Real-Time Mosquito Species Identification using Deep Learning Techniques

Pratik Mulchandani, Muhammad Umair Siddiqui, Pratik Kanani

Abstract: According to the World Health Organization, diseases such as malaria and dengue account for almost one million deaths every year. Carrier mosquitoes for a particular disease remain exclusive to it. A majority of carrier mosquitoes spread the disease throughout a region by reproducing in it. With advancements in Machine Learning and Computer Vision technologies, the species of mosquitoes in a particular region can be easily and swiftly detected using recordings of their wing movements. The wingbeats of a particular mosquito species are unique, making this a reliable method to identify them. Once these solutions are deployed on mosquito traps, a particular region can be alerted if, for example, an Aedes Aegypti mosquito is found. This mosquito species is widely known to carry the Zika virus. The identification of such carrier species can also help in detecting the spread of mosquito-borne diseases in the surveyed region. In this paper, we go through various techniques that show promising results in the identification of mosquito species. The trained models can be deployed on constrained devices to make a cost-effective and efficient mosquito species identification system.

Keywords: carrier mosquitoes, constrained devices, machine learning, mosquito detection, deep learning

I. INTRODUCTION

The spread of diseases such as dengue and yellow fever has resulted in studies where the source of the Zika and West-Nile viruses are identified. Such viruses are spread through a particular species of mosquitoes only, which renders the other species harmless. The Zika virus, for example, is spread by the Aedes species of mosquitoes, typically Ae. aegypti and Ae. albopictus. While researchers have looked into ways to genetically modify certain mosquito species to inhibit them from breeding, many have qualms regarding the consequences of such actions. Identifying whether a carrier mosquito species exists in a particular neighborhood seems to be the first step to a successful disease prevention program. Once a carrier mosquito is found in the area, it can be assumed that there are others, given the speed with which mosquitoes breed. Necessary steps such as fumigation can be carried out later.

Pratik Mulchandani, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, India. pratikm1910@gmail.com

Muhammad Umair Siddiqui, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, India.

Pratik Kanani, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, India.

The identification of the species can be carried out using devices capable of recording mosquito wing movement patterns. The wing beats are unique to each sub-species and can be used to identify the genus of the mosquito. The three target genera are Aedes, Anopheles and Culex. Mosquitoes of these genera are potential vectors for the disease-causing pathogen. Recordings of the wingbeats are then passed on from the sensor to the chosen Deep Learning model, which helps determine its species. Thus, the count of these mosquitoes can be maintained, and the risk of contracting diseases such as dengue, malaria, etc. can be inferred. Programming, especially Deep Learning is able to help us to build different models in health care domain [14-15].

II. DATASET

A. Description

The dataset used in this paper is the WINGBEATS dataset. It consists of 279,566 mosquito wingbeat recordings obtained from 6 mosquito species i.e. Ae. aegypti, Ae. albopictus, An. arabiensis, An. gambiae, Cu. pipiens and Cu. quinquefasciatus which cover 3 genera, namely, Aedes, Anopheles and Culex. The recordings are wav files generated by a Large Aperture Optoelectronic sensor [1] which was exposed to six boxes, each containing mosquitoes of a particular species. As the mosquitoes fly across the sensor, their wingbeats partially restrict light from reaching the sensor, causing a light fluctuation to be recorded. This gets saved as a wav file in accordance with the rhythm of the wingbeat motion.

B. Data Preprocessing

The wav files were processed to obtain unsigned PCM 16kHz signals. The audio singles are one-dimensional signals. Since most of the Neural Networks have been exceedingly successful in applications such as Image Recognition and Classification, we converted the audio files into spectrograms (Fig. 1). The various frequencies from the audio file were treated as a vector of numbers, which were then arranged in time order. This forms a two-dimensional array, which is treated as a single-channel image, i.e. a spectrogram. No transformations such as mirroring or rotating the images were made, as these would change the meaning of the spectrograms. The dataset was split into training, validation and testing data in an 80:10:10 ratio. Images from each class were randomly mixed in the training, validation and testing categories.

Real-Time Mosquito Species Identification using Deep Learning Techniques
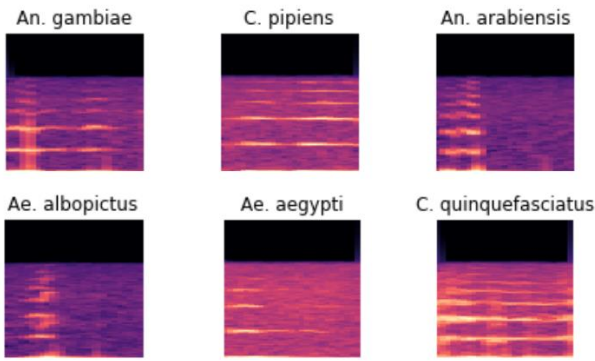

Fig 1. Spectrograms of one audio file from each class

III.   METHODS

A.   Multi-Layer CNN

Convolutional Neural Networks are most commonly used for image classification. The three types of layers in convolutional networks are the Convolutional Layer, the Pooling layer and the Fully Connected layer. The Convolutional layers are used to reduce the input dimensions and filters, which are made up of kernels, are used to extract important features from the images. The Pooling layer is used to extract maximum of average of a region and also reduce dimensionality. In the model used in this paper, Max Pooling is used. The Fully Connected Layer is used to aggregate information from final feature maps and to generate final classification using a Neural Network. A multi-layer convolutional neural network having similar architecture (Fig. 2) to the cnn-trad-fpool3 [2] was trained for 80,000 steps, each having a batch size of 150, on the spectrograms to obtain a maxi mum testing accuracy of 86%. This model uses 940,000 weight parameters and takes about 800 million FLOPs per inference, and thus cannot be used on devices with very limited hardware capabilities. Using TensorFlow, the frozen graph was converted to tflite format. This file was quantized [3] and then deployed on an Android device using a lower-end Snapdragon 410 SOC to simulate potential sensor hardware. The final inference time recorded was 349ms.
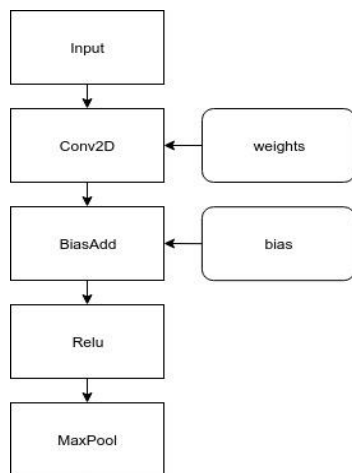

Fig 2. Layout of one layer of the cnn-trad-fpool3 [2] model

B.   ResNet

The residual neural network, ResNet, [4] is a model that uses residual blocks (Fig. 3) in order to preserve the identity function of previous elements. This architecture helps make sequence models which can preserve the context of previous layers. Each residual block has two 3x3 layers, each followed by a normalization layer and a ReLU activation function. Due to this, the outputs are of the same shape as the inputs. ResNet models have been used extensively for applications such as image detection and classification, due to their similarity to VCG models. ResNet-34 (2-layer deep) and ResNet-50 (3-layer deep) models were trained on the data with a batch size of 12 and randomized learning rate and achieved maximum accuracy of 80.7% and 81.03% respectively.
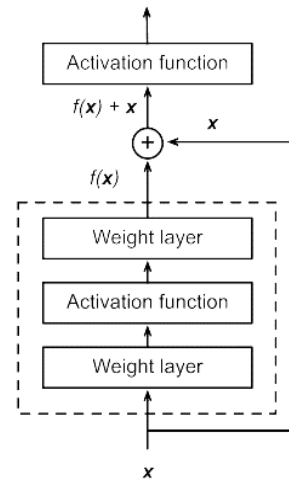

Fig 3. Schematic representation of a residual block

C.   DenseNet

While ResNet uses the context of just the previous layer at each stage, DenseNet [5] takes it a step further by using the concatenated knowledge of all the previous layers in the network. Instead of performing the addition of multiple layers, dense models use layer stacking. Since each layer contains the knowledge of all the previous layers, the network has fewer channels and is more compact. DenseNets require much fewer input parameters than the previous models. While ResNets may contain layers with redundant or less information, DenseNet layers are very narrow. This leads to greater efficiency and smaller model size than ResNet.The architecture of DenseNet is shown in Fig. 4. The final layer is a softmax function. Training a 121-layer deep DenseNet model on our data with a batch size of 16 led to a maximum accuracy of 80%. The loss in training and validation is shown in Fig. 5.
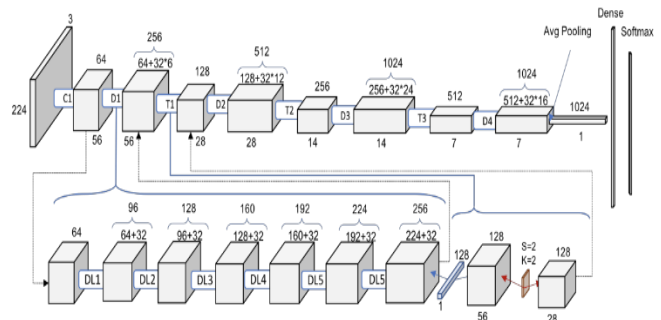

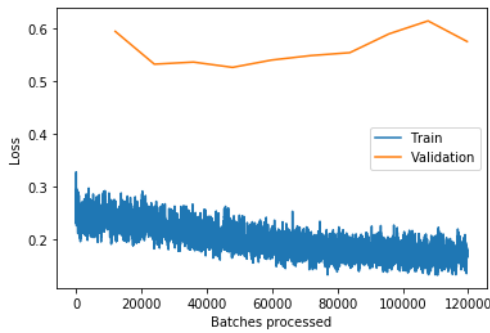**Fig 4. Architecture of DenseNet [5]**

Fig 5. Plot of loss per batch during DenseNet-121 training

D. XGBoost

Extreme Gradient Boosting [6] is a gradient boosted ensemble machine learning algorithm based on Decision Trees. Each record in the dataset is assigned a weight which determines the probability of that record to be selected for creating a particular decision tree. The decision trees are created sequentially based on the updated weights for the records. A model created for a particular decision tree is tested with all the training records and the weights of the incorrectly classified records are updated. Now the records with updated weights will be considered for the creation of the next decision tree. The final classifier is a combination of all the decision trees (Fig. 6). The Learning rate used for this algorithm was 0.2 and the number of trees was set to 650. The accuracy obtained for these parameters was 85.8% (Fig. 7). Unlike the other Neural Network-based Deep Learning methods, XGBoost algorithm was trained on the audio files directly instead of converting them into spectrograms first. This may have helped XGBoost achieve higher accuracy figures than the other models, as there is a minor loss of data when converting audio files to spectrograms
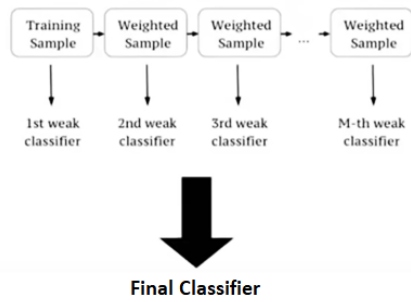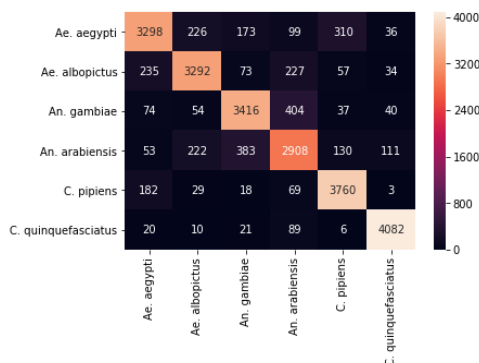

Fig 6. XGBoost Process

Fig 7. Final Confusion Matrix obtained after training XGBoost.

IV. RESULTS

The results obtained on the test set are as shown in Table 1. While it can be inferred that the Multi-Layer CNN performs the best having a maximum accuracy of 86%, XGBoost is second and manages to achieve 85.8% accuracy. The fact that XGBoost outperforms the other CNN-based models shows that audio classification using raw audio files may be more accurate than the spectrogram-based approach. This is due to the fact that a CNN developed for images uses two-dimensional filters across the x and y dimensions. The x and y axes in a spectrogram denote time and strength of the frequency respectively. Thus, any small shifts in either quantity can cause a big difference in sound. This may cause errors during classification.

Table 1. Comparison of accuracy obtained.

Model	Maximum Accuracy	Type of Input Data
ResNet-34	80.7%	Spectrogram
ResNet-50	81.03%	Spectrogram
DenseNet-121	80%	Spectrogram
Multi-Layer CNN	86%	Spectrogram
XGBoost	85.8%	Audio

V. CONCLUSION

The quantized frozen graph of the Multi-Layer CNN when deployed on an Android device shows promising results in the detection of mosquito species. The proposed solution depicted in Fig. 8 can be deployed as a cost-effective means of determining whether the user's area has any threat of mosquito-borne diseases. Such a system can be deployed in urban areas to track the existence of carrier mosquito species. This would help identifying the spread of such mosquito-borne diseases in non-native regions. Areas with higher densities of carrier mosquitoes can be identified as breeding grounds. Consequently, such findings can be used by local bodies to ensure sufficient insecticide use to combat these diseases.
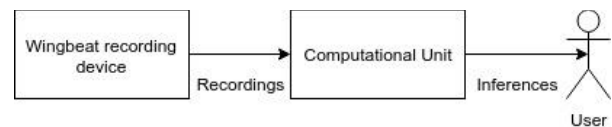

Fig 8. Schema of the mosquito species recognition system

REFERENCES

1. Potamitis, I., Rigakis, I., & Tatlas, N.-A. On Fresnel-Based Single and Multi Spectral Sensors for Insects' Wingbeat Recording. 2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII). doi: 10.1109/transducers.2019.8808662, 2019.
2. Arık, S. Ö., Kliegl, M., Child, R., Hestness, J., Gibiansky, A., Fougner, C., Prenger, R., Coates, A. Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting. Interspeech 2017. doi: 10.21437/interspeech.2017-1737, 2017.
3. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi: 10.1109/cvpr.2018.00286, 2018

4. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2016.90, 2016.

5. Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2017.243, 2017.

6. Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2017.243, 2017.

7. Chen, T., & Guestrin, C. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16. doi: 10.1145/2939672.2939785, 2016.

8. Chen, Y., Why, A., Batista, G., Mafra-Neto, A., & Keogh, E. Flying Insect Detection and Classification with Inexpensive Sensors. Journal of Visualized Experiments, (92). doi: 10.3791/52111, 2014.

9. Nishihara, R., Stoica, I., Moritz, P., Wang, S., Tumanov, A., Paul, W., Schleier-Smith, J., Liaw, R., Niknami, M., Jordan, M. I.. Real-Time Machine Learning. Proceedings of the 16th Workshop on Hot Topics in Operating Systems - HotOS 17. doi: 10.1145/3102980.3102998, 2017.

10. Okawa, M., Saito, T., Sawada, N., & Nishizaki, H. Audio Classification of Bit-Representation Waveform. Interspeech 2019. doi: 10.21437/interspeech.2019-1855, 2019.

11. Potamitis, I., Rigakis, I., Vidakis, N., Petousis, M., & Weber, M. Affordable Bimodal Optical Sensors to Spread the Use of Automated Insect Monitoring. Journal of Sensors, 2018, 1–25. doi: 10.1155/2018/3949415, 2018.

12. Santos, D. A., Rodrigues, J. J., Furtado, V., Saleem, K., & Korotaev, V. Automated electronic approaches for detecting disease vectors mosquitoes through the wing-beat frequency. Journal of Cleaner Production, 217, 767–775. doi: 10.1016/j.jclepro.2019.01.187, 2019.

13. Sultana, F., Sufian, A., & Dutta, P. Advancements in Image Classification using Convolutional Neural Network. 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). doi: 10.1109/icrcicn.2018.8718718, 2018.

14. Kanani, P. & Padole, M. (2018) Recognizing Real Time ECG Anomalies Using Arduino, AD8232 and Java. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T. (eds) Advances in Computing and Data Sciences. ICACDS 2018. Communications in Computer and Information Science, vol 905. Springer, Singapore

15. Kanani, P. & Padole, M., "Deep Learning to detect skin Cancer using Google Colab", International Journal of Engineering and Advanced Technology (IJEAT),Volume-8 Issue-6, August 2019.

AUTHORS PROFILE

Pratik Mulchandani, He is currently a Computer Engineering student at Dwarkadas J. Sanghvi College of Engineering, Mumbai. He has interests in the fields of Machine Learning, Deep Learning, Natural Language Processing, and Computer Vision. He has gained experience through a research internship where he worked on interactive robots and will be pursuing his masters in the future.

Muhammad Umair Siddiqui, He is currently a Computer Engineering student at Dwarkadas J. Sanghvi College of Engineering, Mumbai. He has interests in the fields of Machine Learning, Deep Learning and Computer Vision. He has worked on several projects and will soon begin working as an Analyst in COE Assurance at Ernst & Young.

Pratik Kanani, Pratik received his Bachelors's and Master's Degree from the University of Mumbai. He has more than 28 research contributions in International and National Journals as well as Conferences. He is an active researcher in the field of Network Security, IoT, Machine Learning and Fog Computing.