(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.10, Issue No.6, October – November 2022, 10222 - 10228.

Novel Proportionate Scrutiny On Crop Protection From Creatures By Deep Learning

Mr. D. SRINIVAS
Assistant Professor, Dept. of CAI, Kakinada Institute of Engineering & Technology, A.P., India

Mr. V. VEERENDRA SUBHASH
PG Scholar, Dept. of CAI, Kakinada Institute of Engineering & Technology, A.P., India

Abstract: **The main objective of this paper is to protect the crop from animal attacks. The conventional techniques have the same kind of security applied to all the types of animals detected based on a Passive IR sensor, and only single-stage protection is applied. The images were captured and identified with the help of machine learning and deep learning techniques. The project was designed with a rectangular farm area. On each side of the entrance, the device was installed to capture the image for processing to identify the animals, based on the animal identification, different levels of security were applied, and that will produce different sounds with different Db levels and variety of dazzling light. This work provides a comprehensive description of the design, development, and assessment of an intelligent animal repelling system that allows for to detection and recognition of the animals. The enhancement is done by different levels of protection and different types of protection based on the classified animals. In initial level protection, making the noise and lightning from the opposite side send the animal out of the farm. If the animals are still on the farm, initiating the next stage that the image will send to the owner. The accuracy of all the methods discussed will be compared based on the complexity of the technique, implementation cost, reciprocating time, and accuracy of animal detection. In recent years, edge computing has become an essential technology for real-time application development by moving processing and storage capabilities close to ending devices, thereby reducing latency, improving response time, and ensuring secure data exchange.**

I. INTRODUCTION

Real-time crop protection is one of the challenging issues. To protect the crops from wild animals attack has a wide scope to perform research. The repelling device produces noise and dazzling lights based on the real-time detection of the animal. This helps the farmers to get more productivity and reduces farmer suicides.

Objectives

- Design the repelling device based on the solar and work independently withoutdepending on the external supply.
- Analyze and evaluate the performance of the Machine Learning and Deep Learning techniques to classify wild animals using performance matrix.
- Design the repelling system to produce the actions depending on the detected animals.

The detailed explanation related Image classification and animal detection techniques based on the Machine Learning and Deep Learning techniques. From the literature, the computational time of the classification of the animal and the producing repelling action is a challenging issue. A proper independently operating device for animal detection and repelling was necessary, lack of the design of such a device was identified in the literature. This paper work is designed to fulfill the gaps identified in the literature and Proportionate Scrutiny on Crop Protection from creatures.

Table 1: Percentage of damage caused by the different wild animals

S. N	Wild animal	Name of the crop	Damage	Reciprocating Action
1	Elephant	Sugarcane	72%	Use Bright Lights,
2	Gaur	Mulberry, sandal	62%	High-frequency
3	Sambar deer	Pasture, forestry	17%	Loud noise and
4	Wild boar	Paddy, maize,	16%	Loud noise
5	Monkey	*maize*, wheat,	75%	Lighting with sound
6	Porcupine	*maize*, potatoes,	65%	Dazzling of lights
7	Goral	Maize, Potato,	20%	Loud noise and
8	Bear	field corn,	55%	Fire and
9	Wolf	rice, wheat,	18%	Loud noise and
10	Zebra	Maize or corn,	15%	Loud noise and

II PROPOSED SYSTEM

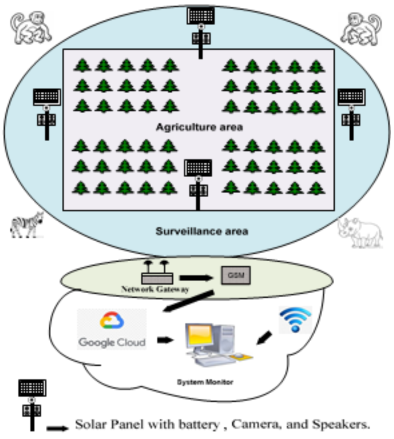


Figure 1 Block diagram of the proposed system

A total of 10 animals were considered in the database for the identification of the list of 10 animals and the damage created by the animals to the specific crops, the percentage of damage to the various crops, and what will be the repelling action for that animal described in table 1. The animals elephant, monkey, gaur, and Porcupine create damage to more than 60% of. The animal sambar deer, wild boar, and wolf created damage to the crops by less than 20%. The majority of the animals attack the maize and vegetable crops. The majority of the animals can have the fear of sounds. The proposed test system and device are shown in figure 1.

The test system consists of the sample area of the monitored agricultural land. The animal repelling system consists of a solar panel with a battery to operate remotely without any power from the external source, a camera was fixed below the solar panel to get the images as well as the video of the farm, and the bottom of the pole consists of speaker and lighting equipment to produce necessary repelling depending upon the detecting animal.

Methods:

7 different algorithms were chosen for the classification of animals. All these methods were compared on the same test conditions and compared the results. The methods were explained below with a detailed algorithm.

K-Nearest Neighbor algorithm (KNN)

KNN is a technique for categorizing objects based on the feature spaces nearest training samples. This algorithm's training phase consists solely of storing the training pictures feature vectors and labels. The classification task entails labeling a testing example using the majority labels of its k closest neighbors.

The KNN algorithm benefits include its simplicity and capacity to handle several classes. However, using all of the features equally for similarity computing is a major disadvantage of the KNN algorithm. This can result in classification mistakes, especially if only a limited number of characteristics are helpful for classification. It accepts that comparable objects are near one another. The KNN algorithm uses simple arithmetic concepts. The straight-line distance, often known as Euclidean distance, is a common and well-known method of determining distance.

The KNN classifier is by far the most simple machine learning and image classification algorithm. It's so simple that it doesn't actually "learn" anything. Instead, this algorithm directly relies on the distance between feature vectors.

Calculating distance:

The first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are Euclidian, Manhattan (for continuous), and Hamming distance.

Euclidian Distance: This is calculated as the square root of the sum of the squared differences between a new point (q) and an existing point (p).

Manhattan Distance: This is the distance between real vectors using the sum of their absolutedifference.

Hamming Distance: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0. Otherwise D=1.

The Dataset

The Animals dataset is a simple example dataset to demonstrate how to train image classifiers using simple machine learning techniques as well as advanced deep learning algorithms. Images inside the Animals dataset belong to ten distinct classes with 1,000 example images per class.

To get the right K, you should run the KNN algorithm several times with different values of K and select the one that has the least number of errors.

The right K must be able to predict data that it hasn't seen before accurately.

Step 1. A data set is usually required when developing an algorithm. As a result, you begin by loading the training and test data.

Step 2. Select the closest data points (the value of K). Any integer can be used as K.

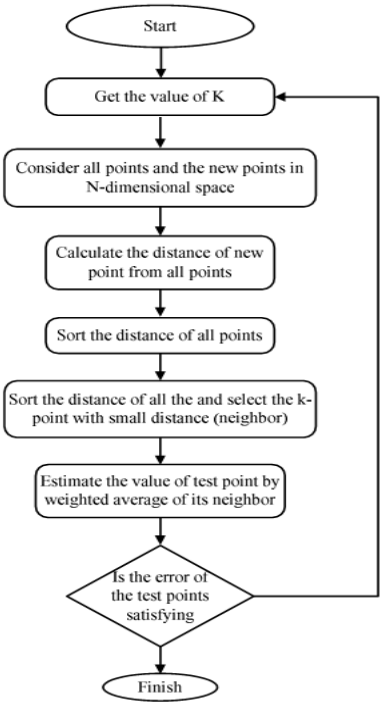
Step 3. Carry out the following steps for each set of test data: –

- ➢ Calculate the distance between test data and each row of training using

Euclidean, Hamming, or Manhattan distances. When determining distance, the Euclidean approach is the most used.

- Sort the data set by distance value in ascending order.
- Select the top K rows from the sorted array.
- It will assign a class to the test point based on the most often occurring class in these rows.

Step 4. End



KNN Flow chart

Figure 2 KNN Algorithm flow chart

Support Vector Machine (SVM) algorithm

SVM is a mapping of examples as points in space, separated by a dividing plane that optimizes the margin between them. SVM's key benefit is that it can use the kernel method to do non-linear classification. However, one of the key drawbacks of SVM classification is the algorithm's speed and size restrictions during both the training and testing stages.

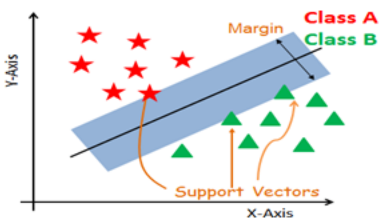


Figure 3 SVM algorithm

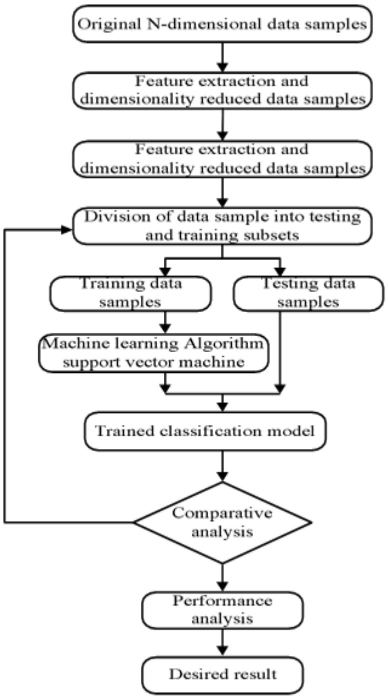


Figure 4 Support Vector Machine algorithm flow chart

Multi-Layer Perceptron (MLP)

MLP differs from logistic regression in that one or more non-linear layers, known as hidden layers, can exist between the input and output layers. MLP's capacity to train non-linear models is one of its primary advantages. However, there are significant drawbacks, including a non-convex loss function and the need to tune for the number of hidden layers and neurons. MLPs are a great place to start learning more about deep learning.

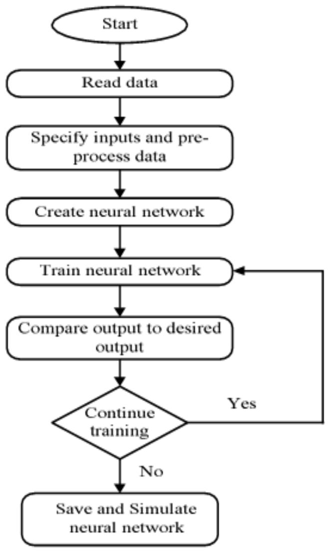


Figure 5 Multi-layer Perceptron algorithm flow chart

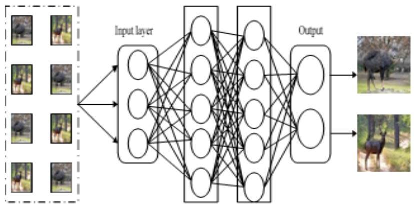


Figure 6 MLP classification

Convolutional Neural Networks (CNNs)

CNNs have emerged as the master algorithm in computer vision in recent years, with increased computational power allowing them to achieve superhuman performance on some complex visual tasks, and Xception is one of the architectures that has been implemented in this project via transfer learning. It integrates the features of numerous well-known CNN architectures, including Inception-v4, GoogLeNet, and ResNet, but it replaces the inception modules with a depth wise separable convolutional layer.

Unlike traditional convolutions, which learn both spatial and cross-channel patterns simultaneously, the separable convolution layer splits feature learning into two phases, in the first phase, each input feature map is subjected to a single spatial filter, followed by a search for cross-channel patterns in the second phase. The detailed classification of images were shown in figure 7.

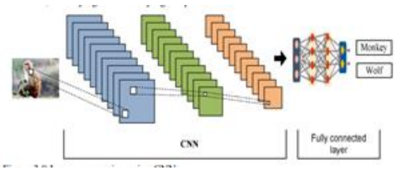


Figure 7 Image processing using CNN

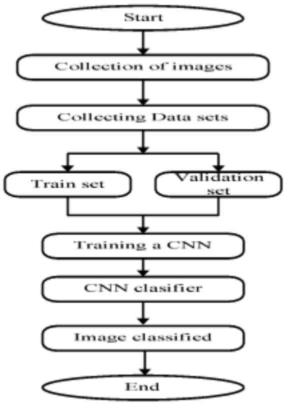


Figure 8 CNN algorithm flow chart

Recurrent Neural Networks (RNNs)

RNN are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words.

The outputs from the LSTM can be fed as inputs to the current phase using RNNs with directed cycles.

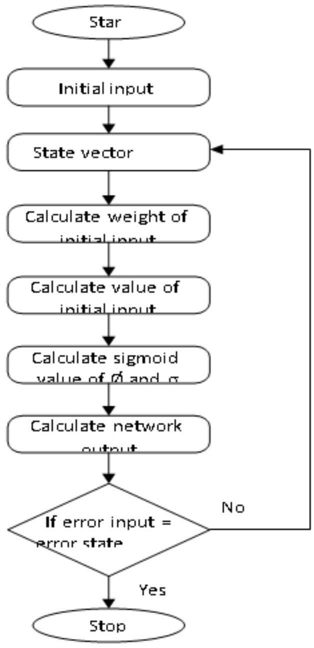


Figure 9 RNN algorithm flow chart

The LSTM's output is used as an input in the current phase, and its internal memory allows it to remember past inputs. Image captioning, time series analysis, natural language processing, handwriting identification, and machine translation are all prominent applications for RNNs.

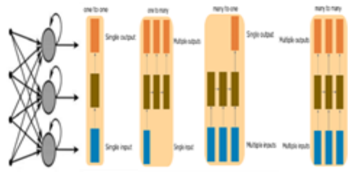


Figure 10 a) Recurrent Neural Network b) one to one c) one to many d) Many to one e) Many to many

Generative Adversarial Networks (GANs)

GANs are deep learning generative algorithms that generate new data instances that are similar to the training data. GAN is made up of two parts: a generator that learns to produce fake data and a discriminator that learns from that data. GANs have been increasingly popular over time. They may be used to enhance astronomy photographs as well as to imitate gravitational lensing for dark matter investigations. GANs are used by video game producers to upgrade low-resolution, 2D graphics in older games by using image training to recreate them in 4K or greater resolutions. GANs aid in the creation of realistic pictures and cartoon characters, as well as the creation of photos of human faces and the rendering of 3D objects.

Self-Organizing Maps (SOMs)

The SOMs, which enable data visualization by using self-organizing artificial neural networks, minimize the dimensionality of data. The problem of humans being unable to visualize high-dimensional data is addressed through data visualization. SOMs are designed to assist people in comprehending this multi-dimensional data.

SOMs or the Kohonen neural networks have high popularity in the competitive neural network field [12.]. SOMs bring the advantages of both clustering and projection methods together. Basically, SOMs analyze the data in an unsupervised manner but there are also several supervised variants like Counter Propagation Artificial Neural Network (CP-ANN), Supervised Kohonen Networks (SKN), and XY-Fused networks (XYF) [13]. The Kohonen network is based on a single layer of neurons that are arranged in two dimensions. Each neuron in the map is included in a weight vector. The number of elements in the weight vectors is equal to the number of variables in the input vector. The elements of all weight vectors (neurons) should be randomly initialized before training the network. In the training phase, each input vector is presented to the network and the most similar neuron (winner neuron) to this input can be found due to the minimal Euclidean distance between the input vector and weight vectors of the neurons.

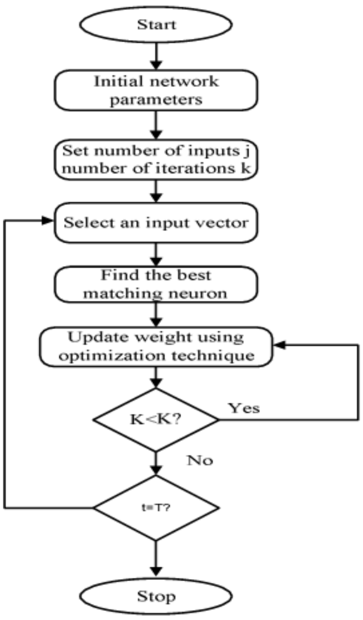


Figure 11 Flow chart of the SOM algorithm

SOMs use a vector at random from the training data to initialize weights for each node. SOMs look at each node to see which weights are most likely to be the input vector. The Best Matching Unit is the winning node (BMU). The BMU's neighborhood is discovered through SOMs, and the number of neighbors decreases with time. The sample vector is given a winning weight using SOMs. The weight of a node varies as it gets closer to a BMU. The more away a neighbor is from the BMU, the less it learns from it. For N iterations, SOMs repeat step two.

A diagram of an input vector with various colors may be found below. This information is sent into a SOM, which transforms it into 2D RGB values. Finally, it separates and categorizes the different colors.

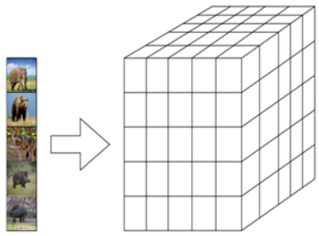


Figure 12 SOM algorithm image processing

III IMPLEMENTATION

Ten wild animal photographs were obtained using a 108-megapixel camera in various weather circumstances. Image resolution may be enhanced to accommodate huge images, but this comes at the cost of greater memory and computing needs. Data argumentation was required before training the picture to boost image variety and improve image representation. It was important to pre-process the pictures with augmentation methods such as jitter, image rotation, flipping, cropping, multi-scale transformation, hue, saturation, Gaussian noise, and intensity to get improved identification accuracy and resilience offered by effective data augmentation.

While training and validating a neural network, Tensorflow is the greatest tool for visualizing various metrics. To check for further details in most cases, such as how a model performs on validation data and to figure out the performance of validation data when training and validation loss and accuracy aren't enough.

The accuracy of each method indicated in the table is calculated using the confusion matrix, to show the performance of all the methods 4 matrices where those formed are Confusion matrix, Precision, Recall, and F1 score. A confusion matrix, also known as an error matrix, is a special table structure that permits visualization of the performance of an algorithm, often a supervised learning one, in the field of machine learning and specifically the issue of statistical classification. The examples in a predicted class are represented by the rows of the matrix, whereas the instances in an actual class are represented by the columns. The term comes from the fact that it's simple to tell whether the system is mixing up two types (i.e. commonly mislabeling one as another).

Deep learning methodologies

A confusion matrix was built to understand the complexity, accuracy, and validation of the method. The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive. The precision reflects how reliable the model is in classifying samples as Positive.

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected. The recall cares only about how the positive samples are classified. This is independent of how the negative samples are classified, e.g. for precision. When the model classifies all the positive samples as Positive, then the recall will be 100% even if all the negative samples were incorrectly classified as Positive.

F1-score is the harmonic mean of precision and recall. It combines precision and recalls into a single number using the following formula:

$$\text{F1} - \text{Score} = 2X \frac{Precision * Recall}{Precision + Recall}$$

The higher the precision and recall, the higher the F1 score. F1-score ranges between 0 and 1. The closer it is to 1, the better the model. For all the methods sample 1000 images were considered for testing.

IV PROPORTIONATE SCRUTINY

All the obtained results illustrate below

S. No	Method	Complexity	Classification time in ms	Time taken for Replication Action ms	Classification Efficiency
1	KNN	Medium	176	232	89.30
2	SVM	Complex	201	255	90.35
3	MLP	Medium	198	252	91.32
4	CNN	Low	140	194	92.68
5	RNN	Low	156	210	91.48
6	GAN	Low	149	203	91.96
7	SOM	Medium	189	243	89.60

Table 2: All NN methods were Scrutinized

Convolution neural network provides better results compared to the other methods discussed in the report. The results were based on the Confusion matrix, Precision, Recall, and F1 Score. The summaries result in the accuracy of all the methods tabulated.

V CONCLUSION

In this work, the test was carried out to classify the various wild animals to protect the crops, after identification repelling action needs to be initiated as part of the protection scheme. The research results using deep learning methods can save crop damage from wild animal attacks. Various methods were used to archive the crop protection, among the described methods, CNN provided the best results with the identification accuracy of 92.68% and the repelling time of less than 200ms. All the described methods were validated through the software same test system using python code and Tensor Flow.

The proposed hardware setup can be developed a future work, the cost estimation and full implementation need a lot of mechanical and electrical-related work. The combined project will stand as interdisciplinary research work and novel in nature.

REFERENCES

[1] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, ‘‘From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges,’’ IEEE Trans. Ind. Information., vol. 17, no. 6, pp. 4322–4334, Jun. 2021.

[2] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, ‘‘A survey on the role of IoT in agriculture for the implementation of smart farming,’’ IEEE Access, vol. 7, pp. 156237– 156271, 2019.

[3] K. Kirkpatrick, ‘‘Technologizing agriculture,’’ Commun. ACM, vol. 62, no. 2, pp. 14–16, Jan. 2019.

[4] M. O. Ojo, D. Adami, and S. Giordano, ‘‘Network performance evaluation of a LoRa- based IoT system for crop protection against ungulates,’’ in Proc. IEEE 25th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD), Sep. 2020, pp. 1–6.

[5] A. Levisse, M. Rios, W.-A. Simon, P.-E. Gaillardon, and D. Atienza, ‘‘Functionality enhanced memories for edge-AI embedded systems,’’ in Proc. 19th Non-Volatile Memory Technol. Symp. (NVMTS), Oct. 2019, pp. 1–4.

[6] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, ‘‘Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey,’’ J. Netw. Comput. Appl., vol. 181, no. 1, 2021, Art. no. 103005.

[7] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, ‘‘Industrial edge computing: Enabling embedded intelligence,’’ IEEE Ind. Electron. Mag., vol. 13, no. 4, pp. 48–56, Dec. 2019.

[8] E. Li, L. Zeng, Z. Zhou, and X. Chen,

‘‘Edge AI: On-demand accelerating deep neural network inference via edge computing,’’ IEEE Trans. Wireless Commun., vol. 19, no. 1, pp. 447–457, Jan. 2020.

[9] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, ‘‘Edge intelligence: Paving the last mile of artificial intelligence with edge computing,’’ IEEE Proc., vol. 107, no. 8, pp. 1738– 1762, Aug. 2019.

[10] G. Codeluppi, A. Cilfone, L. Davoli, and G. Ferrari, ‘‘LoRaFarM: A LoRaWAN-based smart farming modular IoT architecture,’’ Sensors, vol. 20, no. 7, p. 2028, Apr. 2020.

[11] M. O. Ojo, D. Adami, and S. Giordano, ‘‘Experimental evaluation of a LoRa wildlife monitoring network in a forest vegetation area,’’ Future Internet, vol. 13, no. 5, p. 115, Apr. 2021.

[12] I. Martinez-Alpiste, P. Casaseca-de-la Higuera, J. Alcaraz-Calero, C. Grecos, and Q. Wang, ‘‘Benchmarking machine-learning-based object detection on a UAV and mobile platform,’’ in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC) Apr. 2019, pp. 1–6.

[13] Y. Yu, K. Zhang, D. Zhang, L. Yang, and T. Cui, ‘‘Optimized faster R-CNN for fruit detection of strawberry harvesting robot,’’ in Proc. ASABE Annu. Int. Meeting, 2019, p. 1.

[14] R. Shi, T. Li, and Y. Yamaguchi, ‘‘an attribution-based pruning method for real-time mango detection with YOLO network,’’ Comput. Electron. Agricult, vol. 169, Feb. 2020, Art. no. 105214.

[15] J. Wang, M. Shen, L. Liu, Y. Xu, and C. Okinda, ‘‘Recognition and classification of broiler droppings based on deep convolutional neural network,’’ J. Sensors, vol. 2019, pp. 1–10, Nov. 2019.

[16] J. Redmon and A. Farhadi, ‘‘YOLO9000: Better, faster, stronger,’’ in Proc. IEEE Conf. Comput. Vis. pattern Recognit., Jul. 2017, pp. 7263–7271.

[17] J. Redmon and A. Farhadi, ‘‘YOLOv3: An incremental improvement,’’ 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804. 02767

[18] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.

[19] K. Langendoen, A. Baggio, and O. Visser, ‘‘Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture,’’ in Proc. 20th IEEE Int. Parallel Distrib. Process. Symp., Apr. 2006, p. 8.

[20] Studies on Crop Damage by Wild Animals in Kerala and Evaluation of Control Measures, Kerala Forest Research Institute Peechi, Thrissur, May 1999.

[21] Mehta, Piyush, et al. "A study on managing crop damage by wild animals in Himachal Pradesh." International Journal of Agriculture Sciences. Bioinfo Publications, New Delhi, India 10.12 (2018): 6438-6442.

