The Journal of Computational Science and Engineering.
ISSN: 2583-9055

UTILIZING TRANSFER LEARNING AND DEEP LEARNING METHODS FOR ANIMAL INTRUSION DETECTION

Dr S Jumlesha[1], V Bhargavi[2], G Chandana Priya[3] and Vijayan Sushma Swarna Kumari[4]

[1,2,3,4]Department of Computer Science and Engineering,

Annamacharya Institute of Technologies and Sciences, Tirupati AP India

ahmedsadhiq@gmail.com[1], bhargaviv1426@gmail.com[2], gchandanapriya85@gmail.com[3], sushmask029@gmail.com[4]

Keyword:
Image processing, CNN, VGG-16, VGG-19, MobileNet, Animal Intrusion, Deep Learning and Transfer Learning.

ABSTRACT

One of the significant dangers to diminishing the collect yield is crop harm brought about by beast assaults. Due to the expansion of developed land into natural habitats, crop attacking is one of the most aggravating conflicts between tamed and untamed life. Pests, natural disasters, and animal attacks that reduce production put India's ranchers in grave danger. Contrary to ranchers' conventional wisdom, employing gatekeepers to monitor crops and deter wild animals is nota viable option. In order to ensure the safety of bothhumans and animals, crops must be shielded from animal-caused damage and the animal must be redirected in a manner that minimizes its potential for mischief. To conquer these issues and arrive at our point, we utilize man-made consciousness (simulated intelligence) to distinguish animals as they enter our farm utilizing a division of PC vision known as a profound mind organization. The ability to identify natural organisms is emphasized in this paper. It might be hard to physically tell the species apart because there are so many different kinds. Animals are arranged by this algorithm based on their images so that we can filter them more effectively. In this paper, we will use a camera to regularly view the ranch as a whole while simultaneously taking daily photographs of the surrounding area. We are able to recognize the movement of creatures and play theappropriate sounds to scare them away with the assistance of sophisticated learning models. The different convolutional mind network libraries and ideas used to make the model are recognized in this Paper.

Corresponding Author: Email: ahmedsadhiq@gmail.com[1]

INTRODUCTION

Wild animals must now cohabit with human settlements and hunt domesticated animals due to deforestation, a lack of stable prey, and environmental devastation. As a result, animals began to hunt humans for sustenance. The elephant population in India is involved in a tragic war and is deemed responsible for the slaughter of live creatures. As a result of conflict between humans and other creatures, as well as population increase, people began destroying the wilderness for their way of life, affecting the animals and their habitats. As a result ofthe increasing industrialization of woodland regions, creatures were brought into surrounding communities. Because of asset hardship and dry season, they assureto pursue crops, trained beasts, occasionally humans, and cultivable landscape. Ranchers commonly employelectrical barriers to protect their fields from animals that may shock themselves and respond unexpectedly. It is critical to ensure the safety of both humans and animals. To address this issue, we require a precise screening system capable of detecting a photograph ofa creature part, intuitively recognizing it, and informing the public. This overview research is based on a number of wired and remote apps that shield people from being interrupted by other species. Many applications rely on research into the processing of organisms in photos [1-3]. IoT (Internet of Things) has grown into a completely new mode of communication throughout the world. It provides the capacities with a wide range of applications, some of which are already important to our general public.One of the primary cultural issues that today effects cultivating is the devastation of agricultural product by wild animals. Ranchers have long been concerned about wild animals interfering with other wild creatures. Deer, wild hoards, moles, elephants, monkeys, and other species are among the numerous that pose a threat to the reaper. These animals may consume crops, travel the field unnoticed by the rancher, and eventually destroy the crops. As a result, the yield may suffer a substantial loss, demanding extra financial safeguards to rectify the harm. When employing his innovation, every farmer should be aware that animals are present in the vicinity and must be safeguarded from pain. This problem necessitates a quick and efficient response. As a result, the purpose of our study is to address this issue. The primary goal of the paper is to keep wild animals away from agricultural areas while also protecting them by driving them away rather than killing them. In addition, the organization seeks to defend people against monster assaults. To safeguard crops from animal assaults, we are building a monitoring and deterrence system based on an integrated approach in the field of deep learning [4-5].

LITERATURE SURVEY

Salish et al. [6] both present approaches for identifying elephant encroachment in order to reduce agricultural losses. A camera trap, according to Suganthi et al., captures a photo according on the amount of active vibration sensors, and the Latest API is used to detect whether an intruder is present. Sailesh and his colleagues employ a camera linked to a Raspberry Pi device. This camera collects photographs on a regular basis and looks for elephants in them. Making loud noises or putting on bright lights, for example, can be employed as elephant deterrents in any technique. Pooja et al. [7] and Pragma et al. employ numerous inactive infrared sensors to assess if wild animals are present. In the meanwhile, Pragma and others Poojas et al. employed signal detection as part of a camera trap to identify the type of animal photographed by comparing acquired photos to a prepopulated database. Use the number of activated movement sensors to determine the size of the creature, and then its kind. Depending on the kind of animal detected, these systems are capable of a range of activities. This might include making various noises, flashing lights, or even spraying aerosol treatments to keep animals out of farmsAgadir et al. [8] present the interruption location paradigm that is most similar to the methods described in this study. When movement is detected, camera traps employ Convolutional Neural Networks (CNNs) for object identification to identify animals and take photographs. The SSD model [9] is used for object detection, and if any wild animals are detected in the image, an alert message directs the owner on what to do. All processing is conducted by a Raspberry Pi computer located near the camera trap.

The Animal Intrusion and Detection System (ANIDERS) [5] assists farmers in reducing crop loss by detecting and preventing animal incursions throughthe use of an AIR (Active Infrared) and PIR sensor combo. Alarms are activated when intruders are detected to deter them. According to the WWF-India3 pilot study, the framework correctly identified thecreature numerous times out of the 539 gets that wereinside the ANIDERS' range, implying an exactness of around 68%. This is an area where there is room for improvement. There are a few issues with each of the options considered. Most significantly, animal incursion detection systems must provide correct species identification. Different animals react differently to different stimuli; for example, a wild hog may be scared off by obvious commotions, whilstan elephant may be astonished and go on the attack [10-14].To ensure less misleading benefits, it is critical to identify the creature as well as determine itsexpectations prior to delivering messages. Furthermore, it is uncertain how frequently these gadgets may generate erroneous alarms. It's also unclear whether the frameworks can inform the client whether deterring the monster was successful after a warning sounded. If a significant number of animals were located, it would be useful to determine the animal's most recent known location as well as its direction of movement in addition to counting them. This should be done continually or very close to it. Because the solution must be cost-effective, compact computer units such as embedded systems that may be placed on-site must be employed. The structure presented in this article attempts to resolve these concerns, and an explanation of the strategy is provided in the next section [15].

PROPOSED ALGORITHM:

The invasion of private land and road crossings by wild animals has been covered multiple times in the India Times [16]. In these cases, the primary goal is to exterminate wild animals without endangering human or animal lives. In contrast to manual approaches that involve labor and hardware- based systems to detect animals that need to be repaired on a regular basis, the proposed solution just requires normal software code maintenance. The integration of the Deep Learning (DL) and Convolutional Neural Network (CNN) ideasallows for the recognition of wild animals filmed on video. Our planned work architecture is depicted in Figure 1. By playing the right repellant noises, any detected animals may be promptly scared away[17].

A. Deep Learning Approach

Because the aforementioned problem persists despite all attempts, we applied deep learning to automatically drive the animals away. In our study, we used tools like Keras and Tensorflow to perform the necessary preprocessing steps and deliver an acceptable output depending on the model's detected output. Here, the system analyses and forecasts the camera's edges, and the appropriate animal-repelling sound is produced to scare the designated beast away. The CCTV (Closed Circuit Television) system is the input source [18].

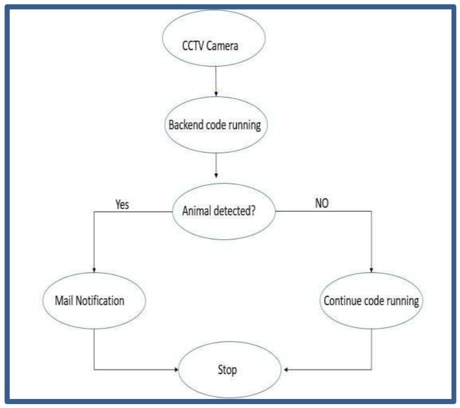


FIG. 1: PROPOSED WORK ARCHITECTURE

A. Dataset

To train our model, we used an image dataset. We collected data on 10 different animal species by hand. We evaluate 10 categories because we believe these creatures pose a threat to Indianagriculture and human life: 500 Camel, 500 Cow, 500 Rabbit, 500 Pig, 500 Horse, 500 Monkey, 500 Tiger, and 500 Elephant. The dataset is divided into two sections: testing data and training data [19].

B. Image Processing

For this case, we gathered ten distinct class images and created ten organizers with class names, saving each picture in its own envelope. To read the photos, the complete path must be provided, and the folder name must also be used to extract the class name. The pictures may be read using Opencv, and the split function can be used to retrieve the class names using "os.walkdir." When we manually gather data, we receive driftnet pixel pictures, but models won't accept different pixels. After you've read the photographs, convert them to a certain format and reshape them all. Then there are object-based class labels that must be transformed into numerical data. The numerical data was converted into categorical data using Label Encoder, and the data was divided into 70% test data and 30% train data [20].

C. CNN model

Send the data immediately to the model after finishing image processing and data splitting before building the CNN architecture. After sequentially importing the model, use the input layer to correct the form pictures and offer the activation function. Simply add a 32-filter, (5,5)-kerner size, Maxpoling, and dropouts to the input layer after that. As a result of this dropout, our model will not overfit. Two new input layers with the Relu activation function and 64 and 128 filters, 5,5kernel sizes, the same maxpooling, and dropouts were added. We only use Relu by default since Tanh or Sigmod will produce a vanishing gradient issue; we prevent this issue byusing the Relu activation function. Add dense layers to the data using simply RELU activation, then definethe output layer using softmax activation. In this output layer, we also defined the total number ofclasses. After the model construction process is completed, it is compiled. Characterise calamity and execution metric and analyzer in this assemble stage. Because we are working on a multi-class classification issue, we utilize categorical_crossentropy as a loss function and accuracy as a measure. Based on accuracy and loss, we may select the optimal model. The accuracy per epoch is acquired after fitting the model. Use the confusion metre to validate the outcome and identify which classes will be misclassified or overlooked.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	2432
conv2d_1 (Conv2D)	(None, 224, 224, 32)	25632
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18496
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 64)	0
dropout_1 (Dropout)	(None, 56, 56, 64)	0
conv2d_4 (Conv2D)	(None, 56, 56, 128)	73856
conv2d_5 (Conv2D)	(None, 56, 56, 128)	147584
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 128)	0
dropout_2 (Dropout)	(None, 28, 28, 128)	0
global_max_pooling2d (Global	(None, 128)	0
dense (Dense)	(None, 256)	33024
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570

Total params: 340,522
Trainable params: 340,522
Non-trainable params: 0

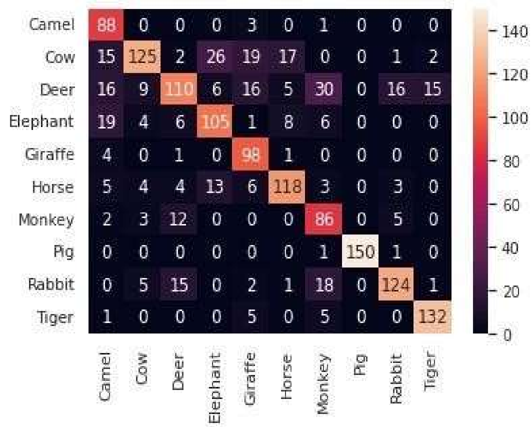
Fig2: PROPOSED CNN MODEL SUMMARY

D. Transfer Learning Models

We are aiming to advance learning forward so that we can predict accurately. In CNN, we define our architecture in terms of input layers and dense layers; however, there is no need to define anything in transfer learning because the system has already been trained on a large number of data sets; all that is required is to modify the input layer and out layer and retrain it using our own data. We experimented with MobileNet, VGG-19, and VGG-16. In this case, we merely change the input and output layers by obtaining the appropriate VGG16 model from keras.applications.vgg16. After downloading the vgg16 weight for our input shape (224,224,3), declare the output layer. Finally, organize and fit the model with precision in mind. The VGG-19 model imports, downloads, and builds weights in the same way as the VGG16 model does. Only this library is used to get the weights from the MobileNet model, which is imported from keras.applications.mobilenet. After the model has been created, fit it using categorical_crossentropy as a loss and accuracy as a measure. After training all three models, plot the accuracy, loss plot, and confusion metric for each class.

IV. RESULTS AND DISCUSSIONS

On the elephant, monkey, and boar picture datasets, convolutional neural networks and transfer learning techniques will be employed to train the proposed model. The saved model will be executed by the driver code, allowing the training photos and the fresh test images from the live capture to be compared. If one of the prepared critters is discovered during the live capture, a frightening sound is produced using speakers to drive it away. Various test photographs are given and their classes are recognized to check the model's correctness.



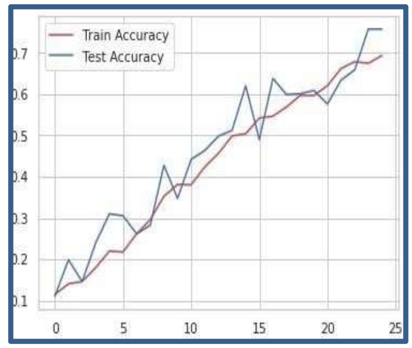
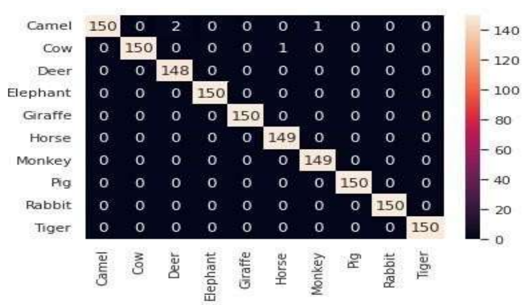


FIG. 4: CNN MODEL CONFUSION METRIC PLOT. **FIG. 5: CNN MODEL TRAIN AND TEST ACCURACY PLOT**

We may also check it from the disarray metric by using CNN, which has a 75% exactness. With this disarray framework, we can easily differentiate which class photos are appropriately characterized and which class pictures are misclassified. As seen in Figure 4, some camel images are classified as cow, deer, and elephant, while some deer images are classified as monkey and hare. These data points have all been mislabeled. Figure 5 displays the accuracy of the train and test data for each class, indicating that the number of misclassified data points must be reduced.



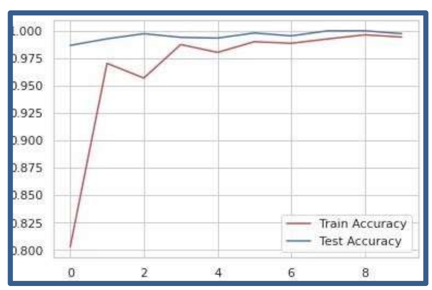
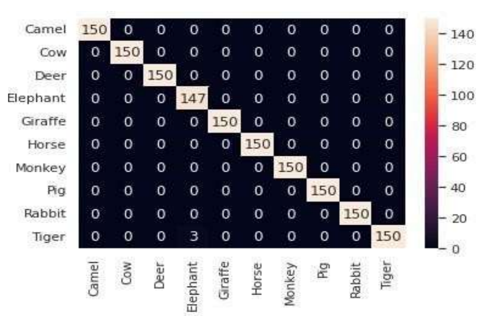


FIG. 6: VGG-16 MODEL CONFUSION METRIC PLOT **FIG. 7: VGG-16 MODEL TRAIN AND TEST ACCURACY PLOT**



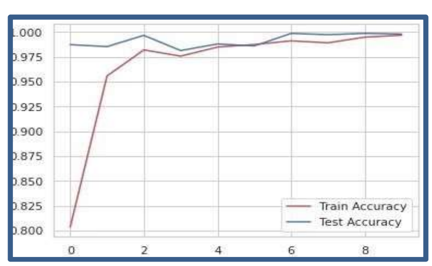


FIG. 8: VGG-19 MODEL CONFUSION METRIC PLOT **FIG. 9: VGG-19 MODEL TRAIN AND TEST ACCURACY PLOT**

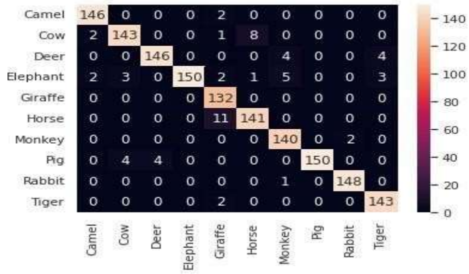


FIG. 10: MOBILENET MODEL CONFUSION MATRIX PLOT

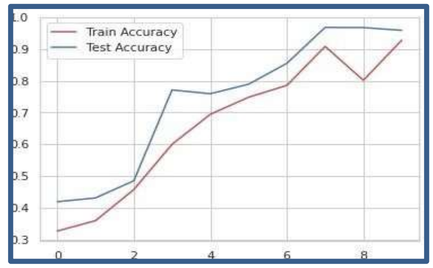


FIG.11: MOBILENET MODEL TRAIN AND TESTACCURACY PLOT

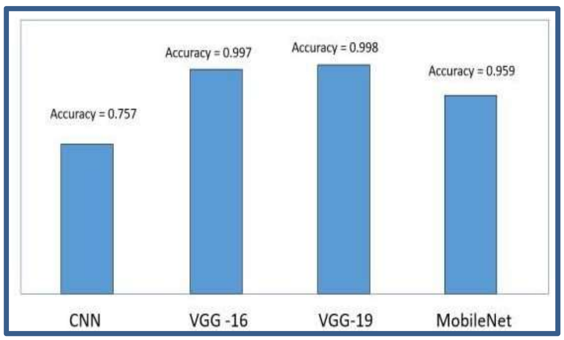


FIG. 12: ALL THE MODELS COMPARISON PLOT

We obtained 75% exactness with CNN, but this precision is insufficient because we are hoping to predict with 100 percent precision, therefore we proceed with the move learning model vgg-16. Predict 99.7% using Vgg-16, as seen in the confusion matrix in Figure 6 and the accuracy graphs for the test and training in Figure 7. Then I tested VGG-19, which delivered 99.8% accuracy like VGG-16 (see figures. 8 and 9). VGG-19 is somewhat better than VGG-16 in this one aspect. Finally, we experimented with the mobilenet concept. Figures 10 and 11 reveal that we received 94.9 percent. Finally, we anticipate 99.8% accuracy using the vgg19 model. This model will be used as the final model. During the deployment step, use opencv to retrieve photos from the camera and apply a stored model to them to forecast the class. When the model predicts an animal, a simple email to famer is all that is necessary. Figure 12 depicts the accuracy comparison of four models, with VGG-19 serving as our final model.

CONCLUSION:

Crop damage by wild animals is increasingly a big societal issue. To put it another way, every farmer should be aware that animals are living creatures that must be protected from any potential discomfort while utilized in the production of their food. A practical solution and prompt action are necessary. As a consequence, this programmed will spare farmers from making pointless attempts to preserve their fields, will aid farmers in protecting their crops, and will save farmers from huge financial losses.

REFERENCES

1. Santhiya, S., Dhamodharan, Y., Priya, N. E. K., Santhosh, C. S., & Surekha, M. (2018). A smart farmland using Raspberry Pi crop prevention and animal intrusion detection system. Int. Res. J. Eng. Technol, 5(3), 3829-3832.
2. Andavarapu, N., & Vatsavayi, V. K. (2017). Wild-animal recognition in agriculture farms using W-COHOG foragrosecurity. International Journal of Computational Intelligence Research, 13(9), 2247-2257.
3. Duhart, C., Dublon, G., Mayton, B., & Paradiso, J. (2019). Deep learning locally trained wildlife sensing in real acoustic wetland environment. In International Symposium on Signal Processing and Intelligent Recognition Systems (pp. 3-14). Springer, Singapore.
4. Jordan, M., & Jacobs, R. (1991). Hierarchies of adaptive experts. Advances in neural information processing systems, 4.
5. He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition (2015). cite. arXiv preprint arxiv:1512.03385.
6. Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737.
7. Ahmed, S. T. (2017, June). A study on multi objective optimal clustering techniques for medical datasets. In 2017 international conference on intelligent computing and control systems (ICICCS) (pp. 174-177). IEEE.
8. Sailesh, K., Balina, H. V., Sivakumar, T., & Vijaya Poojitha, P. (2019). Detection of wild elephants using image processing on raspberry pi3. 2019 International journal of Computer Science and Mobile Computing. ijcsmc, 8(2).
9. Karanth, K. K., & Kudalkar, S. (2017). History, location, and species matter: insights for human–wildlife conflict mitigation from India. Human dimensions of wildlife, 22(4), 331-346.
10. Gunashree, M., Ahmed, S. T., Sindhuja, M., Bhumika, P., Anusha, B., & Ishwarya, B. (2020). A New Approach of Multilevel Unsupervised Clustering for Detecting Replication Level in Large Image Set. Procedia Computer Science, 171, 1624-1633.
11. Sreedhar, S., Ahmed, S., Flora, P., Hemanth, L. S., Aishwarya, J., & Naik, R. (2021, January). An Improved Approach of Unstructured Text Document Classification Using Predetermined Text Model and Probability Technique. In Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020, 16-17 May 2020, Chennai, India.
12. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., ... & Ferrari, V. (2020). The open images dataset v4. International Journal of Computer Vision, 128(7), 1956-1981.
13. Korschens, M., & Denzler, J. (2019). Elpephants: A fine- grained dataset for elephant re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).
14. Li, S., Li, J., Tang, H., Qian, R., & Lin, W. (2019). ATRW: a benchmark for Amur tiger re-identification in the wild. arXiv preprint arXiv:1906.05586.
15. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
16. Liu, C., Zhang, R., & Guo, L. (2019). Part-pose guided amur tiger re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.
17. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
18. Luo, H., Gu, Y., Liao, X., Lai, S., & Jiang, W. (2019). Bag of tricks and a strong baseline for deep person reidentification. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 0-0).

19. Rangarajan, K. (2021, March). Digital Borders: Design of an Animal Intrusion Detection System Based on Deep Learning. In Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part III (Vol. 1378, p. 186). Springer Nature.
20. Divya, S., Kiran, U., & Praveen, M. (2018). Iot-based wild animal intrusion detection system. International Journal on Recent and Innovation Trends in Computing and Communication, 6(7), 06-08.