

“Smart Animal Repelling System for Crop Protection Using Deep Learning”

VIPPE SATYANARAYANA¹, Miss G.PRAMEELA², Mr.K.SURESH³

M.Tech Research scholar in Department of computer science and engineering & KITS, DIVILI
Assistant professor in Department of computer science and engineering & KITS, DIVILI College
Assistant professor in Department of computer science and engineering & KITS, DIVILI College

Abstract - 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. Much research and numerous attempts have been made to apply the new IoT technology to agricultural areas.

The main objective of this report 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.

Key Words: Deep Learning, Artificial Intelligence, Animal Detection, Crop proection, Smart Framing, IoT Based Agriculture, Image Processing, Ultrasonic Repelling System, Farm Security, Eco-Friendly Framing

1.INTRODUCTION

Wild animal detection is an interesting area of research. This interest arises due to the increasing threat of animal rustling in agriculture fields. Hence to best protect and monitor the livestock or the conservation of wild animals, there is a need to deploy technological systems with the competence to combat the above-stated problems. One such technological system is the use of neural network systems and computer vision techniques combined with Machine learning-based methodologies algorithms and Deep neural learning techniques algorithms to deal with these problems. This thesis concentrates on the use of computer vision techniques, Machine learning-based methodologies, and Deep neural learning techniques for performing detection of wild animal attacks on the fields. Once the animal is detected the repelling

system will produce necessary action. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process

2. Objectives

The following objectives were addressed in this project

Design the repelling device developed using the solar and work independently without depending on the external supply. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process.

Analyze and evaluate the performance of the Machine learning-based methodologies and Deep neural learning techniques to classify wild animals using performance matrix. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process.

Design the repelling system to produce the actions depending on the detected animals. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process

3. Real-world problems of Image classification

Before going through the various picture categorization techniques that can be utilized. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process.

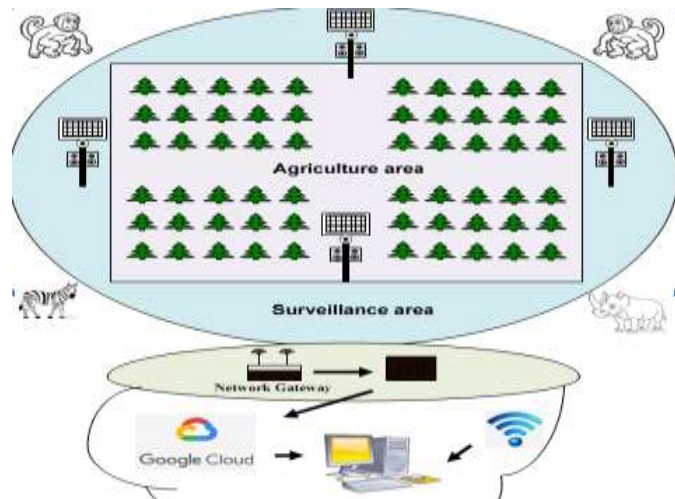
Let's have a look at some of the computer vision issues that would be easy for a person to complete:

Viewpoint variation, ii) Scale variation, iii) Deformation, iv) Occlusion, v) Illumination conditions, vi) Background clutter, vii) Intra-class variation



4. Test system description

The test system was designed by considering the agriculture crop field of half hectors land. At the Center of each side land, the pole arrangement of the camera with audio and lighting equipment was placed to identify the animal and the repelling action need to produce developed using the identification. Furthermore, the obtained observations were carefully examined to ensure reliable interpretation of the experimental process.



S. No	Wild animal name	Image of the animal	Name of the crop	Percentage of damage	Reciprocating action
1	Elephant		Sugarcane, Coconut, plantain, paddy, maize	72%	Use Bright Lights, Noise
2	Gaur		Mulberry, sandal	62%	High-frequency sound waves
3	Sambar deer		pasture, forestry plantations, gardens	17%	Loud noise and dazzling lights
4	Wild boar		Paddy, maize, bean corn, and fruit trees	16%	Loud noise
5	Monkey		maize, wheat, rice, vegetable crops	75%	Lighting with sound
6	Porcupine		maize, potatoes, groundnuts, sugarcane	65%	Dazzling of lights
7	Goral		Maize, Potato, Millet, Wheat, Paddy	20%	Loud noise and dazzling lights
8	Bear		field corn, oats, and sweet corn	55%	Fire and noise
9	Wolf		rice, wheat, maize, pulses, and mustard	18%	Loud noise and dazzling lights
10	Zebra		Maize or corn, potato, tomato, carrot, and other vegetables.	15%	Loud noise and dazzling lights

The test architecture consists of the sample area of the monitored agricultural land. The animal repelling architecture 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

5.Introduction to the literature

To get a good understanding of the topic, a literature review must be done, here for The present investigation, all types of documents, reports, journal articles, conference papers, and review papers were considered. A total of 75 reference documents were analyzed to get a better insight into the animal detection and repelling system

Agriculture is vital to the economies of many countries throughout the world. Despite economic progress, agriculture remains the economy's backbone. Agriculture is the economy's mainstay. It adds to the country's gross domestic product. Agriculture satisfies people's dietary needs while also producing a variety of raw resources for industry. However, there will be significant crop loss due to animal interference and fires in agricultural fields. The crop will be completely ruined. There will be a significant number of farmer losses [1].

It is critical to preserve agricultural fields and farms from animal and fire damage to minimize financial losses. To address this issue, a computational approach to prevent animals from entering the farm in our planned work. The main goal of The present investigation is to create an intruder alarm system for the farm to prevent losses due to animals and fire. These intruder alerts safeguard the crop from damage, increasing agricultural output. The embedded development system will not be hazardous to animals or humans. The goal of the project is to create an intelligent security system for agricultural safety [2].

Because they are noninvasive and simpler to operate, motion-sensitive wildlife cameras, sometimes known as camera traps, are becoming a more popular survey tool for animal populations [3]. Camera traps are the most successful and cost-effective solution for numerous species when compared to other wildlife monitoring approaches [4]. Ambitious initiatives are expanding the scale at which cameras are utilized on the landscape, with hundreds of sensors currently rotating across thousands of sites [5], sometimes with citizen scientists' aid [6].

6.Methods

7different algorithms were chosen for the categorization task 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

1. K-Nearest Neighbor algorithm (KNN)
2. Support Vector Machine (SVM) algorithm
3. Multi-Layer Perception (MLP)
4. Convolutional Neural Networks (CNNs)
5. Recurrent Neural Networks (RNNs)

6. Generative Adversarial Networks (GANs)

7. Self-Organizing Maps (SOMs)

7. 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).

The distance used here is the Euclidean distance:

$$d(p, q) = \sqrt{\sum_{i=1}^K (q_i - p_i)^2} \quad (1)$$

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

$$d(p, q) = \sum_{i=1}^K |q_i - p_i| \quad (2)$$

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.

$$D_H = \sum_{i=1}^K |(q_i - p_i)| \quad (3)$$

$$q = p \rightarrow D=0$$

$$q \neq p \rightarrow D=1$$

The Dataset

The Animals data collection is a simple example data collection to demonstrate how to train visual sample classifiers using simple ML techniques as well as advanced DL models algorithms. Images inside the Animals data collection 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 utilized as K.

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

1.calculate the distance between test data and each row of training using Euclidean, hamming or manhattan distance

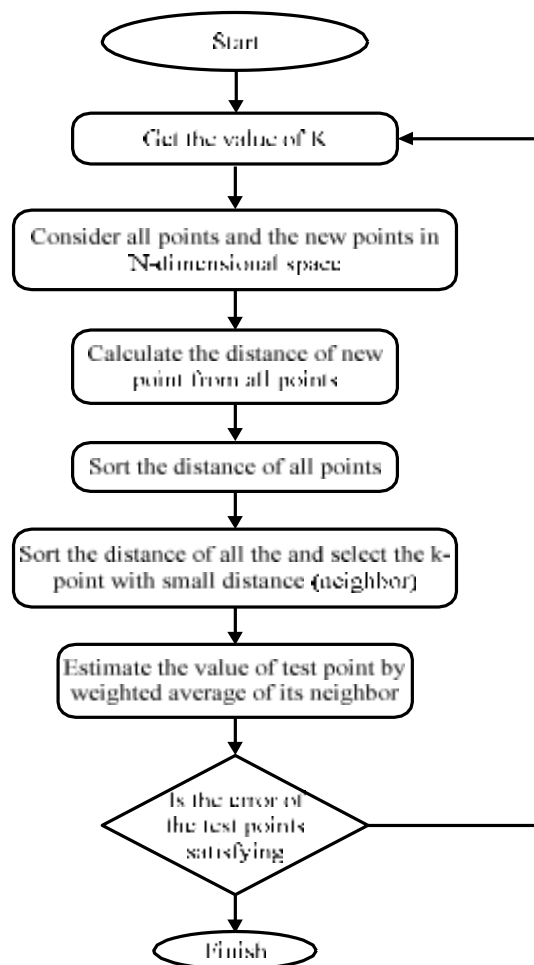
when determining distance the Euclidean approach is the most used.

2.sort the data set by distance value in ascending order.

3.Select the top K rows from the sorted array.

4.It will assign a class to the test point based on the most often occurring class in these rows.

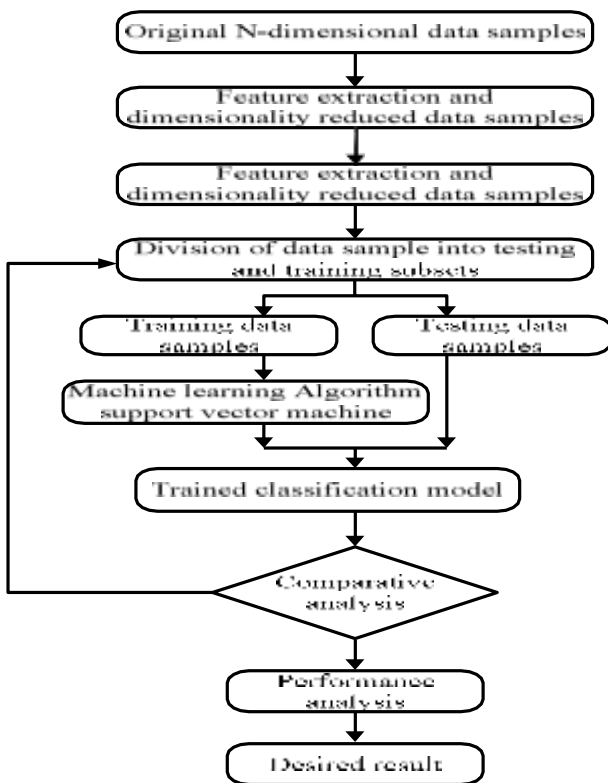
Step:4 .End



KNN Algorithm flow chart

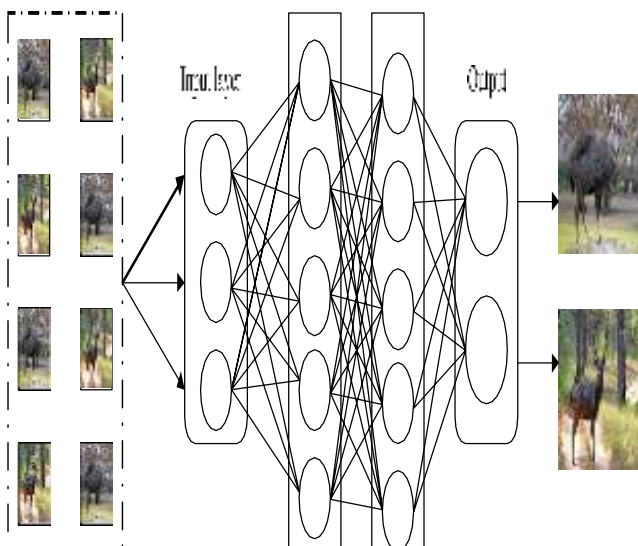
SVM is a mapping of examples as points in space, separated by a dividing plane that optimizes the margin between them. SVM's key benefit is that it can use the kernel method to do non-linear classification. However, one of the key drawbacks of SVM classification is the algorithm's speed and size restrictions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

SVM algorithm in which a line will be drawn between two different categories to differentiate between the classes. SVM is also known as the support vector network.



Support Vector Machine algorithm flow chart

MLPs use the weights that exist between the input layer and the hidden layers to compute the input. To decide which nodes to fire, MLPs employ activation functions. ReLUs, sigmoid functions, and tanh are all activation functions. From a training data set, MLPs train the framework to hold the correlation and learn the dependencies between the independent and target variables. An MLP is shown below in a example. To categorize photos of deer and goral, the diagram computes weights and bias, and applies appropriate activation functions.



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 The present study 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 depthwise 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.

To execute the convolution process, CNN contains a convolution layer with many filters.

Rectified Linear Unit (ReLU)

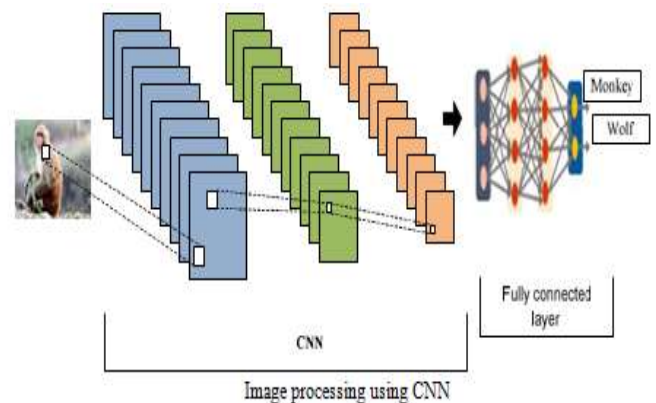
- A ReLU layer is used by CNNs to conduct operations on items. A corrected feature map is the result.

Pooling Layer

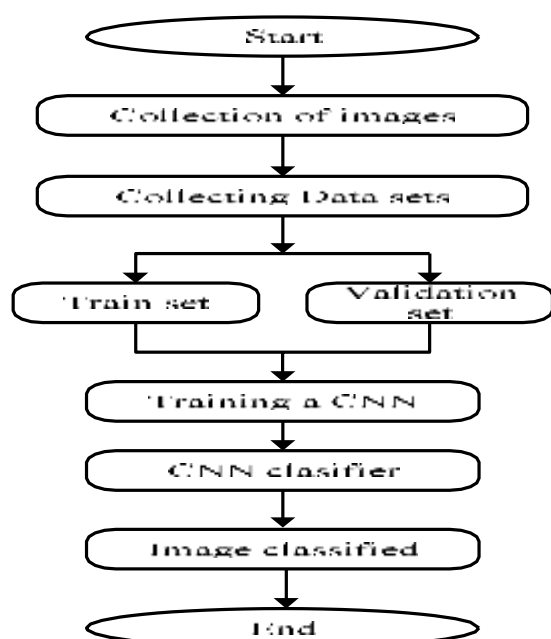
- After that, the corrected feature map goes into a pooling layer. Pooling is a downsampling procedure that decreases the feature map's dimensionality. By flattening the two-dimensional arrays from the pooled feature map, the pooling layer turns them into a single, long, continuous, linear vector.

Fully Connected Layer

- When the pooling layer's flattened matrix is supplied as an input, a fully linked layer arises, classifying and identifying the pictures.



It requires fewer parameters, memory, and calculations than traditional convolutional layers, in addition to providing greater performance. CNN, also known as ConvNets, are multilayer neural networks that are primarily utilized for visual sample processing and object detection. In 1988, Yann LeCun created the first CNN, which he dubbed LeNet. It could recognize characters such as ZIP codes and numerals. CNNs are commonly utilized to detect abnormalities, identify satellite photos, interpret medical imaging, forecast time series, and identify anomalies.



CNN algorithm flow chart

8. IMPLEMENTATION AND RESULTS

Ten wild animal photographs were obtained using a 96-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.

To avoid using a real machine in our research and instead used a virtual machine supplied by Google Cloud Platform (GCP) to train our models. The virtual machine runs Ubuntu 18.04 LTS and has 16 vCPU 2, 64GB of RAM, and a single NVIDIA R Tesla R T4 GPU. On-board memory on a T4 GPU is 16GB GDDR6, with NVIDIA Tensor Cores for quicker training and RTX hardware acceleration for faster ray tracing. T4 is a passively cooled board that relies on system airflow to stay within its operating temperature limitations. This GPU is optimized for low latency and high throughput inference, or predictions provided by deep learning models.

While training and validating a neural network, Tensor flow 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). In deep learning methodologies which were described in this paper, 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.

Confusion matrix for KNN method

Predicted Class→	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Actual Class ↓										
Elephant	910	35	0	1	1	1	0	50	1	1
Gaur	30	920	10	0	0	0	0	0	10	20
Sambar deer	0	30	870	10	0	0	30	0	30	30
Wild boar	0	20	23	900	0	0	22	0	18	17
Monkey	0	0	18	16	930	0	18	0	18	0
Porcupine	0	0	11	9	10	950	9	0	11	10
Goral	0	49	41	10	0	0	820	0	28	52
Bear	21	10	0	0	0	0	0	950	9	10
Wolf	0	10	50	10	0	0	10	0	870	50
Zebra	0	30	50	30	0	0	30	0	50	810

Precision Matrix for KNN method

Predicted Class→	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Actual Class ↓										
Elephant	0.91	0.04	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00
Gaur	0.03	0.93	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02
Sambar deer	0.00	0.03	0.87	0.01	0.00	0.00	0.03	0.00	0.03	0.03
Wild boar	0.00	0.02	0.02	0.90	0.00	0.00	0.02	0.00	0.02	0.02
Monkey	0.00	0.00	0.02	0.02	0.93	0.00	0.02	0.00	0.02	0.00
Porcupine	0.00	0.00	0.01	0.01	0.01	0.94	0.01	0.00	0.01	0.01
Goral	0.00	0.05	0.04	0.01	0.00	0.00	0.82	0.00	0.03	0.05
Bear	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.95	0.01	0.01
Wolf	0.00	0.01	0.05	0.01	0.00	0.00	0.01	0.00	0.87	0.05
Zebra	0.00	0.03	0.05	0.03	0.00	0.00	0.03	0.00	0.05	0.81

Recall Matrix for KNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	0.95	0.03	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00
Gaur	0.03	0.83	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02
Sambar deer	0.00	0.03	0.81	0.01	0.00	0.00	0.03	0.00	0.03	0.03
Wild boar	0.00	0.02	0.02	0.91	0.00	0.00	0.02	0.00	0.02	0.02
Monkey	0.00	0.00	0.02	0.02	0.99	0.00	0.02	0.00	0.02	0.00
Porcupine	0.00	0.00	0.01	0.01	0.01	1.00	0.01	0.00	0.01	0.01
Goral	0.00	0.04	0.04	0.01	0.00	0.00	0.87	0.00	0.03	0.05
Bear	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.95	0.01	0.01
Wolf	0.00	0.01	0.05	0.01	0.00	0.00	0.01	0.00	0.83	0.05
Zebra	0.00	0.03	0.05	0.03	0.00	0.00	0.03	0.00	0.05	0.81

F1-Score for KNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	0.93	0.03	-	0.00	0.00	0.00	-	0.05	0.00	0.00
Gaur	0.03	0.88	0.01	-	-	-	-	-	0.01	0.02
Sambar deer	-	0.03	0.84	0.01	-	-	0.03	-	0.03	0.03
Wild boar	-	0.02	0.02	0.91	-	-	0.02	-	0.02	0.02
Monkey	-	-	0.02	0.02	0.96	-	0.02	-	0.02	-
Porcupine	-	-	0.01	0.01	0.01	0.97	0.01	-	0.01	0.01
Goral	-	0.05	0.04	0.01	-	-	0.85	-	0.03	0.05
Bear	0.02	0.01	-	-	-	-	-	0.95	0.01	0.01
Wolf	-	0.01	0.05	0.01	-	-	0.01	-	0.85	0.05
Zebra	-	0.03	0.05	0.03	-	-	0.03	-	0.05	0.81

KNN Method is the least efficiency

Confusion matrix for CNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	945	19	0	0	0	0	0	36	0	0
Gaur	22	931	11	0	0	0	12	0	10	14
Sambar deer	0	21	912	12	0	0	18	0	13	24
Wild boar	0	21	19	902	0	0	21	0	19	18
Monkey	0	0	11	9	953	0	12	0	15	0
Porcupine	0	2	12	12	2	955	6	0	6	5
Goral	0	23	32	11	0	0	881	0	31	22
Bear	11	10	0	0	0	0	0	971	3	5
Wolf	0	11	13	12	0	0	14	0	925	25
Zebra	0	17	19	21	0	0	29	0	21	893

Precision Matrix for CNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	0.95	0.02	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00
Gaur	0.02	0.93	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01
Sambar deer	0.00	0.02	0.91	0.01	0.00	0.00	0.02	0.00	0.01	0.02
Wild boar	0.00	0.02	0.02	0.90	0.00	0.00	0.02	0.00	0.02	0.02
Monkey	0.00	0.00	0.01	0.01	0.95	0.00	0.01	0.00	0.02	0.00
Porcupine	0.00	0.00	0.01	0.01	0.00	0.96	0.01	0.00	0.01	0.01
Goral	0.00	0.02	0.03	0.01	0.00	0.00	0.88	0.00	0.03	0.02
Bear	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.01
Wolf	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.93	0.03
Zebra	0.00	0.02	0.02	0.02	0.00	0.00	0.03	0.00	0.02	0.89

Recall Matrix for CNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	0.97	0.02	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00
Gaur	0.02	0.88	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01
Sambar deer	0.00	0.02	0.89	0.01	0.00	0.00	0.02	0.00	0.01	0.02
Wild boar	0.00	0.02	0.02	0.92	0.00	0.00	0.02	0.00	0.02	0.02
Monkey	0.00	0.00	0.01	0.01	1.00	0.00	0.01	0.00	0.01	0.00
Porcupine	0.00	0.00	0.01	0.01	0.00	1.00	0.01	0.00	0.01	0.00
Goral	0.00	0.02	0.03	0.01	0.00	0.00	0.89	0.00	0.03	0.02
Bear	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00
Wolf	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.89	0.02
Zebra	0.00	0.02	0.02	0.02	0.00	0.00	0.03	0.00	0.02	0.89

F1-Score for CNN method

Predicted Class→ Actual Class ↓	Elephant	Gaur	Sambar deer	Wild boar	Monkey	Porcupine	Goral	Bear	Wolf	Zebra
Elephant	0.96	0.02	-	-	-	-	-	0.04	-	-
Gaur	0.02	0.91	0.01	-	-	-	0.01	-	0.01	0.01
Sambar deer	-	0.02	0.90	0.01	-	-	0.02	-	0.01	0.02
Wild boar	-	0.02	0.02	0.91	-	-	0.02	-	0.02	0.02
Monkey	-	-	0.01	0.01	0.97	-	0.01	-	0.01	-
Porcupine	-	0.00	0.01	0.01	0.00	0.98	0.01	-	0.01	0.00
Goral	-	0.02	0.03	0.01	-	-	0.88	-	0.03	0.02
Bear	0.01	0.01	-	-	-	-	-	0.97	0.00	0.00
Wolf	-	0.01	0.01	0.01	-	-	0.01	-	0.91	0.02
Zebra	-	0.02	0.02	0.02	-	-	0.03	-	0.02	0.89

CNN METHOD gives high efficiency

Different methods tested reports

A comparative analysis

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

All the obtained results illustrate that the 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.

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.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide Miss G.PRAMEELA for the valuable guidance continuous support and encouragement throughout the development of this project titled smart animal repelling system for crop protection using deep learning

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.



Mr.VIPPE SATYANARAYANA
RESEARCH SCHOLAR
M.Tech(CSE)
KITS-DIVILI