



ELEPHANT RECOGNITION AND ACOUSTIC SYSTEM FOR PREVENTION IN AGRICULTURAL FIELD

M.Bharath¹

UG student/CSE
IFETcollege of engineering,Villupuram

S.Balamurugan²

UG student/CSE
IFETcollege of engineering,Villupuram

Abstract—Presently a-days wild creatures are trespassing human livable areas and killing the living creatures frequently. Creatures, for example, Elephants harming the rural fields and enormous feline species, for example, tigers, cheetah assaults people and their dairy cattle by entering their livable places close-by thick woods. We concocted an answer for keep the harm by methods for recognizing and delivering acoustics against creatures. In this work we focus on elephant recognition and the acoustic flag produced sounds like bumble bee humming. The proposed work can diminish harms in farming field and furthermore human life.

Keywords— Motion detection, image classification, machine learning.

1. INTRODUCTION

Although technology is well developed there are fewer implementations to provide security for both animals and humans over forest borders.

In order to secure the life of both animals and humans, camera-equipped devices should be mount over vulnerable and agricultural lands near hills where animals often trespass.

We are using image processing library to capture or activate camera whenever motion is detected. For every image, our machine learning library performs a huge image classification process and detects the type of animal which has been caught in the camera. For every animal, it produces some appropriate acoustic sounds that disturb the particular animal and stop its progress

2. OVERVIEW

We've implemented 3 separate python programs for Raspberry pi implementation. We too built android application

to detect the type of animals and distracts them with sound mechanism and alerts forest department by sending SMS. Separate python programs with three modules for raspberry pi or desktop 7 webcam implementation is listed below

- motion_detect.py - To capture images whenever the motion is detected using OpenCV (Image processing library)
- classify_image.py - To analyse the captured image & identify the type of species found using TensorFlow (machine learning library)
- classify_text.py - To analyse output & triggers the acoustics mechanism

Our Image processing & machine learning methodology to detect the elephants is working flawlessly based on the manual mechanism that we've implemented. Right now, we're lagging in the enhanced automation. To automate our manual working prototype to be implemented in raspberry pi we are creating shell scripts to execute based on logical algorithmic approaches. Yet we've achieved automation with our android application(APK). Our APK detects elephant & produces alert sound to keep elephants away from crops, thereby agricultural fields can be saved form damage. It can also be used to detect wild animals which may cause harm to public and their pets. Thus, prevents the damage before they caught up in action by means of acoustics.

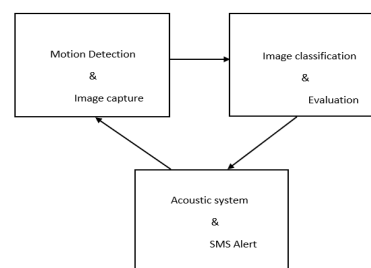


Fig 1: Working mechanism - MVC model



3. WORKING MODULES

A similar arrangement can be accomplished with the assistance of embedded frameworks by mounting sensors and other equipment gadgets. Be that as it may, it is more cost of usage than our product based execution of camera acknowledgment by our picture preparing and machine learning libraries.

TENSORFLOW

TensorFlow is an open-source programming library for dataflow programming crosswise over extensive variety of assignments. It is an emblematic math library and likewise utilized for machine learning applications, for example, neural systems. It is utilized for both research and creation at Google, regularly supplanting its shut source ancestor, DistBelief. It was created by the Google Mind group for inner Google utilize. It was discharged under the Apache 2.0 open source permit. For this task it is utilized to recognize pictures by looking over camera.

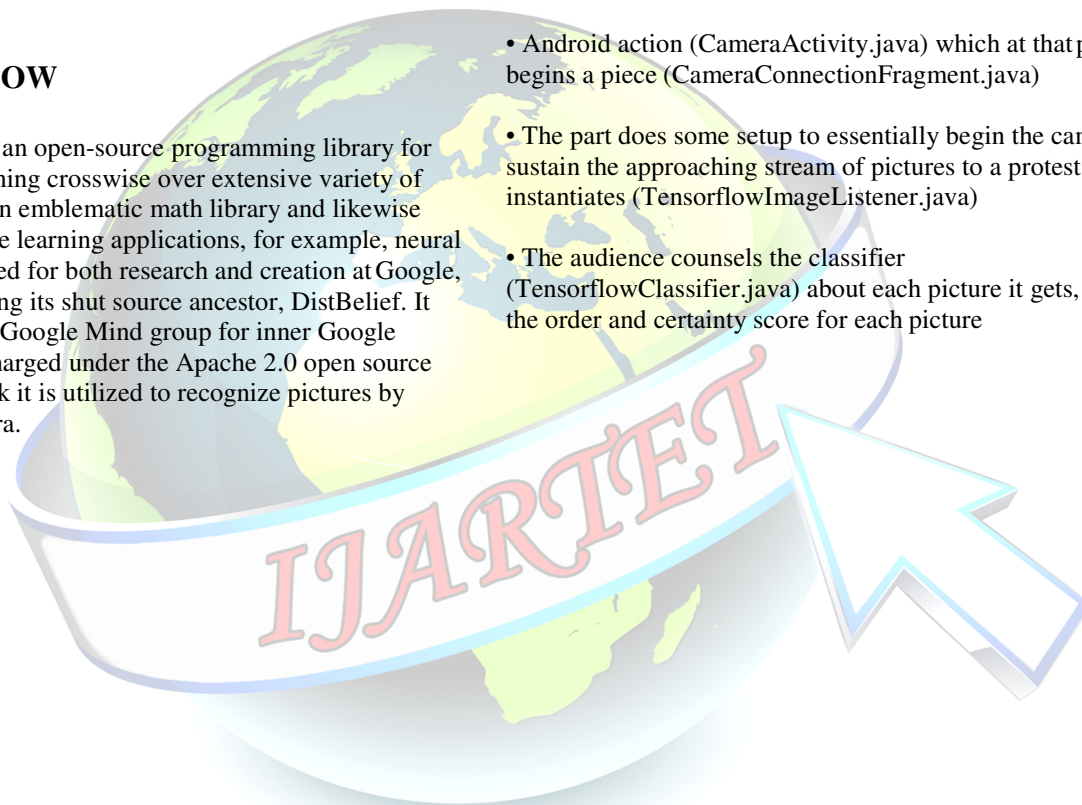
Android Implementation

By utilizing TensorFlow Programming interface for Android, portable application has been made which performs constant acknowledgment of creatures This procedure is abridged in Fig 2, where the checked picture hits the pre - prepared database display.

The application gets to the camera which underpins JNI (Java Local Interface) and associates with TensorFlow's core engine.

On startup, the application dispatches as takes after:

- Android action (CameraActivity.java) which at that point begins a piece (CameraConnectionFragment.java)
- The part does some setup to essentially begin the camera and sustain the approaching stream of pictures to a protest it instantiates (TensorflowImageListener.java)
- The audience counsels the classifier (TensorflowClassifier.java) about each picture it gets, and gets the order and certainty score for each picture



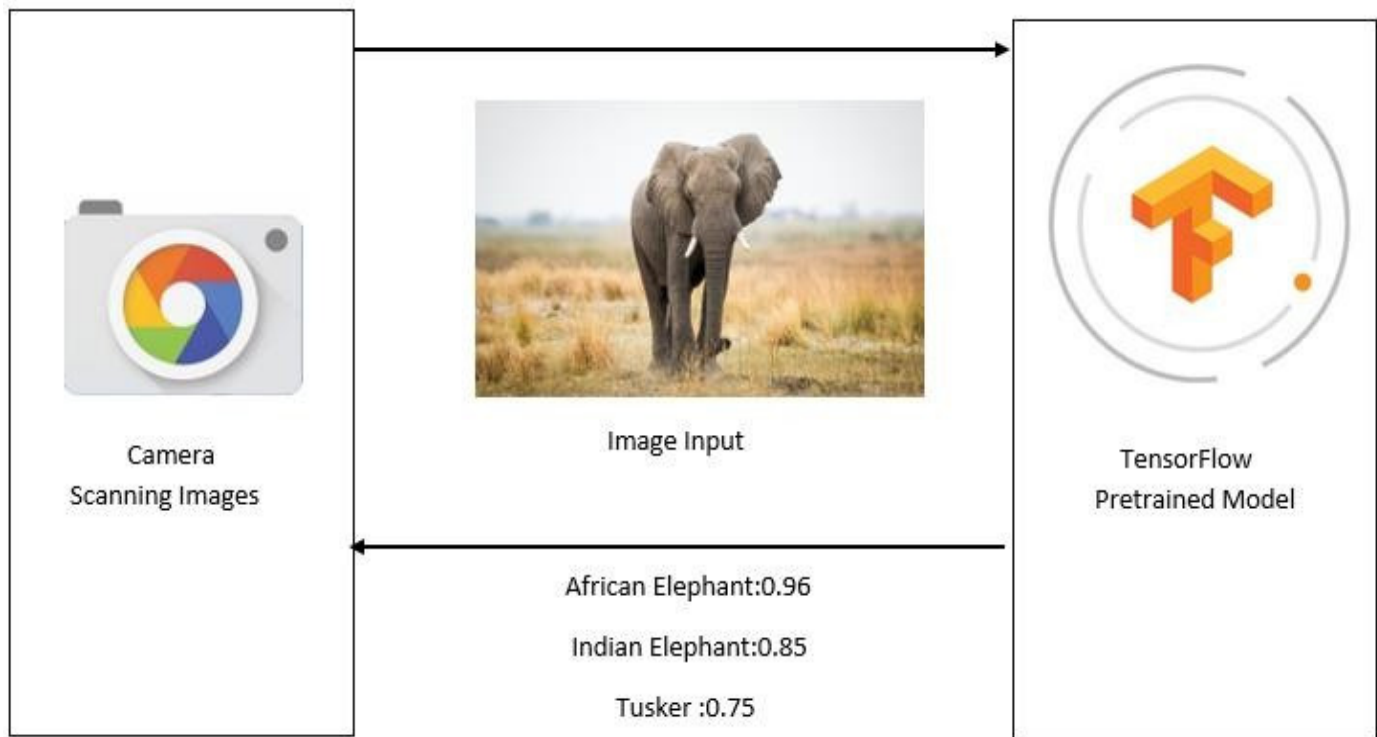
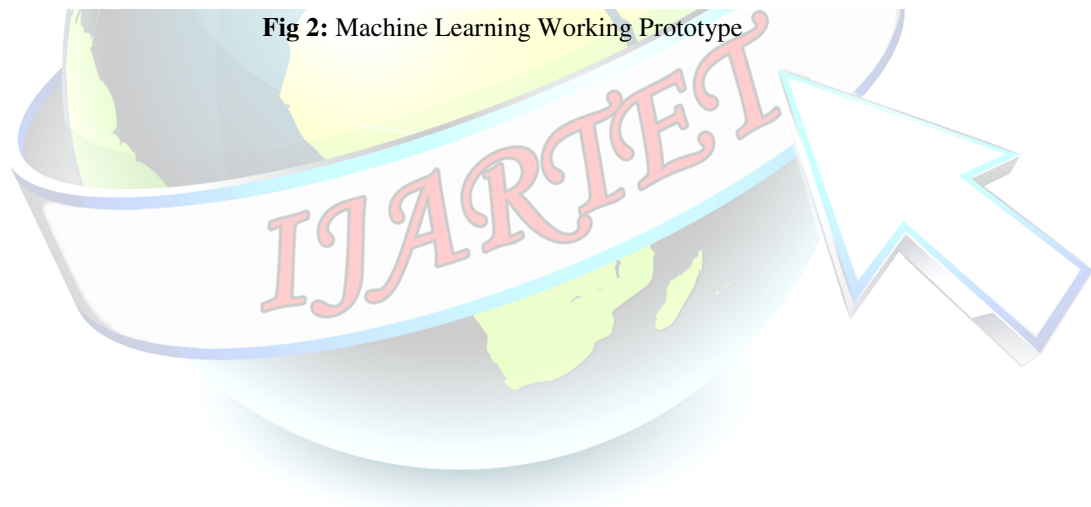


Fig 2: Machine Learning Working Prototype



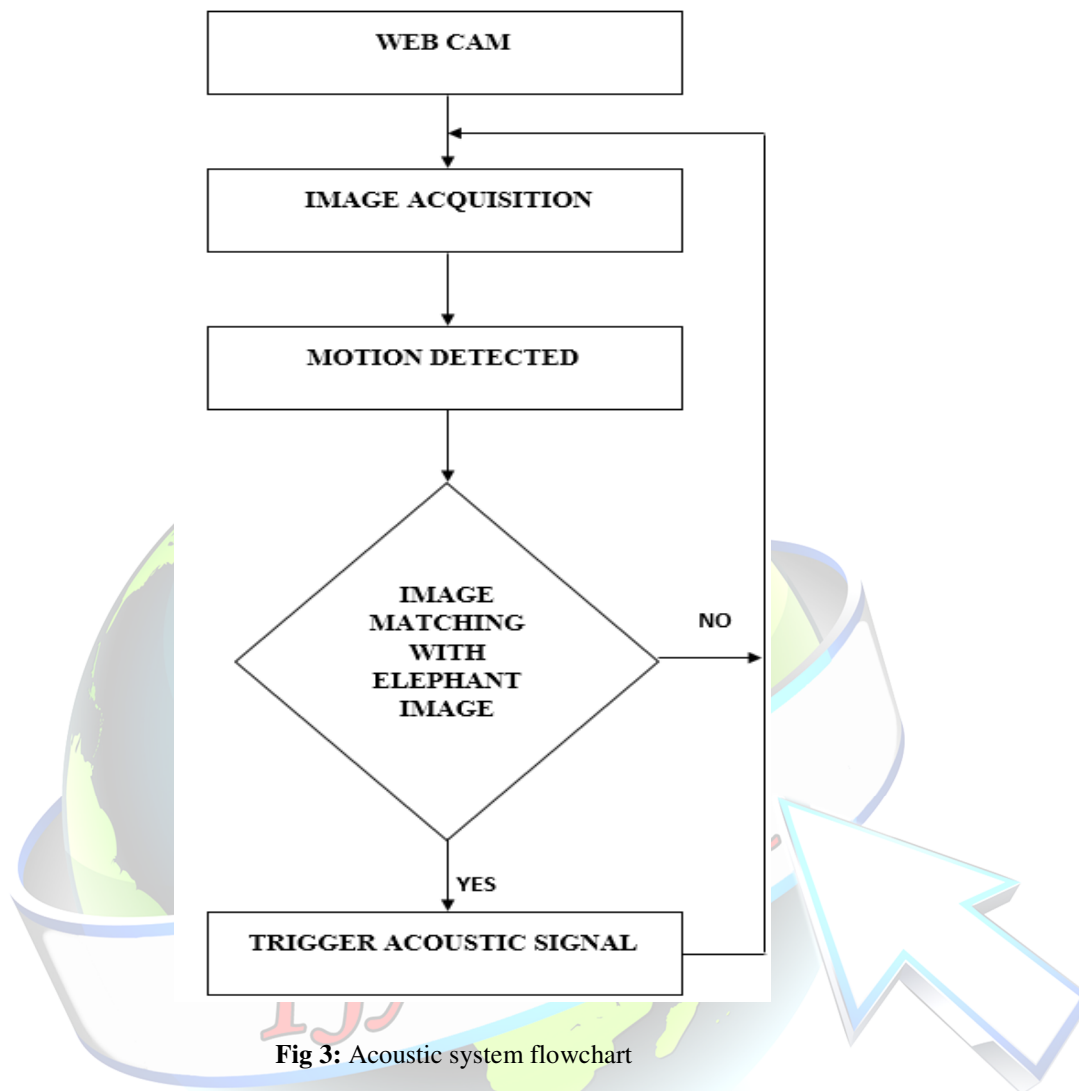


Fig 3: Acoustic system flowchart

Web Cam Implementation OpenCV

OpenCV (Open Source PC Vision) is a library of programming capacities for the most part went for constant PC vision. Initially created by Intel, it was later upheld by Willow Carport and is currently kept up by Itseez. The library is cross-stage and free for use under the open-source BSD permit. It bolsters the profound learning systems TensorFlow, Light/PyTorch and Caffe.

The library has in excess of 2500 advanced calculations, which incorporates a far reaching set of both exemplary and cutting edge PC vision and machine learning calculations.

Motion Detection

OpenCV library is used to activate camera whenever the motion is detected. In Webcam based implementation the camera module captures the images and send it for image classification. Then the TensorFlow python API triggers the classification and at once the image hits the pretrained model of elephants it activates the acoustic system. Based on the threshold value the terminal outputs are evaluated within the terminal and executes the operations. Automate.sh – Linux shell script helps to automate the task of motion detection, image classification and acoustic triggering operation at once the animal has been found as shown in Fig 4



```
siveshversion@Inspiron:~/Documents$ python motiondetect.py
siveshversion@Inspiron:~/Documents$ python classify_image.py
>> Downloading inception-2015-12-05.tgz 100.0%
Successfully downloaded inception-2015-12-05.tgz 88931400 bytes.
2018-03-08 14:08:19.430164: I tensorflow/core/platform/cpu_feature_guard.cc:137] You
mpiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
2018-03-08 14:08:19.904189: W tensorflow/core/framework/op_def_util.cc:343] Op Batc
n GraphDef version 9. Use tf.nn.batch_normalization().
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.89107)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00779)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.0029
custard apple (score = 0.00147)
earthstar (score = 0.00117)
siveshversion@Inspiron:~/Documents$ python classify_image.py --image_file "1.jpg"
2018-03-08 14:09:32.974003: I tensorflow/core/platform/cpu_feature_guard.cc:137] You
mpiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
2018-03-08 14:09:33.288418: W tensorflow/core/framework/op_def_util.cc:343] Op Batc
n GraphDef version 9. Use tf.nn.batch_normalization().
African elephant, Loxodonta africana (score = 0.68901)
tusker (score = 0.26093)
Indian elephant, Elephas maximus (score = 0.00546)
ice cream, icecream (score = 0.00021)
coucal (score = 0.00019)
siveshversion@Inspiron:~/Documents$ python classify_text.py
african elephant
elephant found
^Z
[1]+  Stopped                  python classify_text.py
```

Fig 4: Image Processing & Machine Learning Webcam Prototype

SOFTWARE REQUIREMENTS

- Raspbian OS
- Python
- TensorFlow
- OpenCV
- Numpy
- SD formatter
- Android Studio
- Ubuntu OS

5. CONCLUSION

Along these lines, machine learning and picture handling is the immense combo to give solutions for ensuring rural fields and guaranteeing security to regular man and untamed life. In our future work we will actualize a similar instrument utilizing IOT packs, for example, Raspberry pie to accomplish improved robotization with a cautioning framework to edges and timberland office.

6. REFERENCE

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] M. Abadi, A. Agarwal, P. Barham, Brevdo et al., "Tensorflow: Largescale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [3] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.
- [4] S. Citrigno, S. Graziano, and D. Sacca, "Cooperation of smart objects and urban operators for smart city applications," in *Management of Cyber Physical Objects in the Future Internet of Things*. Springer, 2016, pp. 157–174.
- [5] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.
- [6] M. K. Schultz, "A case study on the appropriateness of using quick response (qr) codes in libraries and museums," *Library & Information Science Research*, vol. 35, no. 3, pp. 207–215, 2013.



[7] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA, 2016.



