International e-Conference on New Horizons And Multidisciplinary Applications In Science And Technology
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
https://doi.org/10.32628/IJSRST2196174

Major Nocturnal Pest Classification Model using Faster RCNN Architecture of Deep Learning

Deven J. Patel, Nirav Bhatt
*[1]Research Scholar, RK University, Rajkot and Assistant Professor, Junagadh Agricultural University, Junagadh, Gujarat, India
[2]Professor, RK University, Rajkot, Gujarat, India

ABSTRACT

Agriculture research improves the quality and quantity of crops, but pests degrade them. Pesticides are used to prevent these pests from reproducing. However, excessive pesticide use is extremely detrimental to both production and the environment. As a result, initial pest detection is required. We analyzed the most frequently used methodologies in order to determine the most appropriate technique for the first diagnosis and early detection of significant nocturnal flying pests such as White Grub, Helicoverpa, and Spodoptera. We identified and analyzed three frequently used deep learning meta-architectures (Faster R-CNN, SSD Inception, and SSD Mobilenet) for object detection using a small Pest dataset. The faster RCNN meta-architecture outperforms other meta-architectures. To address the issue of class imbalance, we used image augmentation with a Faster RCNN meta-architecture. The proposed work demonstrates how to classify Nocturnal Pests using a Faster RCNN of Deep Learning with a better accuracy performance on a limited dataset and utilization as decision-making tool based on classified results.

Keywords : Convolutional Neural Network, CNN, Deep Learning, Pest Detection, Pest Detection, Faster RCNN

I. INTRODUCTION

India is the world's biggest agricultural producer. The quality of information inputs can help improve rural decision-making capacities and hence provide a higher quality of life. IT can play a significant role in easing the transition process by addressing these difficulties and addressing the rapidly growing digital agriculture in rural India[1]. Cropping systems differ according to available resources and limits on individual farms [2]. Pests and plant diseases have resulted in deterioration in agricultural product quality [3]. Pests are responsible for a significantly greater global potential loss of crop productivity in India [4]. The most popular methods farmers use to detect pests are expert opinion or naked eye observation. This demands an in-depth knowledge of various pest species. Expert counsel is not always affordable or convenient, and it might take a long time to obtain[5]. Gujarat is known for its groundnut and cotton production. Cereal grains are still another type of agriculture. A digital view is used to identify the major nocturnal flying Pest species in these crops, and that is the focus of the proposed research.

Infested groundnut crops are sucking parasites such as the white grub (Phyllophaga). Helicoverpa armigera is a serious pest of peppers, according to the literature. The armyworm Spodoptera, a serious grain pest, is the most common culprit (Spodoptera litura). Since they're nocturnal, they only fly at night. Manual pest detection equipment in fields like light traps and sticky traps. Convolutional neural networks (CNNs) are a highly successful data processing technique for huge datasets. It delivered state-of-the-art results in computer vision [6], data mining [7], and machine learning [8]. CNN is a subset of the Deep Learning (DL) technique [9]. CNN and deep learning in general can be used to tackle a wide variety of agricultural prediction and classification issues, not just for computer vision but also for data analysis [10]. These architectures can be utilized as rich feature extractors for performing complex tasks such as image classification, object identification, and picture segmentation. This is employed in order to identify objects and tell them apart from the surrounding environment Whether it's a guy, a hand, a face, or just a bunch of things standing [6]. This study used a Convolutional Neural Network Architecture to detect and classify nocturnal flying Pests. Classification of major nocturnal Pests is possible using the integrated knowledge base in the proposed effort.

II. RELATED WORK

We have performed the execution of three ways of literature studies: (1) The study of practical knowledge; (2) The study of integrated similar knowledge; and (3) the review of previous knowledge.

The study of practical knowledge

A team of researchers have developed an object detection system with high accuracy and low false-positive that can detect objects such as cars, people and animals. Detection Results are presented in the paper of a trainable object detector system for static images [11]. Graph-based image segmentation identifies objects. This is beneficial when the background has several similar things. Their approach might find a search object in a jumbled background. The technique misses multi-part objects like the human body [6]. SURF is a rapid, rotation-invariant scale interest point detector/descriptor. Integral pictures boost speed by reducing the number of operations necessary for simple box convolutions. The Hessian approximation outperformed typical interest point detectors [12].Multi-scale A CNN is a multi-scale deep convolutional neural system. The detection occurs at network levels whose receptive fields match different object scales. This allows all object sizes to be detected by sending one image across the network. It makes a quick detector. An alternative to up sampling is CNN feature approximation. It reduces processing and memory requirements [13].Multiscale characteristics are crucial for enhancing vehicle detection performance using satellite pictures and hybrid CNN. The current state-of-the-art Deep CNN approach cannot extract multiscale features. The authors suggest hybrid Deep CNNs. Each convolutional or max-pooling layer map has many blocks with varied receptive fields or max-pooling fields [14].Person detection using three Convolutional Neural Networks. People can be detected at various scales. This is the first deep learning pedestrian detection model. Raw pixel data is used to train networks. Because numerous Convolutional Neuro Networks are used at various scales, the learnt features are scale-dependent. This improves performance [15].

The study of integrated similar knowledge

E-noses have various benefits, including excellent sensitivity and real-time processing. They're also portable and easy to use. Thus, E-nose technology can be used to detect plant pests. E-noses can reliably anticipate plant diseases and perform well as sensors. This has been confirmed in the field and in the lab. There are still issues with sensor selectivity, environmental interference, and detecting plant diseases in open fields. These concerns need more research and improvement [16]. Adult white-fly larvae can be seen on rose leaves. They used enhanced cognitive vision and scanner image acquisition to recognize biological things in complicated backdrops. Extensible knowledge-based systems that can image analyze and recognize natural things are included, as is machine intelligence. A combination of complementary approaches and disciplines resulted in an automated, reliable, and versatile system. The prototype system could reliably detect whiteflies. It's easy to use and performs as well as manual methods. Three-fold method can increase in whitefly detection and leaf surface coverage [17]. PestNet is a large-scale automated multi-class pest detection network. Their new Channel-Spatial Attention module could help PestNet automatically extract higher quality features. In pest classification and box regression, they chose Position-Sensitive Scope Map over Fully Connected. PestNet includes Regions of Interest as background information to improve detection performance [18]. A visual spectrograph can detect yellow rust in winter wheat. Normalizing irradiating spectrometry spectra is required. Because of canopy architecture and lighting variations, intensity levelling can greatly reduce spectral variability. One-plant spatial averaging windows can increase discrimination [19]. Experimenting with transfer learning in millet crops they employ the popular deep learning frameworks Keras and TensorFlow. The test employed the 80/20 setup to improve performance. This technique finds the convergence value with the least validation loss, avoiding overfitting. This is done by defining the maximum number of epochs without improving the validation set before stopping training. They show the power of tiny data to transmit illness categorization learning. The team found mildew in crop millet using pre-trained ImageNet feature extraction [20].

The review of previous knowledge

A moving camera was used to examine several scenarios for detecting moving objects. HOG3D, SIFT, and Extended SURF were the most effective key-point detectors and feature identifiers. Long-term tracking of moving objects relies on learning procedures and motion prediction models. Deep neural networks were also used in visual tracking because of the availability of large annotated datasets [21]. R-CNN has a number of tweaking options. Generic object detection pipelines serve as the foundation for all subsequent systems. Recognition of pedestrians, finding of prominent objects, and detection of faces are the next three frequent tasks we'll go over briefly. They point to possible future directions that will help us gain a better understanding of the object detection environment [22]. The most effective algorithm is SIFT, whereas the fastest is ORB. SIFT outperforms all other algorithms when the degree of rotation is 90 degrees or more. In noisy images, ORB, SIFT and SIFT all perform nearly identically. For each method, the number and rate were shown together with important spots and execution time to show the matching parameters [23]. As education and information may be significantly improved through the use of ICT, we need to consider bringing in actors from both within and beyond the agriculture system [24]. Using natural language processing and deep learning for computer vision is becoming more and more efficient. Deep learning frameworks have grown in popularity as a result of this. Finding the ideal deep learning framework is a challenge. There were 18 frameworks in total, according to the results of the survey. There was an additional six framework evaluation indicators supplied in the report [25], along with performance comparison data.

Many studies have shown that CNN outperforms alternative computer vision approaches in terms of execution pace and accuracy. Image analysis and computer vision can be utilised to detect pests, as well as data analysis [3]. There have been comparisons made between the accuracy and performance of Convolution Neural Networks and those of other approaches. My decision to use the framework was based on these scholarly works. Python-based deep learning toolbox will allow me to study more about Keras once I've learned the language. Also, TensorFlow and Theano can be interfaced with Keras. Based on literature investigations, I discovered two options: either use deep learning in the research or develop a new deep-learning system from scratch.

III. MATERIALS AND METHODS

A. Convolutional neural network (CNN)

Convolutional neural networks (CNNs) are utilised in computer graphics for object detection and image classification. Neural networks are constructed by connecting many of our "neurons" together in a web like fashion, in order for the output of one neuron to become the input to another.

Introduction:

The convolutional network was inspired by Hubel and Wiesel's [26] research on the cat's visual brain, which revealed certain parts of the visual field seem to excite particular neurons. This bigger notion guided the sparse architecture of convolutional neural networks. The structure of a neural network is comprised of a layer with an input and numerous hidden layers. Each layer can have a direct connection to the layer above it. In a convolutional neural network, the many layers have their states arranged according to a grid-like pattern. Keeping these spatial links between grid cells is necessary to perform the convolution process and subsequent transformation to the next layer [27]. Deep learning teaches computers how people learn best by modelling. Deep learning is a promising new image processing and data analytics approach. Deep learning uses neural networks. The term deep neural networks refers to deep learning models. "Deep" refers to the amount of layers concealed in a neural network. A typical neural network has 2-3 hidden layers. Deeper networks can hold 150.

Meta-architecture

The development of object detection methods based on CNN has opened up a new era in end-to-end methods that can achieve rapid detection and high performance. This topic is challenging because it deals with issues such as object size, lighting levels and perspective, rotation angle, background similarities, and intra-class similarities [28]. Depending on the application, each method differs.

Faster R-CNN

Region Proposal Network is a network that shares full-image convolutional elements with the detection network. This allows for nearly free region proposals. RPN is trained from end-to-end in order to generate high-quality regions proposals. For the very deep VGG-16 model, they combined RPN and Fast R–CNN to create a single network [29]. Faster-RCNN has three neural networks: Feature Network, RPN, and Detection Network.

The Feature Network is a well-known pre-trained image classification network. This network extracts useful features from photos. With this network, you get photographs that look like the original. RPNs are input into two layers: classification and bounding box regression. RPN creates ROIPs (Regions of Interest Pools). These are boxes that might hold any object. This network generates bounding boxes with pixel coordinates. The Detection Network (R-CNN network) receives inputs from both the Feature Network and the RPN. Final classes and bounding boxes are generated [30].

SSD Inception

Single Shot Multi-Box Detector (SSD) is one of the fastest algorithms. It employs a single convolutional neural network to recognize objects in images. SSD is fast, but not as fast as mAP. This work presents Inception SSD, a way to improve SSD algorithm classification accuracy without impacting performance. Inception blocks replace extra layers in SSD. The proposed network can capture more data without becoming more sophisticated. This network uses batch normalization (BN) and residual structure in I-SSD network architecture. To solve the model's lack of expression capabilities, I-SSD adopts a non-maximum suppressing method [31]. SSD's additional layers use a single sort of convolution kernel - the three-dimensional kernel. These additional layers contain feature maps that, via tiny convolution operations, generate the object's position offset and confidence. Inception minimizes the number of feature mappings in each layer of an Inception block while maintaining the total of feature maps in the original layers [31].

SSD Mobilenet V2

MobileNet builds lightweight deep neural networks using a simple design. It makes use of separable convolutions along the depth axis and includes two straightforward global hyperparameters that enable you to trade off accuracy and latency. It demonstrates superior performance and considerable experimentation with resource and precision trade-offs [32]. Google developed the Mobilenet network model to replace VGG16 and simplify the VGG-16 detector's processing cost. This improves the real-time performance of the SSD detection, which operates in real time. It is faster than other object identification networks due to the shift in the backbone model from VGA-16 to Mobilenets V2. The backbone network model is the second-generation Mobilenett V2, which has six distinct feature maps that enable multi-scale item identification [33].

B. TensorFlow and Keras

TensorFlow Distributions are a collection of 56 distributions that give rapid and robust numerical methods for sampling, determining log densities, and performing other statistical operations. The technology has gained widespread adoption both inside and outside of Google. External developers have utilised it to develop probabilistic programming systems and statistical systems.[34]. Keras is a Python-based deep learning API that runs on TensorFlow's machine learning framework. It provides the necessary abstractions and building pieces for rapidly developing and shipping machine learning solutions. Keras may be operated on huge clusters or TPUs, as well as exported for use in the web or on mobile devices[35]. Deep learning based on kernels enables rapid development and runs flawlessly on both CPU and GPU simultaneously. This framework is written in Python, making it easy to debug and maintain.

C. Capturing and Labeling Pest Images

Algorithms for deep learning are capable of establishing relationships, comprehending, making decisions, and assessing their confidence based on training data. Numerous photos are obtained during the early stages of research under stringent lighting conditions that are hard to impose in real-world applications. This is critical to remember when performing non-destructive analysis.With the assistance of the Department of Entomology at Junagadh Agricultural University, an image dataset of nocturnal Pests was obtained using an android phone camera. Object detection begins with the critical task of image labelling. While labelling requires considerable effort, the more dedication you have, the more time you will save. LabelImg[36] is a free, open-source programme for graphically annotating images. The user interface is written in Python and utilizes Graphical User Interfaces (GUI). Along with labelling the entire thing, the label must include some non-object buffer area, most notably surrounding a substantial piece of the object. Place boxes that are realistic representations of the target objects while exposing as much of the target objects as possible. The following images illustrate the suggested taxonomy of bug species for this work's labelling. As illustrated in Figure 1, pest image annotations are stored as XML files using LabelImg. This will be processed in advance of training.

Figure 1 : Image labeling sample using LabelImg tool

The figure 1 shows the annotation interface or software. In this example, the highlighted item is an Pest. The goal is to identify the region of interest and name it according to its class. The boxed portion must be given the suitable class name. As a result, a new directory is formed with all of the photographs and their highlighted annotations in.xml files. The above approaches are used to annotate and name each category depending on the number of Pests displayed in Table 1.

Sr.	Pests	Identical Name	Number of Pests
1	*Phyllophaga*	INS-A	218
2	*Spodoptera litura*	INS-B	201
3	*Helicoverpa armigera*	INS-C	174

Table 1 : Identical name of Pests

D. Comparison of CNN Meta-Architectures

An experiment to assess the accuracy of three well-known deep learning meta-architectures using a small dataset of pest images has been published. The proposed work assessed the performance of each model in a similar TFRecords environment, as well as the generic layer settings for all three CNN Meta-Architectures [37]. TensorBoard is a machine learning platform that delivers metrics and graphics. It tracks experiment parameters like loss and accuracy and visualises the model graph in a two-dimensional space. Loss functions are used to measure the gap between a model's predicted output and the trained model's ground truth. A graph is very beneficial when developing an application for a classifier or recognizer [38]. The frozen graph contains the network's overall model. The current work has enabled the model and TensorFlow layers to be displayed in a TensorBoard. Some argue that visualising a network model is unnecessary because it shows how training works and which blocks are responsible for which functions. As a result, export inference graph will be written. Saving the frozen graph is optional, but encouraged for model reuse, improvement, and testing. The network is ready to be tested once the trained model has been generated using the latest checkpoint. The command "test image.py" runs a test programme on an image. The results are detailed in the Results and Discussion section.

E. Faster RCNN Performance evaluation with Data Augmentation

Class imbalance in the dataset, caused by a scarcity of pest images, can drastically influence classifier performance, resulting in a prediction bias towards the dominant class. Certain pests are restricted to specific crop and agro climatic zone conditions, making it impossible to catch all pest classes simultaneously [39].

The proposed work performs pest detection operations by including a dataset of imbalance classes and applying a Faster RCNN for increased accuracy. According to the training datasets listed in Table 2, the number of images captured for INS-A is greater than the number of images captured for INS-B and INS-C. The class divide is not negligible.

Sr.	**Pest**	**Identical Name**	**Number of Images**
1	*Phyllophaga*	INS-A	319
2	*Spodoptera litura*	INS-B	131
3	*Helicoverpa armigera*	INS-C	143

Table 2 : Identical name of Pest images and training size of class

By extending the dataset size and using an oversampling approach, we may enlarge limited datasets through image augmentation. In this study, we used horizontal flipping and rotation in 90-degree increments to improve the architecture of the configuration file. While image pre-processing methods are performed to both training and test sets, image augmentation manipulations are applied exclusively to training data[40]. The process flow for Faster RCNN is described in Figure 2 with the pre-processing of pest images.

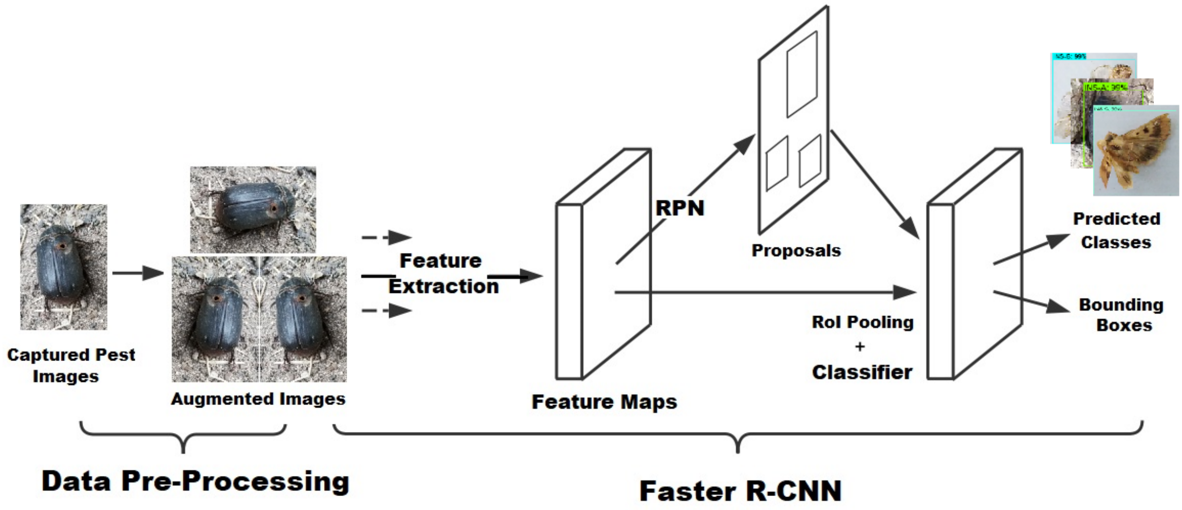

Figure 21 : Faster RCNN architecture with Data Augmentation technique

KBIS is a knowledge-based decision-making tool that makes decisions based on the facts and information stored in its database. It is built on the tornado web framework, and the backend makes judgments using the Faster-RCNN model. Farmers and others in the agricultural industry may profit from the technology by receiving more accurate pest advisory information.

Pest Detection System Working Model

The techniques outlined below in Table 3 apply to the entire pest detection system, which includes a knowledge-based decision-making tool. The procedure is separated into two stages: the first is the model training process, which is carried out as a backend process when necessary, and the second is the tool execution, which is carried out as a front process.

	Stage 1 : Training Process of Model
Step-1	Collect Pest Images
Step-2	Labelling the Images using LabelImg tools for training dataset
Step-3	Training dataset divide into 80:20 for training and validation respectively.
Step-4	Convert CSV File to TFRecords for Training
Step-5	Set the environment, related parameter and pre-processing for model
Step-6	Execute Faster-RCNN Model training process up to optimal learning rate
Step-7	Evaluate the model and check validation loss
Step-8	If Validation loss higher: <u>**go to Step-6**</u>
	Stage 2 : Execution of Tool
Step-1	Upload Image on Pest Identification System
Step-2	If Pest Found : <u>Go to Step-3</u>
Step-3	Display identified Pest and related advisory and END
Step-4	Else Pest Not Found : <u>Go to Step – 5</u>
Step-5	Move uploaded image to non-detection dataset
Step-6	Manual validation of non-detection dataset
Step-7	If non-detection dataset subset of the Pest Identification System: <u>Go to Step-2 of Stage: 1</u> <u>**Training Process of Model**</u> for Retraining in working model.

Table 3 Pest Detection System Working Model

Performance Evaluation Metrics

To compare the performance of the Deep CNNs architectures for Pest classification task, various evaluation metrics such as specificity, sensitivity and accuracy were used. According to Equations (1)-(4), True Positives (TP) is the numbers of specific Pests that predicted as specific Pests; False Negatives (FN) is the numbers of Pests not predicted as specific Pests. True Negatives (TN) is the numbers of non-Pests that not predicted as Pests, while False Positives (FP) is the numbers of non-Pests predicted as Pests.

Sensitivity is the measure of Pest detection that correctly classified and is expressed as Sensitivity or True Positive Rate (TPR).

$$TPR = \frac{TP}{(TP + FN)} \quad \ldots (1)$$

Specificity is the measure of non-Pest that not classified and is expressed as Specificity or True Negative Rate (TNR).

$$TNR = \frac{TN}{(TN + FP)} \quad \ldots (2)$$

Precision or positive predictive value measures what percentage of correctly classified labels is truly positive and is given as Positive Predictive Value(PPV) .

$$PPV = \frac{TP}{(TP + FP)} \quad \ldots (3)$$

Accuracy (ACC) is used to show the number of correctly classified Pest or non-Pest divided by the total number of Pests and is defined as ACC(%):

$$ACC\ (\%) = \frac{(TP + TN)}{(TP + TN + FP + FN)} X\ 100 \quad \ldots (4)$$

IV. RESULTS AND DISCUSSION

A. Comparisons of meta-architecture

This section illustrates the gradual demise of each meta-architecture over time as the epochs stabilize. All outcomes are quantified in terms of their sensitivity, precision, and accuracy. Finally, the overall performance of the system is evaluated.

Total Loss Function

The loss functions in Inception, Mobilenet, and Faster R-CNN all use a training image optimal value of 20k, 60k, and 20k epoch iterations. To ensure that the validation loss did not develop while the training was ongoing, 20% of the training data was divided for validation purposes. Detecting pests use an accurate model that exhibits model degradation. The model's accuracy and loss are determined using a variety of input images and epochs. Total losses for the SSD Inception, SSD Mobilenet, and Faster R-CNN architectures are 1.67, 1.05, and 0.10, respectively (Refer Figure 3).

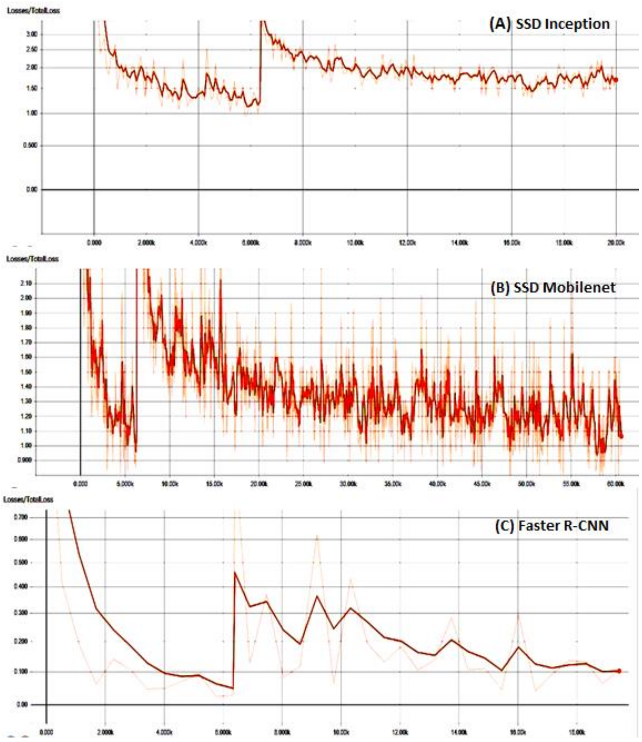

Figure 3 : Total Loss Function chart during transfer learning of (A) SSD Inception, (B) SSD Mobilenet, (C) Faster R-CNN Meta-architectures

Visual Comparative Evaluation

The three architectural versions are visualized using the developed object detector, which makes it simple to see their data and findings. To view the precise detection region, the bounding box reflects the class categorization and the percentage accuracy. Accuracy is a metric that indicates how precisely a class has been discovered. Not only are visualization figures used to demonstrate reliability and trustworthiness, but they also serve to convey assessment images. Faster-RCNN and SSD Mobilenet perform well on this small dataset of pest detection results. Figure 4 illustrates a few selected images, as well as backdrop complexity, variable caught images, and images with varying illumination.

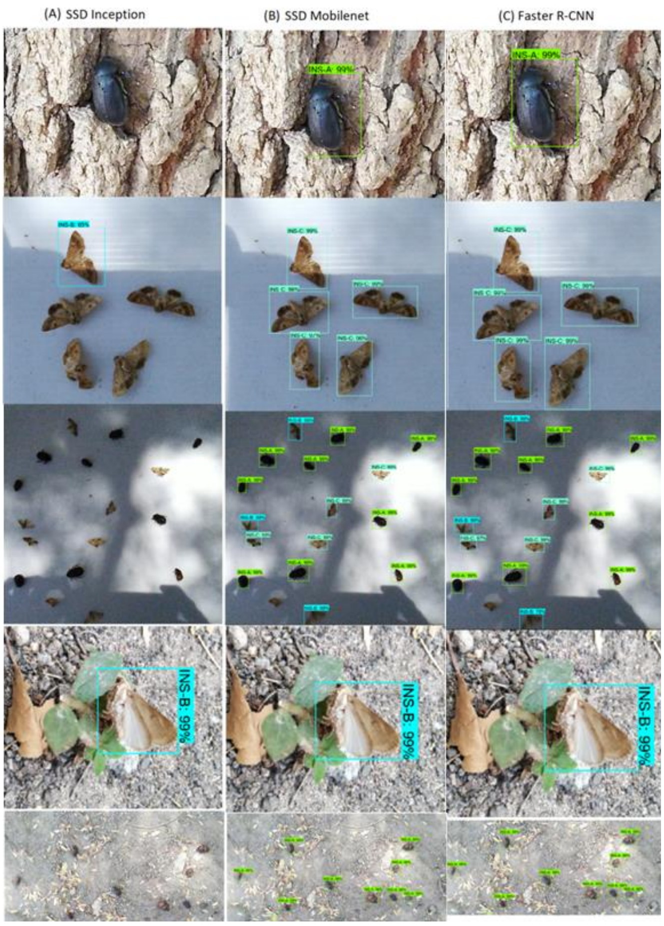

Figure 4 : Detection result of trained model (A) SSD Mobilenet (B) SSD Inception and (C) Faster R-CNN Meta-architectures

Performance Evaluation Metrics

Appendix-II contains the detailed findings of the evaluation for each architecture, as stated in Performance Evaluation Metrics. Table 4 summaries the Pest detection findings for three architectures in the form of confusion matrices. The faster R-CNN performed better on the majority of classes, including INS-A, INS-B, and INS-C, with an average accuracy of 95.33 percent. According to the data, the SSD Mobilenet meta-architecture is also functioning better, with an accuracy rate of 94.53 percent [41].

Meta-architectures	TPR(%)	TNR(%)	PPV(%)	ACC(%)
SSD Inception	22.43	91.72	73.08	57.05
SSD MobileNet	89.54	100	100	94.53
Faster R-CNN	91.06	100	100	95.33

Table 4 : Confusion Matrix - Show the obtained result of specificity, sensitivity and accuracy of different meta-architectures of CNN

B. Faster RCNN with data augmentation technique:

According to the last part, the Faster RCNN meta-architecture outperforms the other two prominent meta-architectures in terms of performance. It was determined to employ Faster RCNN to perform pest detection operations on specified Pests utilising various pest datasets and to begin constructing a new pest dataset. During the experiment uncover a problem with Pest availability varying by agricultural climate zone [42]. This section will begin with visual comparisons of the Faster-RCNN method's augmented and unaugmented trained models.

Visual Comparative Evaluation

The Faster-RCNN model with augmentation performed significantly better than the model without augmentation in this small dataset of pests applying detection findings. Figure 5 contains a selection of carefully picked photographs, as well as images with diverse degrees of backdrop complexity, images shot at various times of day, and images with varying illumination.

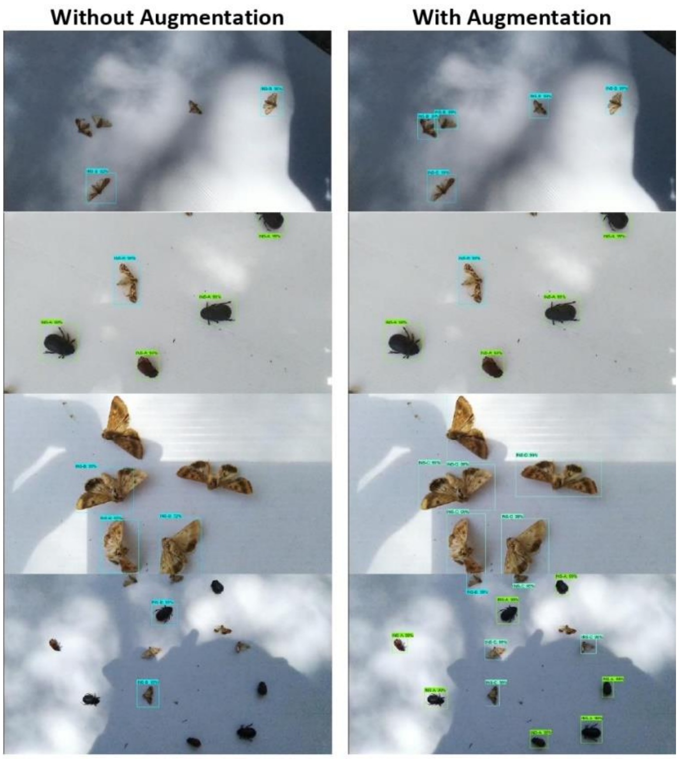

Figure 5. Results of with and without Augmentation method

Performance Evaluation Metrics

When the dataset used to train the classifier contains a mixture of class distributions, an unbalanced classification issue arises. Due to the fact that the pest dataset contains an unequal amount of cases of each class, an augmentation model was utilised to compensate. The results of Pest detection with and without augmentation are summarised in the form of confusion matrices, which are displayed in Table 5 according to the parameter stated in the section on performance evaluation metrics of the chapter materials and methods. When a problem of class imbalance emerges, the augmentation model outperforms the model without augmentation. According to our findings, the assessed model, which was created using a Faster RCNN meta-architecture and pre-processed using augmentation, has a 95.07 percent accuracy [43]. The training model process gets slower as a result of image augmentation as compared to when the model is not augmented, but the accuracy of the performance is improving.

Faster R-CNN Training Model	TPR (%)	TNR (%)	PPV (%)	ACC (%)
Without Augmentation	89.56	97.49	97.44	93.40
With Augmentation	91.02	99.45	99.44	95.07

Table 5. Confusion Matrix of with and without augmentation methods

V. CONCLUSION

The primary objective of this work is to construct a decision-making tool employing advanced deep learning algorithms that is based on classified findings from an inputted digital view of important nocturnal flying Pests. The experiment's primary objective is to enhance classification accuracy, and this objective is carefully addressed throughout the procedure. The pest detection problem by comparing three popular meta-architectures, including SSD Inception, SSD Mobilenet, and Faster RCNN of Convolutional Neural Network (CNN), to identify which architecture is optimal for this task. With an identification accuracy of 95.33 percent for chosen pest detection procedures on small datasets, the comparison results indicate that the state-of-the-art CNN meta-architecture Faster-RCNN outperformed. In comparison to the model without augmentation, the model with augmentation performs much better in the case of class imbalance problem. This demonstrates that when Faster RCNN was used in conjunction with augmentation, the results were considerably better than when no augmentation was employed during the available class imbalance problem. The results indicate that this strategy not only improves detection performance on a completely new pest dataset, but also preserves performance on an existing dataset.

VI. REFERENCES

[1] K. K. Shukla, D. J. Patel, and B. L. Radadiya, "Role of Information Technology in Improvement of Current Scenario in Agriculture," *Peer Rev. Res. J. Publ. By*, vol. 7, no. 3, pp. 390–395, 2014.

[2] U. K. Behera, S. K. Rautaray, A. K. Choudhary, and D. S. Rana, "Integrated farming system research in India: an overview," *Div. Agron.*, no. January 2013, pp. 40–78, 2013.

[3] D. J. Patel and N. Bhatt, "Analytical Review of Major Nocturnal Pests ' Detection Technique using Computer Vision," *Orient. J. Comput. Sci. Technol.*, vol. 11, no. 3, pp. 1–4, 2018, doi: 10.13005/ojcst11.03.06.

[4] R. Arora and S. Sandhu, "Breeding Pest resistant crops for sustainable agriculture," *Breed. Pest Resist. Crop. Sustain. Agric.*, pp. 1–421, 2017, doi: 10.1007/978-981-10-6056-4.

[5] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic," in *2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015*, 2015, pp. 500–505, doi: 10.1109/SPIN.2015.7095350.

[6] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[7] D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, *Computer Vision – ECCV 2014 - Part VII*, vol. 8695, no. Chapter 11. 2014.

[8] B. D. J. Hand and N. M. Adams, "Data mining," *Pc Ai*, vol. 15, no. 6, p. 11, 2001, doi: 10.1002/9781118445112.stat06466.pub2.

[9] N. K. Manaswi, "Deep Learning with Applications Using Python," pp. 97–104, 2018, doi: 10.1007/978-1-4842-3516-4.

[10] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *J. Agric. Sci.*, vol. 156, no. 3, pp. 1–11, 2018, doi: 10.1017/S0021859618000436.

[11] C. Papageorgiou, "A Trainable System for Object Detection in Images and Video Sequences," *Int. J. Comput. Vis.*, vol. 38, no. 1685, pp. 15–33, 2000, doi: 10.1023/A:1008162616689.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008, doi: 10.1016/j.cviu.2007.09.014.

[13] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, 2016, vol. 9908, pp. 354–370, doi: 10.1007/978-3-319-46493-0.

[14] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, 2014, doi: 10.1109/LGRS.2014.2309695.

[15] M. Eisenbach, D. Seichter, T. Wengefeld, and H. M. Gross, "Cooperative multi-scale Convolutional Neural Networks for person detection," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016-Octob, pp. 267–276, 2016, doi:

10.1109/IJCNN.2016.7727208.

[16] S. Cui, P. Ling, H. Zhu, and H. M. Keener, "Plant pest detection using an artificial nose system: A review," *Sensors (Switzerland)*, vol. 18, no. 2, pp. 1–18, 2018, doi: 10.3390/s18020378.

[17] P. Boissard, V. Martin, and S. Moisan, "A cognitive vision approach to early pest detection in greenhouse crops," *Comput. Electron. Agric.*, vol. 62, no. 2, pp. 81–93, 2008, doi: 10.1016/j.compag.2007.11.009.

[18] L. Liu *et al.*, "PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification," *IEEE Access*, vol. 7, pp. 45301–45312, 2019, doi: 10.1109/ACCESS.2019.2909522.

[19] D. Moshou, C. Bravo, J. West, S. Wahlen, A. McCartney, and H. Ramon, "Automatic detection of 'yellow rust' in wheat using reflectance measurements and neural networks," *Comput. Electron. Agric.*, vol. 44, no. 3, pp. 173–188, 2004, doi: 10.1016/j.compag.2004.04.003.

[20] S. Coulibaly, B. Kamsu-foguem, and D. Kamissoko, "Computers in Industry Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, 2019, doi: 10.1016/j.compind.2019.02.003.

[21] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 157–177, 2018, doi: 10.1016/j.cosrev.2018.03.001.

[22] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.

[23] E. Karami, S. Prasad, and M. Shehata, "Image Matching Using SIFT , SURF , BRIEF and ORB : Performance Comparison for Distorted Images Image Matching Using SIFT , SURF , BRIEF and ORB : Performance Comparison for Distorted Images," in *Newfoundland Electrical and Computer Engineering Conference*, 2015, no. November, p. 4, doi: 10.13140/RG.2.1.1558.3762.

[24] D. J. Patel and K. K. Shukla, "Challenges and Opportunities for ICT Initiatives in Agricultural Marketing in India," *Orient. J. Comput. Sci. Technol.*, vol. 7, no. 3, pp. 377–381, 2014.

[25] Z. Wang, K. Liu, J. Li, Y. Zhu, and Y. Zhang, "Various Frameworks and Libraries of Machine Learning and Deep Learning: A Survey," *Arch. Comput. Methods Eng.*, 2019, doi: 10.1007/s11831-018-09312-w.

[26] D. Hubel and T. Wiesel, "RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT ' S STRIATE CORTEX By D . H . HUBEL * AND T . N . WIESEL * From the Wilmer Institute , The Johns Hopkins Hospital and In the central nervous system the visual pathway from retina to striate cortex provides an," *J. Physiol.*, vol. 148(3), 57, no. 12, pp. 574–591, 1959, [Online]. Available: http://doi.wiley.com/10.1113/jphysiol.2009.174151.

[27] C. C. Aggarwal, *Neural Networks and Deep Learning*. 2018.

[28] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8695 LNCS, no. PART 7, pp. 329–344, 2014, doi: 10.1007/978-3-319-10584-0_22.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image

recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[30] S. Goswami, "A deeper look at how Faster-RCNN works," *medium.com*, 2018. https://whatdhack.medium.com/a-deeper-look-at-how-faster-rcnn-works-84081284e1cd.

[31] C. Ning, H. Zhou, Y. Song, and J. Tang, "Inception Single Shot MultiBox Detector for object detection," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2017, no. July, pp. 549–554.

[32] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.

[33] Y. C. Chiu, C. Y. Tsai, M. Da Ruan, G. Y. Shen, and T. T. Lee, "Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems," in *2020 International Conference on System Science and Engineering, ICSSE 2020*, 2020, pp. 0–4, doi: 10.1109/ICSSE50014.2020.9219319.

[34] J. V. Dillon *et al.*, "TensorFlow Distributions," 2017, [Online]. Available: http://arxiv.org/abs/1711.10604.

[35] Github.com/keras-team/keras, "About Keras," *keras.io*, 2015. https://keras.io/about/.

[36] TzuTa, "LabelImg," *Github*, 2015. https://github.com/tzutalin/labelImg.

[37] Tensorflow, "Tutorial Tensorflow," *Official Webpage*, 2017. https://www.tensorflow.org/tutorials.

[38] Morgan, "TensorFlow: How to freeze a model and serve it with a python API," *blog.metaflow.fr*, 2016. https://blog.metaflow.fr/tensorflow-how-to-freeze-a-model-and-serve-it-with-a-python-api-d4f3596b3adc.

[39] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0151-6.

[40] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[41] D. J. Patel and N. Bhatt, "Pest Identification Among Deep Learning ' s Meta-architectures Using TensorFlow," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 1910–1914, 2019, doi: 10.35940/ijeat.A1031.109119.

[42] K. G. Bapatla, S. K. Singh, V. Sengottaiyan, B. Singh, and N. P. Singh, "Seasonal patterns of Pest pest in major pigeonpea and chickpea growing agro-climatic zones of India and their management inferences," *Int. J. Trop. Pest Sci.*, vol. 41, no. 2, pp. 1601–1609, 2021, doi: 10.1007/s42690-020-00361-y.

[43] D. Patel and N. Bhatt, "Improved accuracy of pest detection using augmentation approach with Faster R-CNN," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1042, no. 1, p. 012020, doi: 10.1088/1757-899x/1042/1/012020.