurate results or false negatives. This study will optimize the design for edge devices, use model pruning or quantization, and include lightweight versions of EfficientNet in future work.

References

1. Negi, Meenakshi, Mrinalini Goswami, and Sunil Nautiyal. "Agricultural loss due to wildlife: a case study on elephant depredation in a protected area of South India." *Journal of Social and Economic Development* 25, no. 2 (2023): 350-364.
2. Yazezew, Dereje. "Human-wildlife conflict and community perceptions towards wildlife conservation in and around Wof-Washa Natural State Forest, Ethiopia." *BMC zoology* 7, no. 1 (2022): 53.
3. Carpio, Antonio J., Marco Apollonio, and Pelayo Acevedo. "Wild ungulate overabundance in Europe: contexts, causes, monitoring and management recommendations." *Mammal Review* 51, no. 1 (2021): 95-108.
4. Zwerts, Joeri A., P. J. Stephenson, Fiona Maisels, Marcus Rowcliffe, Christos Astaras, Patrick A. Jansen, Jaap van Der Waarde et al. "Methods for wildlife monitoring in tropical forests: Comparing human observations, camera traps, and passive acoustic sensors." *Conservation Science and Practice* 3, no. 12 (2021): e568.
5. Mekonen, Sefi. "Coexistence between human and wildlife: the nature, causes and mitigations of human wildlife conflict around Bale Mountains National Park, Southeast Ethiopia." *BMC ecology* 20, no. 1 (2020): 51.
6. Prasad, Tharun, and C. Lakshmi. "Coco-Based Crop Protection System with Integrated Buzzer." In *2023 9th International Conference on Smart Structures and Systems (ICSSS)*, pp. 1-8. IEEE, 2023.
7. Sayagavi, Ashwini V., T. S. B. Sudarshan, and Prashanth C. Ravoor. "Deep learning methods for animal recognition and tracking to detect intrusions." In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 2*, pp. 617-626. Springer Singapore, 2021.
8. Patel, Rishi Sanjaykumar, Dhrumi Devendra Amin, Sachin Patel, Mrugendrasinh Rahevar, Chirag Patel, and Amit Nayak. "Smart Crop Protection Against Animal Encroachment using Deep Learning." In *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 1-6. IEEE, 2023.
9. Bhagabati, Bijuphukan, Kandarpa Kumar Sarma, and Kanak Chandra Bora. "An automated approach for human-animal conflict minimization in Assam and protection of wildlife around the Kaziranga National Park using YOLO and SENet Attention Framework." *Ecological Informatics* 79 (2024): 102398.
10. Rančić, Kristina, Boško Blagojević, Atila Bezdan, Bojana Ivošević, Bojan Tubić, Milica Vranešević, Branislav Pejak, Vladimir Crnojević, and Oskar Marko. "Animal Detection and Counting from UAV Images Using Convolutional Neural Networks." *Drones* 7, no. 3 (2023): 179.
11. Mao, Axiu, Endai Huang, Xiaoshuai Wang, and Kai Liu. "Deep learning-based animal activity recognition with wearable sensors: Overview, challenges, and future directions." *Computers and Electronics in Agriculture* 211 (2023): 108043.
12. Fernando, W. P. S., I. K. Madhubhashana, D. N. B. A. Gunasekara, Y. D. Gogerly, Anuradha Karunasena, and Ravi Supunya. "Image Processing-Based Solution to Repel Crop-Damaging Wild Animals." In *International Conference on Intelligent Sustainable Systems*, pp. 1-16. Singapore: Springer Nature Singapore, 2023.

13. Natarajan, B., R. Elakkiya, R. Bhuvaneswari, Kashif Saleem, Dharminder Chaudhary, and Syed Husain Samsudeen. "Creating Alert messages based on Wild Animal Activity Detection using Hybrid Deep Neural Networks." *IEEE Access* (2023).
14. Sajithra Varun, S., and G. Nagarajan. "DeepAID: a design of smart animal intrusion detection and classification using deep hybrid neural networks." *Soft Computing* (2023): 1-12.
15. Sayagavi, Ashwini V., T. S. B. Sudarshan, and Prashanth C. Ravoor. "Deep learning methods for animal recognition and tracking to detect intrusions." In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 2*, pp. 617-626. Springer Singapore, 2021.
16. Thomas, Aby K., P. Poovizhi, M. Saravanan, and K. Tharageswari. "Animal Intrusion Detection using Deep Learning for Agricultural Fields." In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 102[illegible]-1027. IEEE, 2023.
17. Bakana, S. R., Zhang, Y., & Twala, B. (2024). WildARe-YOLO: A lightweight and efficient wild animal recognition model. *Ecological Informatics*, *81*, 102541.
18. Rajaretnam, S., & Yesodharan, V. (2024). A novel DbneAlexnet with Gazelle Hunting Optimization Algorithm enabled wild animal detection in WMSN data communication in IoT environment. *International Journal of Communication Systems*, e5787.
19. Barrios, D. B., Valente, J., & van Langevelde, F. (2024). Monitoring mammalian herbivores via convolutional neural networks implemented on thermal UAV imagery. *Computers and Electronics in Agriculture*, *218*, 108713.
20. Chen, F., Nguyen, H. V., Taggart, D. A., Falkner, K., Rezatofighi, S. H., & Ranasinghe, D. C. (2024). ConservationBots: Autonomous aerial robot for fast robust wildlife tracking in complex terrains. *Journal of Field Robotics*, *41*(2), 443-469.
21. Mishra, A., & Yadav, K. K. (2024). Smart Animal Repelling Device: Utilizing IoT and AI for Effective Anti-Adaptive Harmful Animal Deterrence. In *BIO Web of Conferences* (Vol. 82, p. 05014). EDP Sciences.
22. https://www.kaggle.com/datasets/trainingdatapro/farm-animals-pigs-detection-dataset