A Deep Learning Framework to Automatically Identify Mosquito Species using Smartphone Images

Mona Minakshi, Pratool Bharti, Willie B. McClinton III, Jamshidbek Mirzakhalov, Ryan M. Carney and Sriram Chellappan

Abstract—Among all animals, mosquitoes are responsible for the most deaths worldwide. However, not all of them are deadly, and detecting the small percentage of disease-transmitting mosquito species is hence critical to surveillance and control. In this paper, we design a deep learning framework for the automatic identification of mosquito genus and species by processing mosquito images taken via smartphone cameras. The training dataset was generated from 250 mosquito specimens, primarily trapped in the wild in Florida. These mosquitoes were evenly distributed among *nine* species (*three* species each across *three* genera). Among these, our dataset includes *Aedes aegypti* and *Anopheles stephensi*, which are critical vectors for many diseases such as dengue and malaria, respectively. Each mosquito specimen was imaged using multiple smartphones, and on various backgrounds. The resulting dataset generated (and appropriately augmented) yielded a total of 25867 mosquito images for training and validation. Our deep learning framework for classification is a hybrid of Inception Net and ResNet architectures, and leverages the principle of Transfer Learning. This design ensures faster training and processing, and is easier to optimize. Using this framework, three distinct learning models were developed - a) to identify genus alone; b) to identify species based on genus identification; and c) to identify species directly. Performance evaluations results demonstrate high practical value for our models, with accuracies of around 85% in identification. Also presented are important discussions on the contextual relevance and potential impact of our models, results, and scope of future work in the realms of public health and mosquito biology.

Index Terms—Mosquitoes, Smartphones, Citizen Science, Epidemiology, Machine Learning, Computer Vision, Deep Learning, Public Health

1 Introduction

Mosquito-borne diseases like malaria, dengue, West Nile virus and Zika are significant public health concerns, with major human and economic cost. For instance, malaria alone is responsible for more than 1 million deaths per year worldwide, with most of them being children. There are generally no vaccines or cures available for these diseases, and thus prevention relies upon mosquito surveillance and control. This in-turn requires accurate and real-time knowledge on the geographic presence and abundance of the relevant mosquitoes. This is because, while there are close to 4500 species of mosquitoes (spread across 34 genera) [17] only a select few are competent vectors to spread diseases. These mosquito vectors primarily belong to three genera - *Aedes* (*Ae.*), *Anopheles* (*An.*) and *Culex* (*Cu.*). Within these genera, there are multiple species responsible for transmitting particular pathogens. Malaria is spread primarily by *An. gambiae* in Africa and by *An. stephensi* in India. Dengue, yellow fever, chikungunya, and Zika viruses are spread primarily by the species *Ae. aegypti*. *Cu. nigripalpus* is a vector for West Nile and other encephalitis viruses. Certain species within all these three genera can also transmit parasitic worms that cause elephantiasis.

Interestingly, in areas where mosquito-borne diseases are prevalent, the vectors co-exist with other mosquito types that are not competent enough to spread diseases. As such, identifying which (i.e., genus and species) types of mosquitoes are present and in what densities become critical in controlling vector populations. To do so, in advanced economies like the USA (e.g., in Florida), dedicated personnel employed for mosquito control lay scores of traps to capture specimens, and identify them one by one under a microscope to record its genus and species, and log population densities. This manual process is very taxing since hundreds of mosquitoes can get trapped each time.

In this paper, we design a deep learning framework to automate this process, wherein anyone (a citizen or expert) could take an image of a mosquito specimen that is either alive (but still), or dead (possibly via spraying or trapping, but still physically undamaged) to automatically identify its genus and species. To generate the dataset for this challenging problem, we partnered with the Hillsborough County mosquito control district in Florida to lay traps and capture mosquito specimens (details presented later in Section 3). Subsequently, the experts there helped us identify the genus and species of 250 trapped mosquitoes. We then captured multiple images of

Mona Minakshi is a PhD student in the Computer Science & Engineering Department at University of South Florida, Tampa, FL 33620 USA. (email: mona6@mail.usf.edu).
Pratool Bharti is with the Computer Science Department at Northern Illinois University, Dekalb, IL 60115 USA. (email: pbharti@niu.edu).
Willie B. McClinton III is an undergraduate student in the Computer Science & Engineering Department at University of South Florida, Tampa, FL 33620 USA. (email: wmcclinton@mail.usf.edu).
Jamshidbek Mirzakhalov is an undergraduate student in the Computer Science & Engineering Department at University of South Florida, Tampa, FL 33620 USA. (email: mirzakhalov@mail.usf.edu).
Ryan M. Carney is with the Integrative Biology Department at University of South Florida, Tampa, FL 33620 USA. (email: ryancarney@usf.edu).
Sriram Chellappan is with the Computer Science & Engineering Department at University of South Florida, Tampa, FL 33620 USA. (email: sriramc@usf.edu).

each mosquito specimen from ten smartphone cameras, across different orientations, and on top of multiple backgrounds under normal indoor light conditions. With these, we generated a set of 6807 mosquito images. Then, we augment these images via zooming and adjusting brightness/ contrast (to compensate for variance at run-time imaging) to yield a dataset of 25867 images) spread across three genus, and three species each per genus for model development. Our dataset is presented in Table 1, along with the diseases spread by these mosquitoes and their geographical range.

Our deep learning framework to classify mosquitoes from smartphone images is a hybrid of Inception Net and ResNet architectures. Briefly, these architectures are state-of-the-art in image classification, and lend themselves to easier optimization. Furthermore, we speed up training time using the principle of Transfer Learning. Leveraging from this general framework, we design separate models for three classes of problems: a) a model to identify the genus alone; b) models to classify the species based on known genus identification; and c) a model to directly identify the species. Our performance evaluations reveal very interesting results of significant practical use. Subsequently, we contextualize all our results, present details of implementing our models as a smartphone app, and also highlight the practical significance and impact of our contributions in public health and mosquito biology.

2 Related Work

In the context of combating mosquito-borne diseases, corporations like Microsoft and agencies like NASA, Global Mosquito Alert Consortium and the U.S. military are investing in drone and satellite based technologies to identify mosquito habitats. Organizations like iNaturalist are also encouraging citizens across the world to generate and upload images of animals and insects such as mosquitoes they encounter in nature for large scale data collection. Nevertheless, of interest to this paper is related work on AI technologies for mosquito identification, which we present below.

In [10], a Support Vector Machine based algorithm was designed and evaluated to detect *Ae. aegypti* mosquitoes using a dataset of 40 images taken via a 500x optical zoom camera. In other related work [13, 14], learning algorithms are designed to detect mosquitoes from other insects like bees and flies, where the images are captured once again using sophisticated digital cameras. In another 2016 paper [33], the authors imaged wings, and more specifically looked at coordinates at intersections of wing veins as a feature to classify genus of mosquitoes. However, only 12 mosquitoes in total were considered for model development, and also, the requirement of imaging wings alone is cumbersome.

In parallel, researchers are looking at acoustic markers to classify mosquitoes. Work here include [8, 23], wherein the authors use "optical" sensors, and smartphone microphones to record sound of mosquito flight, and then use features derived from sound for classification of mosquitoes. But these techniques, while certainly innovative, cannot be used for larger scale surveillance, or by common citizens, since the sensing devices must be close to mosquitoes in flight. Recently, researchers are also exploring ways to identify *larvae* of mosquitoes from citizen generated imagery [25], a problem that is related, but orthogonal to our work in this paper.

We have done preliminary work in identifying mosquitoes from smartphone images [23]. Specifically, in [23], we designed a Support Vector Machine-based algorithm to identify the species of a mosquito specimen captured via smartphone images. But our preliminary work had limitations. Our dataset in [23] included only 303 images spread across nine mosquito species, and served more as a proof of concept, while our dataset in this paper is much larger with 25867 images. The dataset in this paper contains images taken with many more types of smartphones, and in multiple backgrounds and orientations. With this larger scale dataset, we design deep learning architectures (which are state-of-the-art) in this paper for three classes of problems a) to identify genus alone; b) to identify species based on knowledge of genus type; and finally c) to directly identify the species type. We also contrast the performance of each architecture and provide contextual relevance to ensuing results. As such, we believe that our work in this paper is significantly advanced from our work in [23], and enables more practical citizen/ smartphone assisted surveillance of mosquitoes in nature.

3 Generation of Image Database and Noise Removal

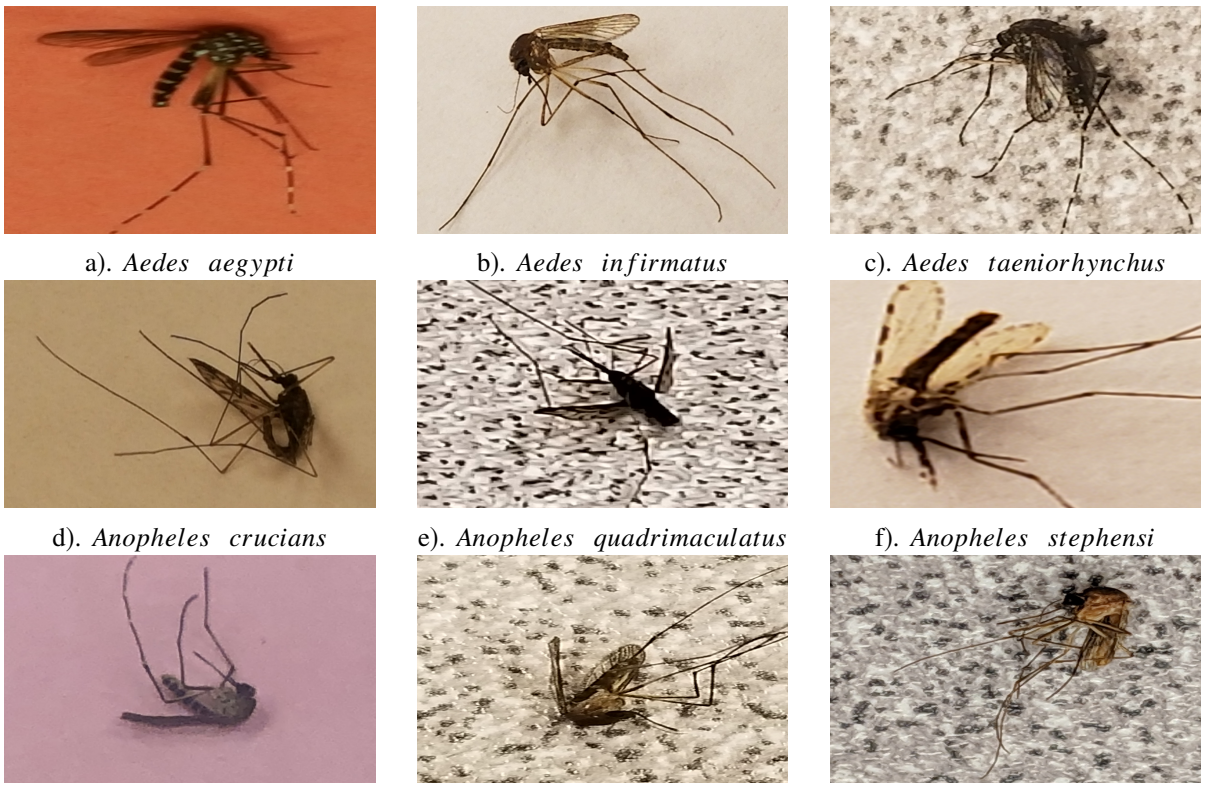
3.1 Image Generation

We present a brief overview of our data collection procedure here. Between Fall 2018 and Spring 2019, we visited the Hillsborough County mosquito control board in Florida to lay mosquito traps, collect specimens the next day, and bring them back to the lab for identification. Some of these specimens (250 in number) after identification by experts were given to our team for image generation. These specimens were spread across three genera and nine species presented in Table 1.

Subsequently, each specimen was imaged by our team under normal indoor light conditions using ten different smartphones, across multiple orientations, and on three different backgrounds. The ten phones used for imaging were Samsung Galaxy S8 (3 phones), Samsung Galaxy S9 (2 phones), iPhone 7 (2 phones), iPhone 8 plus (2 phones), and Pixel 3 (1 phone). These phones are relatively state-of-the-art. The three different backgrounds were white, pink and a cream colored tile. As a result if this process, 6807 images in total were generated by our team. Figure 1 illustrates one representative image from our dataset of each of the nine species (spread across three genera) we classify in this paper [1]

While our dataset may seem to be relatively large and comprehensive for model development, there is one practical challenge that needs to be overcome. During run-time execution by citizens, unseen variances will creep in images taken by them that will compromise classification. Such variances are numerous and stem from camera hardware, degree of zoom, blurriness, ambient light and more. While it is simply impossible to account for every possible source of variance, we still augment each image in our training dataset by zooming

1. Among the nine species classified in this paper, eight were trapped in the wild. Only *Anopheles stephensi* mosquitoes were raised in a lab environment in Florida, whose ancestors were originally trapped in India.



a). *Aedes aegypti* b). *Aedes infirmatus* c). *Aedes taeniorhynchus*

d). *Anopheles crucians* e). *Anopheles quadrimaculatus* f). *Anopheles stephensi*

g). *Culex coronator* h). *Culex nigripalpus* i). *Culex salinarius*

Fig. 1: One Representative Sample in our Dataset for Each Species Classified. This Figure is best viewed in Color.

TABLE 1: Relevant Details on our Dataset of Mosquito Species [4]

Genus	Species	No. of Image Samples	Disease Spread	Geographical Location
Aedes	*aegypti*	759	Chikungunya, Dengue, Yellow Fever, Zika	Africa, Asia, North America, South America
	infirmatus	741	Eastern Equine Encephalitis	North America, South America
	taeniorhynchus	761	Eastern Equine Encephalitis, West Nile	North America, South America
Anopheles	*crucians*	790	Malaria	Africa, North America, South America
	quadrimaculatus	810	Malaria	North America
	stephensi	756	Malaria	Africa, Asia
Culex	*coronator*	712	St. Louis Encephalitis, West Nile	Africa, North America, South America
	nigripalpus	703	Eastern Equine Encephalitis, St. Louis Encephalitis, West Nile	Africa, North America, South America
	salinarius	775	Eastern Equine Encephalitis, St. Louis Encephalitis, West Nile	Africa, North America, South America

in/ out (randomly chosen between 75% and 150%), and adjusting brightness/ contrast (randomly chosen between 1.0 and 1.5). Via this procedure, the total number of mosquito images generated and augmented for model development and

validation was 25867.

A note of gender of mosquitoes trapped: We point out that all the traps laid to capture mosquitoes were CO_2 traps. This meant that the specimens we trapped were all female, since only female mosquitoes seek out a bloodmeal (to provide nutrients for their eggs), and they are attracted to CO_2 emanating from the traps, thinking that the CO_2 comes from a blood source. As such, our model is trained to only recognize female mosquitoes. But, it is important to note that, male and female mosquitoes belonging to the same genus and species are anatomically similar except that the males have a distinct feather like antennae [16]. As such, we are confident that our technique can be easily adapted to identify male mosquitoes as well (not done in this paper though).

3.2 Noise Removal

Mosquito images taken from smartphone cameras contain three common types of noise: random noise, fixed pattern noise and banding noise. These types of noise stem from camera hardware, ambient conditions, variations in shutter speed, hand movements etc., and must be removed prior to classification. It is easy to infer that similarities within each anatomical component of a mosquito image should ideally be leveraged for noise removal. But, as we see in Figure 1, the same anatomical component (e.g., legs and wings of a mosquito) appear at multiple locations within a single image. Thus, exploiting purely localized trends in the image for noise removal has limited applicability for our problem. As such, we employ a non-local means denoising technique [3], wherein a mosquito image is smoothed by taking the mean RGB value of all pixels in the image, weighted by how similar these pixels are to any target pixel. This method ensures that similarities in anatomical components within a mosquito are effectively integrated, hence resulting in superior post-filtering clarity, and less loss of detail in the image. We explain below.

Consider a noisy mosquito image $v = \{v(i) \mid i \in I\}$, where $v(i)$ is the RGB value of pixel i. Then, the estimated value $NL[v](i)$, for a pixel i in image v after noise removal, is computed as a weighted RGB average of all pixels in the image as,

$$NL[v](i) = \sum_{j \in I} w(i,j)\, v(j),$$

where the weights $w(i,j)$ depend on the similarity between the pixels i and j in the image, and satisfy the conditions $0 \leq w(i,j) \leq 1$ and $\sum_j w(i,j) = 1$. The similarity between two pixels i and j depends on the similarity of the intensity vectors $\hat{v}(N_i)$ and $\hat{v}(N_j)$, where $\hat{v}(N_i)$ denotes RGB values of entries in a square matrix N_i, which in-turn denotes a square neighborhood of fixed size and centered at a pixel i. This similarity is measured as as an Euclidean distance, $||\hat{v}(N_i) - \hat{v}(N_j)||^2$. With this method, the goal is to ensure that those pixels within the image that represent the same anatomical component of pixels in the neighborhood $\hat{v}(N_i)$ of any pixel i (chosen for noise removal) will have higher weights on the average, hence smoothing the image more effectively and retaining anatomical consistencies within the image. The weight $w(i,j)$ is computed as

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||\hat{v}(N_i) - \hat{v}(N_j)||^2}{h^2}},$$

where $Z(i)$ is normalizing factor given by,

$$Z(i) = \sum_j e^{-\frac{||\hat{v}(N_i) - \hat{v}(N_j)||^2}{h^2}},$$

and where h denotes the degree of filtering. For best results, we kept the neighborhood size around a target pixel as $[7, 7]$ and h was set as 10. All images in our dataset were processed in this manner to remove noise before model training.

4 Our Deep Neural Network Architecture to Classify Mosquitoes

We now present our neural net architecture to classify mosquitoes from smartphone images. To keep this simple, and easier to explain, we first present our technique to identify only the *genus* of the mosquito from an image. Once we do that, we then present how the technique is adapted for classifying species based on genus, and classifying species directly.

4.1 Architecture Rationale and Transfer Learning

The problem of classifying mosquitoes from smartphone images is complex and challenging. Recall from the above that we focus these discussions on the problem of identifying the genus of a mosquito only. Some obvious challenges here include the need for faster training, reducing computational overhead, and minimizing overfitting. Apart from these, one unique challenge is that the size and position of the actual 'mosquito' within an image is not consistent as we see in Figure 1. Traditionally stacking convolution layers deeper and deeper for learning is computationally expensive and will not give good performance. Furthermore, they cannot effectively handle images, where the object of interest, in this case, a mosquito, can be of any arbitrary size and be at an arbitrary location within the image captured.

To overcome these challenges, we started our model development with the Inception Net architecture [32]. In this architecture, multiple kernel sizes (compared to a single sized kernel) are used at the same layer to better compensate for objects of interest being at arbitrary locations and of arbitrary size in an image. Specifically, in the Inception Net architecture, these filter sizes are set as $1 \times 1, 3 \times 3$ and 5×5. Also, the Inception Net architecture enables smart factorization methods to break down larger sized kernels into multiple smaller sized kernels that speed up operation without compromising learning.

However, this itself is not enough for our problem. As the network goes deeper, the gradient of the activation function that propagates through these layers becomes smaller, effectively preventing learning at later layers. To overcome this problem, we combined the Inception Net architecture with another popular neural network architecture called ResNet [13]. Basically, in the ResNet architecture, residual connections are introduced. A residual connection is a connection

from layer x directly to layer $x + n$ (where $n > 1$). In this modification, gradients that are sufficient to improve learning are still maintained at later layers, hence improving learning ability of the architecture, while also speeding up training. To summarize here, the initial architecture we started with for our problem is the Inception-ResNet-V2 (IRV2) [31] architecture (that combines the Inception and ResNet architectures), which has a total of 782 layers. The IRV2 architecture has been trained on the well known Imagenet dataset [7] that consists of 14 million images categorized across 1000 diverse classes commonly encountered in daily life, and the architecture achieves very good accuracies in classification for this dataset.

Leveraging Transfer Learning in Architecture Design: The Inception-ResNet-V2 architecture is very well studied. However, among the 1000 classes that it was trained to recognize, none are representative of mosquitoes. As such, the issue at hand is how can we leverage core learning abilities of this already trained architecture, while simultaneously adapting it to our problem of classifying mosquitoes. In this realm, transfer learning [28] is an excellent approach, wherein the essential idea is to leverage from a well trained neural net architecture and adapt it for a different problem. What needs to be addressed though are a) how many layers from the Inception-ResNet-V2 (IRV2) architecture do we keep as our initial layers; b) how many additional layers do we add customized to our contexts; and c) how to adapt the weights across each layer for best training to classify mosquitoes. These are presented below.

4.2 Optimization of Hyperparameters

Now that the rationale of our neural net architecture for mosquito genus identification is clarified, there are several key parameters in the architecture (called as hyperparameters) that we need to optimize. We elucidate these below, following which the final architecture for genus identification will be presented. Also note that since the three problems of interest to this paper (genus, species within genus, and direct species identification) all relate to classifying mosquitoes, the choice of hyperparameters are not going to be too different for each problem.

a). Image Resizing: To keep the images consistent, we need resizing. Since, for our problem, we collected data from multiple smartphones, we had images of different resolutions ranging from 450×550 to 2988×5322 pixels. To bring uniformity, and reduce the image size (for faster training without loss of quality), we resized each input image to 299×299 pixels irrespective of their original size. Finally, we normalized the RGB value of each pixel of the image by dividing it by 255 before training starts.

b). Loss function: This is the evaluation criteria of training, and measures the learning outcome of an architecture. Typically, it ranges from 0 to 1 where 0 means perfect learning and 1 means no learning. The goal of training is to ensure that the architecture designed and the weights in each layer best minimizes the loss over training data. An ideal loss function should be straightforward to calculate, and it should be easy to find its derivative, since we calculate the loss and its derivative for each image in every training iteration. For this study, we employed the categorical cross entropy loss function. It is defined for a singe training image j as:

$$CCE_j = -\frac{1}{K}\sum_{i=1}^{K}(y_i \log(p_i)), \tag{1}$$

where y_i is a 1-d ground truth indicator if class i (i.e., genus) is true ($y_i = 1$), or false ($y_i = 0$); K is the total number of classes; and p_i is the predicted probability of class i. Naturally, for all the training images (denoted as N) in the dataset, the categorical cross entropy loss is given by

$$CCE = -\frac{1}{N}\sum_{j=1}^{N} CCE_j. \tag{2}$$

c). Optimizer: An optimizer is an algorithm which helps converge an architecture during training from an initial value to the optimized one where the loss is minimum. In this study, we have employed Adam (Adaptive Moment Estimation) [18] optimizer algorithm. The idea here is to use adaptive learning rates for weights among layers in the architecture, such that lower rates are assigned to the weights that are getting bigger updates, and higher rates are assigned to weights that are getting smaller updates.

d). Learning Rate: This is the rate of change which a parameter in an architecture undergoes every epoch, while the optimizer navigates through the loss function in search of the minima. The learning rate essentially influences the speed of training. If the learning rate is too low, then the network will take more time to converge as the gradient step to change will be very small in each iteration, and on the other hand if it is too high, then it might overstep the minima. A good learning rate should be high in the initial epochs so that it quickly gets closer to the minima, but low in later epochs so as to not overstep it. In this realm, we have utilized Cyclic Learning Rate (CLR) [29] technique. The core idea here is to let the learning rate vary within a range of values, instead of pre-defining a linearly or an exponentially decreasing rate. Subsequently, we set a clear range of learning rates by cyclically vary the rates from a pre-defined range from 2×10^{-7} to 2×10^{-5}. While multiple function forms can vary the rate cyclically, the triangular form (linearly increasing and then linearly decreasing) has demonstrated to be simple and effective for our problem in this paper.

e). Architecture Fine-tuning and Compensating for Overfitting for Genus classification: Having discussed choices for key hyperparameters in our architecture, we now present how our architecture is trained and fine-tuned for genus classification. Many steps are involved here including decision on which layer to start from the Inception-ResNet-V2 (IRV2) [31] architecture (among the 782 layers), how to add remaining layers for our specific genus classification problem, how to assign weights to them, how to avoid overfitting problems, and finally, when to stop the training.

For our problem, we initially started at layer 350 in the Inception-ResNet-V2 (IRV2) [31] architecture. Starting from layers too early will lead to poor learning, and starting from layers too deep will likely lead to over-fitting and also induce

computational overhead. After initialization, subsequent layers were initialized with Glorot uniform initialization technique [9]. Since, IRV2 weights are highly optimized we didn't want to change them too much but wanted to optimize the weights of remaining layers. As such, we divided the training process in two phases. In first phase, we froze the weights of IRV2 layers and only allowed the changes in remaining layers which we iteratively added with high learning rate. After 500 epochs, the loss curve reached the plateau where we saved the weights of the model. In second phase, we unfroze the weights of IRV2 and decreased the learning rate to 0.00001 to setup slow training. In this phase, every single weight across all layers in the architecture was allowed to change. After 1200 epochs, training and validation loss function attained a plateau and we stopped the training. Note that during training (and as is common in complex classification problems), we identified that our architecture suffered from overfitting problems, which we compensate by infusing a combination of three different regularization techniques between layers, namely dropout, batch normalization and maxnorm. Briefly, dropout technique [30] randomly ignores few neurons in each layer depending on the percentage set, while batch normalization [14] is another technique to normalize the output of previous activation layer using the mean and standard deviation of the batch of input images on which training is done. Maxnorm [21] is another regularization technique to enforce an absolute upper bound on the magnitude of the weight vector for every neuron and uses projected gradient descents to enforce the constraint.

f). The Finalized Architecture for Genus Classification: Table 2 illustrates the key parameters of the finalized architecture for classifying genus types of mosquitoes. The term "block17_10_Conv" denotes the 433^{rd} layer of the IRV2 architecture [31], upto which was utilized for our problem. The remaining layers (elaborated upon below) are added after that as specified in Table 2 for genus identification. The entries in the fields "Size In" and "Size Out" in the table refer to dimensions of the input and output matrices in the corresponding layer. After the 433^{rd} layer (of the IRV2 architecture), we add a Global Average Pooling Layer to reduce dimensionality to only one dimension, following which four dense layers are added to the architecture (which essentially means all neurons in one layer are connected to those in the next layer). The dimensions of the dense layers are also presented in Table 2. Then, we concatenate the dense layers and finally perform softmax operation to output probabilities of classification between 0 and 1 for each of the three genera we attempt to classify (i.e., *Aedes*, *Anopheles* and *Culex*). This essentially summarizes our architecture for genus classification.

4.3 Adaptations of our Architecture for other Classification Problems

Tables 3, 4 and 5 illustrate the architectures for classifying species within the *Aedes*, *Anopheles* and *Culex* genus types respectively. Recall again that these architectures are designed to classify a species within the corresponding genus types only. Finally in Table 6, we present the architecture for classifying directly the species only. Interestingly, the optimized architecture for this problem was the same the one optimized for genus level classification, except that the number of possible classes here is *nine* (for the nine species).

For clarity purposes, an illustration of the architectures for all classification problems in presented in Figure 2.

TABLE 2: Genus Architecture

Layer	Size In	Size Out
$block17\_10\_conv$ (Layer 433 in IRV2)	($None$, 17, 17, 384)	($None$, 17, 17, 1088)
$GlobalAveragePooling2D$	($None$, 17, 17, 1088)	(1, 1088)
$dense\_1$	(1, 1088)	512
$dense\_2$	512	256
$dense\_3$	256	128
$dense\_4$	128	256
$concat\_1$	($dense\_1$, $dense\_2$, $dense\_3$, $dense\_4$)	1152
$softmax$	1152	3

TABLE 3: *Aedes Architecture*

Layer	Size In	Size Out
$conv2d\_93$ (Layer 346 in IRV2)	($None$, 17, 17, 160)	($None$, 17, 17, 192)
$GlobalAveragePooling2D$	($None$, 17, 17, 192)	(1, 192)
$dense\_1$	(1, 192)	512
$dense\_2$	512	512
$dense\_3$	512	256
$dense\_4$	256	128
$concat\_1$	($dense\_1$, $dense\_4$)	640
$softmax$	640	3

5 Evaluation of our Architectures

We now are ready to present the evaluation of our neural net architecture for classifying mosquitoes from smartphone images. There are two broad mechanisms we present for evaluation. The first is based on Class Activation Map, and the next one is based on accuracy of classification.

5.1 Class Activation Map

The first approach we take to evaluate our technique is to determine whether or not the architecture we have designed to classify mosquitoes is actually able to focus in on the pixels corresponding to anatomical components within the mosquito

TABLE 4: *Anopheles Architecture*

Layer	Size In	Size Out
$block17\_8\_conv$ (Layer 401 in IRV2)	($None$, 17, 17, 384)	($None$, 17, 17, 1088)
$GlobalAveragePooling2D$	($None$, 17, 17, 1088)	(1, 1088)
$dense\_1$	(1, 1088)	512
$dense\_2$	512	512
$dense\_3$	512	256
$dense\_4$	256	256
$dense\_5$	256	256
$softmax$	256	3

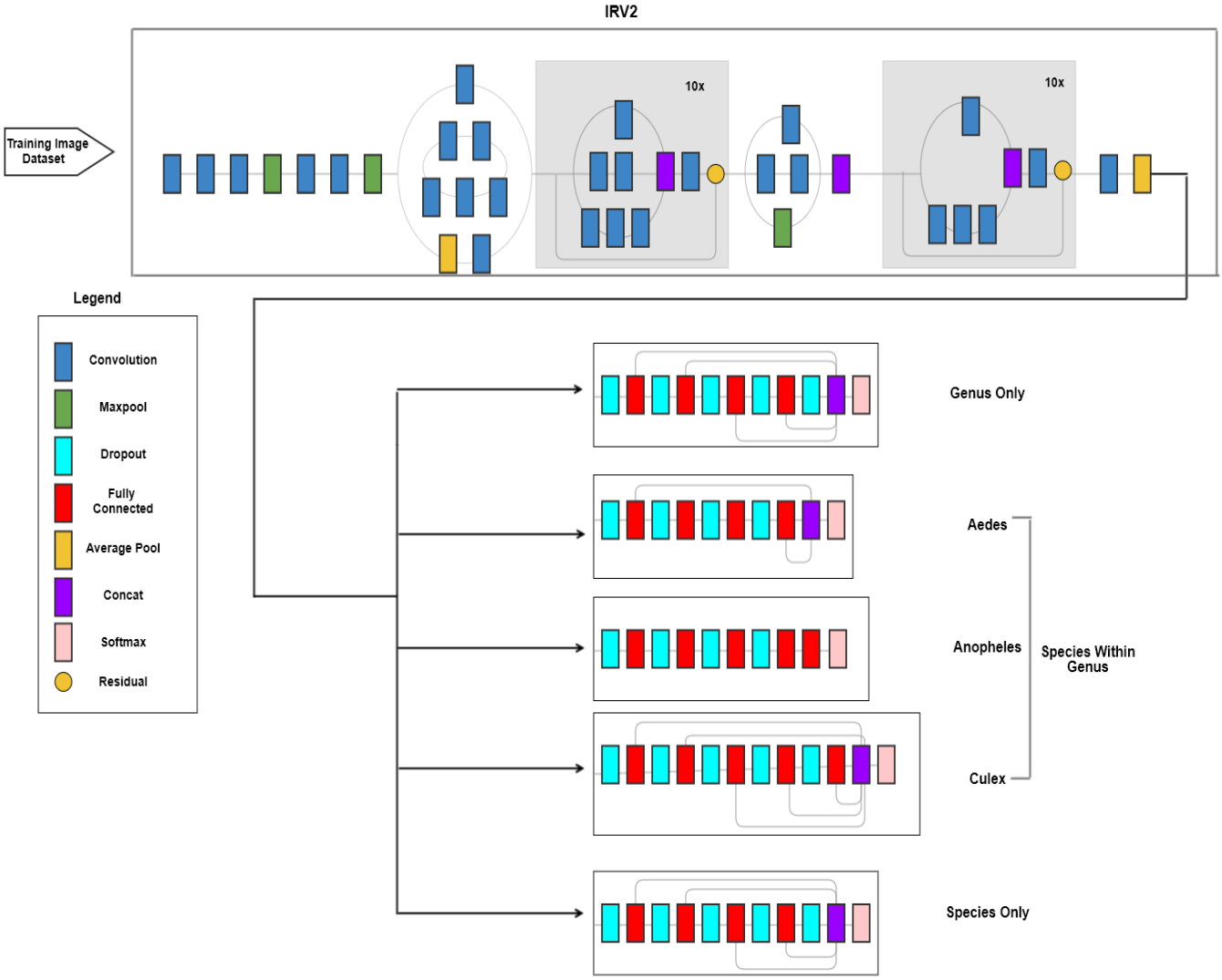


Fig. 2: Our Neural Net Architectures for Classifying Mosquitoes. This Figure is best viewed in Color.

TABLE 5: *Culex Architecture*

Layer	Size In	Size Out
$conv2d\_111$ (Layer 407 in IRV2)	$(None, 17, 17, 128)$	$(None, 17, 17, 160)$
$GlobalAveragePooling2D$	$(None, 17, 17, 160)$	$(1, 160)$
$dense\_1$	$(1, 160)$	512
$dense\_2$	512	128
$dense\_3$	128	256
$dense\_4$	256	512
$dense\_5$	256	256
$concat\_1$	$(dense\_1, dense\_2, dense\_3, dense\_4, dense\_5)$	2484
$softmax$	2484	3

TABLE 6: Species-only Architecture

Layer	Size In	Size Out
$block17\_10\_conv$ (Layer 433 in IRV2)	$(None, 17, 17, 384)$	$(None, 17, 17, 1088)$
$GlobalAveragePooling2D$	$(None, 17, 17, 1088)$	$(1, 1088)$
$dense\_1$	$(1, 1088)$	512
$dense\_2$	512	256
$dense\_3$	256	128
$dense\_4$	128	256
$concat\_1$	$(dense\_1, dense\_2, dense\_3, dense\_4)$	1152
$softmax$	1152	9

image, while simultaneously being able to exclude the background pixels. Needless to say, if a model is trained to classify a mosquito by processing *only* those pixels corresponding to the anatomical components within the image, then such a model has high fidelity. The technique of Class Activation Map (CAM) allows us to do that.

Once an input image is classified after the end of the Softmax layer in Figure 2, the CAM technique enables us

to infer which pixels within the image were used by the neural-net architecture to make the classification. To do so, we traverse back in the architecture to identify the feature map for that particular image (at the conclusion of the last convolutional layer in the architecture in Figure 2), and the weights corresponding to the class (i.e., the type of mosquito) that the image was identified with. If we denote the feature map for an image at Kernel k in the last convolutional layer as $f_k(i, j)$, and if we denote the weight of each kernel connection for class c as w_k^c, then, we compute the expression,

$$M_c(i, j) = \sum_k w_k^c f_k(i, j), \tag{3}$$

for each spatial location (i, j) in the convoluted image. This essentially computes the importance of each feature map in the convoluted image when a classification is made. Subsequently, the value of $M_c(i, j)$ in the convoluted image is projected back onto the corresponding pixels in the original image to create a heatmap. The higher the value of $M_c(i, j)$, the warmer the color of those corresponding pixels in the heatmap. These are the pixels within the image that were used predominantly to make the classification. As such, if in our images if most of the higher intensities are concentrated in critical anatomical components of the mosquito image, then we can trust our model better, as elaborated below.

Discussions on CAM Results: We present results from Class Activation Map in Figure 3 for the problem of classifying mosquito species. Specifically, we highlight one representative image for each of the nine species we attempt to classify, but our results are representative for other images also. From Figure 3, we see that our architecture is reliable. Irrespective of the background, or phone, we can see that the pixels with the highest intensities are concentrated in the most critical anatomical components of the image (thorax, scutum, wings, abdomen and legs). It is a fact that these are the anatomical components that are most vital for visual classification of a mosquito specimen - scutum for *Ae. aegypti*; wings for *An. crucians* and *An. quadrimaculatus*; abdomen for *Ae. taeniorhynchus*; thorax for *Cu. nigripalpus* and *An. stephensi* etc. A careful observation of heatmaps presented in Figure 3, and more which we manually verified indeed conveys that our model has learnt to recognize the critical anatomical components within the mosquito image for classification. For interested readers, related discussions on anatomical components and markers that entomology experts use for visual classification of mosquitoes are available in [4].

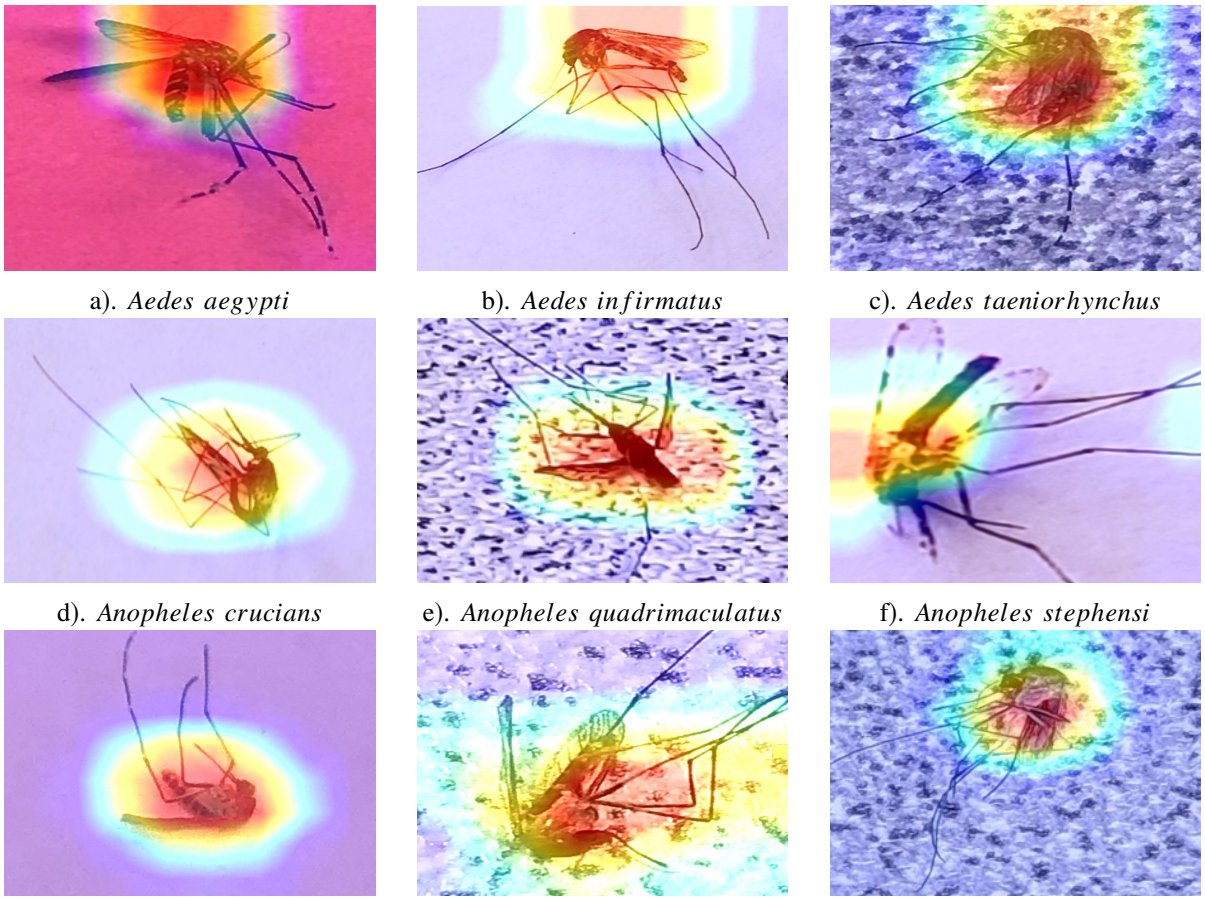
5.2 Accuracy of Classification

We now present our results on accuracy of classification using well-established evaluation procedures. First, we briefly describe our dataset again for clarity. Recall that our dataset for model development contained 250 mosquito specimens evenly spread across three genera, with three species each per genus as presented in Table 1. From these mosquito specimens, we generated a dataset of 6807 images taken via multiple smartphones and under multiple backgrounds as presented in Section 3. Subsequently, 30% of these images (2042 in number) were separated out for validation alone. The remaining 4765 images were augmented for training (following procedures also described in Section 3) to yield a dataset of 23825 images that were used to train the architecture. Table 7 presents results for the three classes of problems attempted in this paper - genus only, species within genus, and species only. The classification accuracies presented are those wherein the accuracy of classification with the training images (23825 images) best matches the classification of accuracy with validation images (2042) when training was concluded. The classification accuracies presented represent the proportion of correctly identified images within each class.

Subsequently, once the architecture was fully trained and validated, we conducted another evaluation strategy, which is to test the architecture with completely unseen images post completion of training. This is a much more rigorous testing strategy, for which we trapped another 50 mosquito specimens evenly distributed among the nine species for which the model was trained. For this, we captured three images of each mosquito specimen in a single phone and under one background. These three images for a mosquito specimen is called a *set* (captured with the same phone and on top of the same background). In this manner, for each of the 50 mosquito specimens, we generated 12 such sets using four different phones and on top of three different backgrounds (wherein no two sets per mosquito specimen had the same phone-background configuration). A total of 600 such sets were generated for testing, wherein we reiterate that each set had had a total of three images per mosquito specimen (that were taken via the same phone, and on top of the same background). This dataset (we reiterate) is completely unseen by our model. Classification results for this dataset are presented in Table 8. Note here that the identification of a mosquito specimen for each *set* (comprising of three images per specimen) is based on computing the maximum of the average of the class probabilities as outputted by our architecture for each of the three images in that set. The results are elaborated below.

Discussions on Accuracy Results: First, off the accuracy of classification during model training and validation in Table 7 is better than those with testing on unseen data after training in Table 8. This is natural, although, the accuracies are not too far off, hence justifying the fidelity of our model.

Secondly, we see that genus classification accuracies are high for all three genera. This is also encouraging since identifying the genus itself is very useful during outbreaks. Once we start the finer-grained process of identifying species, we see interesting results. Mosquitoes belonging to the *Aedes* genus are being classified better. It is generally true that even for experts, visually recognizing *Ae. aegypti* (a deadly vector for many diseases) is easy because of the very distinct ‘lyre’ shaped pattern on the thorax of those species. Our model can also identify *An. crucians* and *An. stephensi* much more accurately. This is because *An. crucians* mosquitoes have three distinct dark spots on the wings, which aid identification. However, *An. quadrimaculatus* mosquitoes have four dark spots on their wings, and so there is some confusion between *An. crucians* and *An. quadrimaculatus* mosquitoes. *An. stephensi*



a). *Aedes aegypti* b). *Aedes infirmatus* c). *Aedes taeniorhynchus*

d). *Anopheles crucians* e). *Anopheles quadrimaculatus* f). *Anopheles stephensi*

g). *Culex coronator* h). *Culex nigripalpus* i). *Culex salinarius*

Fig. 3: One Representative Image Sample with Class Activation Map in our Dataset for Each Species Classified, across multiple backgrounds and phones. This Figure is best viewed in Color.

TABLE 7: Validation Accuracy

Genus	% of Accuracy	Species Within Genus	% of Accuracy	Species	% of Accuracy
Aedes	92%	*aegypti*	85%	*aegypti*	86%
		infirmatus	84%	*infirmatus*	83%
		taeniorhynchus	81%	*taeniorhynchus*	78%
Anopheles	93.5%	*crucians*	89%	*crucians*	93%
		quadrimaculatus	79%	*quadrimaculatus*	72%
		stephensi	98%	*stephensi*	100%
Culex	92%	*coronator*	69%	*coronator*	68%
		nigripalpus	64%	*nigripalpus*	71%
		salinarius	72%	*salinarius*	63%

TABLE 8: Testing Set Accuracy

Genus	% of Accuracy	Species Within Genus	% of Accuracy	Species	% of Accuracy
Aedes	81%	*aegypti*	82%	*aegypti*	82%
		infirmatus	90%	*infirmatus*	40%
		taeniorhynchus	50%	*taeniorhynchus*	38%
Anopheles	77%	*crucians*	89%	*crucians*	73%
		quadrimaculatus	60%	*quadrimaculatus*	40%
		stephensi	98%	*stephensi*	93%
Culex	92%	*coronator*	48%	*coronator*	20%
		nigripalpus	43%	*nigripalpus*	40%
		salinarius	51%	*salinarius*	30%

mosquitoes have distinct yellow colored scutum, thorax and wings, which makes it a unique species, easy to identify. Most confusion happens with species belonging to the *Culex* genera, since they share similar brownish-copper color patches with *Ae. infirmatus* mosquitoes. As such, we can infer that the results we obtain for classification and the sources of confusion directly correlate with corresponding morphological similarities/ dissimilarities among specimens in our model, which aid or confuse experts during visual classification of mosquitoes. To elaborate the sources of confusion better, we present the corresponding confusion matrix for the nine species we classify in this paper for the species only model is presented in Figure 4b. We once again direct interested readers to [4] for related discussions on anatomical markers that entomology experts use for visual classification of mosquitoes.

One very intriguing finding from our analysis is that, at both the genus and species levels, the classification accuracies we obtain match the evolutionary relationships of mosquitoes in our model. These relationships are reflected in the phylogenetic tree Figure 4a, primarily reconstructed based on genetic sequencing [10] [11] [27]. Starting from the left, the *Anopheles* clade diverged from the other mosquitoes $\sim$ 217 million years (mya) ago [27]. Interestingly, the *Anopheles* mosquitoes here have the highest classification accuracies in our model (i.e., high true positive rate and low false positive rate), as seen in Figure 4b. This correspondence between phylogenetic pattern and classification accuracy in our model holds true for species within each genus as well. Within the *Anopheles* genus for instance, the Old World species, *An. stephensi* ($\sim$ 107 mya divergence) [24] has a higher classification accuracy in our model than those of the more closely-related New World sister groups, *An. crucians* and *An. quadrimaculatus*. Similarly, within *Aedes*, *Ae. aegypti* is the most evolutionarily distinct species ($\sim$ 92 mya divergence), and it has the highest classification accuracy in our model for the *Aedes* genus. The evolutionary relationships among species in the *Culex* genus were unable to be resolved morphologically even by biologists today (illustrated pictorially in Figure 4a) [11], and this fact is indeed reflected in the lower levels of classification accuracy in our model for species in the *Culex* genus (Figure 4b).

Ultimately, this suggests that evolution – as measured by relative divergence times – drives greater anatomical disparity, which in turn yields higher classification accuracies when using AI. In other words, our paper shows that signals from millions of years of natural selection is being revealed by real-time AI in our models. Naturally, these discoveries provide much more validity to our work in this paper, and also its potential for future broader impact in diverse scientific disciplines.

Summary: To summarize here, we believe that our system to identify mosquitoes from smartphone images is practical, and useful, especially for detecting the presence of really deadly mosquitoes like *Ae. aegypti* (vector for dengue, yellow fever, chikungunya, and Zika) and also mosquitoes in the *Anopheles* genera (vectors for malaria). Potential sources of confusion in our system are due to visual similarities among various types of mosquitoes that complicate classification even for experts in the field. In Section 7, we present more details on the practical ramifications of our contributions and how to improve practical utility.

6 Smartphone App Development

We implemented our entire system for classifying mosquitoes as a smartphone based app interacting with the cloud, that allows any citizen to take a photo of a mosquito specimen for identification. We used a popular cross-platform, open-source application development framework called React Native to develop both an Android and an iOS version of our system.

The app will first pre-process the image to correct for blurriness and brightness. Then the app will execute a simple neural net model embedded within the smartphone to check if the photo is that of a mosquito or not. If the photo is identified to be that of a mosquito, then the image is uploaded to a secure cloud, that executes our neural net models presented above for classifying genus and species of the mosquito whose image was captured. We present critical specs of our system below.

6.1 Pre-processing to Correct for Blurriness and Brightness

Compensating for Blurriness: Images of mosquitoes taken by citizens can suffer from blurriness, and must be corrected. To calculate the blurriness level of an image, we used a method called Laplacian Variance [26], which requires us to obtain a grayscale channel of pixels in an image and convolve them with a 3×3 kernel, which for our problem was set as the following Laplacian kernel

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

After calculating the convolution, we then take the variance of the output. If the variance is above a pre-defined threshold, then the image is considered not blurry. Otherwise, the user is prompted to retake the image.

Compensating for Poor Brightness: In order to guarantee the brightness of the image to enable classification, we compare its brightness value to that of the training set used in model development. One approach is to convert each image at run-time from RGB color space to HSV (Hue, Saturation, Value). Here, Value (V) is the brightness of the color and varies with color saturation, from 0 to 255, wherein higher values indicate more brightness. From out dataset of 25000 mosquito images, we identified that the mean was $V = 187.44$. For each image taken at run-time, its V value is computed and compared against this threshold, and if too different from the threshold, the brightness is automatically adjusted within the app before processing.

6.2 A Binary Classifier to Identify a Mosquito image from one that is not a Mosquito

In order to ensure fidelity, and to minimize wasted computational resources, we implemented a simple neural net model that executed on the phone to classify whether or not an image generated by a citizen is that of a mosquito or not. If the

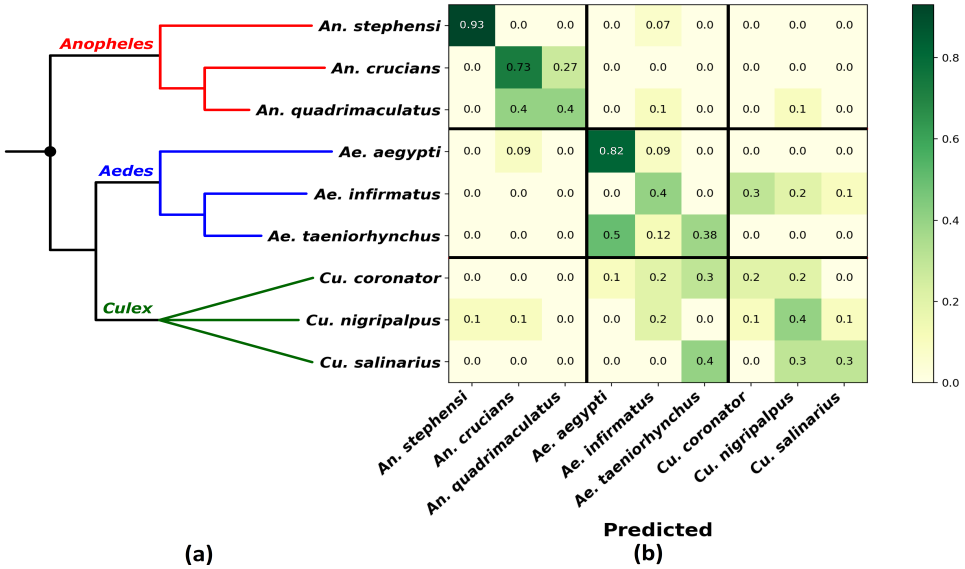


Fig. 4: **(a)** Phlyogenetic tree showing evolutionary relationships of the mosquitoes [10] [11] [27]. Leftward is further back in geologic time. Black dot denotes the common ancestor, ~ 217 million years ago [27]. **(b)** Species testing set confusion matrix with classification accuracies. This Figure is best viewed in Color.

image is not that of a mosquito, then the image is simply discarded with a note to the user. This model was trained using our dataset of more than 25000 mosquito images, against the very popular COCO (Common Objects in Context) data-set [22], that contains more than 330K images categorized into 91 classes. Due to space limitations we do not discuss our neural-net architecture, but results of close to 95% accuracy were achieved by our architecture for this binary classification problem. We reiterate that this preliminary check is vital in order to prevent transmission and processing of images in the cloud that are irrelevant to the problem at hand.

6.3 Critical Specs of the Overall System

The Android version of our application supports devices with SDK above 21, which runs on close to 90% of Android devices worldwide [2]. The total size of the release package is 84MB. The iOS version of the application supports devices with iOS above 8.0, which runs on almost all iPhone devices worldwide. The total size of the release package is 60MB.

Once an image has passed all the pre-processing phases, our app uploads the image to a customized database, which uses Firebase as a backend. The images are stored in the Firebase storage and a request for classification is made to our GPU server that executes the classification model. The classification takes only a few seconds and results are delivered back to the user through the app. The image and classification results are displayed and locally stored on the user's phone, as well on the Firebase backend. On-going pilot tests of our system are very encouraging, and we plan to release our app and datasets to the public very soon.

7 Discussions on Practical Utility of our System, Limitations and Scope for Future Work

Our system proposed in this paper has significant impact in multiple disciplines. First off, during peak seasons in tropical climates (when it rains), hundreds of thousands of mosquitoes can breed, and these include vectors for deadly diseases, along with relatively harmless ones. As such, during disease outbreaks, identification of the type of mosquitoes present in an area of interest and their abundance become vital for public health. Our system can help in such cases, wherein, anyone, even a non-expert, can simply take a picture of the mosquito specimen to perform classification, and results are automatically uploaded to the cloud. With a system like ours, the current trend of manual classification can be replaced with more automated methods, and skill of experts in mosquito control can be used elsewhere. Our system can work in low-income countries as well with minimal cost.

Secondly, and more importantly, common citizens can also participate in this effort. With their smartphones, they could also capture and share images of mosquitoes that they encounter (not un-common where they are endemic). Ultimately, this precise mosquito data could be harnessed to generate real-time distribution maps, as well as habitat maps to model the suitable ecological niches and potential future spread of these disease vectors [15] [20] [6] [5]. The impact towards public health can be significant with enhanced citizen participation and awareness of the problem. Remarkably, such efforts are slowly becoming prominent in the USA and also globally. One major project in this realm includes The GLOBE Mosquito Project [1], where citizens are encouraged to capture and upload images of mosquito breeding spots they encounter in nature. Our work in this paper can add significant value to such efforts on a national and global scale.

Thirdly, from an entomology perspective, our work has im-

pact. With more data and more involvement from citizens, our efforts could enable the creation of a massive scale database with thousands of tagged images of mosquitoes captured in the wild. Such datasets do not exist yet, and can add significant utility to help experts visually compare specimens across genus and species types; observe, verify and comprehend existing or new morphological characteristics across genus and species types; and also study how the same species evolves differently in different parts of the world. Our efforts can help expand the scope to many mosquito species across the globe, and possibly to other insects and tiny organisms as well.

However, we agree that our system does have limitations. First off, our current dataset is limited to only *nine* species, spread across *three* genera. Even in this dataset, our accuracies are far from being perfect to the point where citizens can immediately start using our current system. We are currently working to pilot our data collection platform on a larger scale. We are soliciting the assistance and partnership of multiple mosquito control districts in Florida and beyond to aid us in this effort (including experts with The GLOBE Mosquito Project). One potential avenue of our exploration is 'anatomy-aware' deep learning. Here, instead of attempting to classify a mosquito image in its entirety, we are investigating techniques to extract the most critical anatomical components of the mosquito - thorax, scutum, wings, abdomen and legs, and then use each component independently to make a classification. This is an untested idea and could improve accuracies, because now if the scutum in an image indicates *Ae. aegypti*, then we can give this result much more weightage, since the scutum is the most distinguishing component to classify an *Ae. aegypti* mosquito (due to a unique 'lyre' shaped pattern in the scutum). Leveraging similar anatomical signatures for other types of mosquitoes during classification is our focus here. Recent advances in Computer Vision that are trained to detect and isolate objects of interest within an image [8] [12] can be leveraged for this task, which we are currently pursuing. Additional future work will include differentiating *Ae. aegypti* from sister taxon *Ae. albopictus* using computer vision. Importantly, *Ae. aegypti* is a more competent and anthropophilic disease vector, but is less cold-tolerant [15] and can be competitively displaced by *Ae. albopictus* locally. Therefore it is critical to track the real-time presence and abundance of these mosquitoes during outbreak scenarios, as well as detect introductions into new habitats from human travel and climate change [19].

8 Conclusions

In this paper, we design deep neural network architectures to classify the genus and species of mosquitoes from smartphone images. Our proposed architecture framework employs the principle of Transfer Learning, and is a hybrid of Inception Net and ResNet architectures. Our design also ensures faster training and is easier to optimize. With this framework multiple architectures were designed and evaluated to identify the genus alone; a genus-specific species model; and finally a species only model. We believe that our results and the right contexts in which the results should be interpreted give us the confidence that our proposed system is indeed practical. Our system is also implemented as a user-friendly smartphone app and critical specs were presented. Towards the end of the paper, we present important discussions on the utility of the system, its impact, limitations, and scope for future work.

9 Acknowledgments

We acknowledge support from the Hillsborough County Mosquito Control District in helping us identify mosquitoes trapped in the wild, and Dr. John Adams from the University of South Florida for providing samples of *An. stephensi* mosquitoes. This work was supported in part by US National Science Foundation under Grant CBET #1743985. Any opinions and findings are those of the authors alone and do not necessarily reflect the views of the funding agency.

References

[1] Globe home page.
[2] Distribution dashboard: Android developers. https://developer.android.com/about/dashboards?authuser=1, May 2019.
[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.
[4] N. D. Burkett-Cadena. *Mosquitoes of the southeastern United States*. University of Alabama Press, 2013.
[5] R. M. Carney. Gis-based early warning system for predicting high-risk areasof dengue virus transmission, ribeirão preto, brazil. MPH Thesis, Yale University, 2010.
[6] R. M. Carney, S. C. Ahearn, A. McConchie, C. Glaser, C. Jean, C. Barker, B. Park, K. Padgett, E. Parker, E. Aquino, et al. Early warning system for west nile virus risk areas, california, usa. *Emerging infectious diseases*, 17(8):1445, 2011.
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.
[10] R. E. Harbach. The phylogeny and classification of anopheles. In *Anopheles mosquitoes-new insights into malaria vectors*. IntechOpen, 2013.
[11] R. E. Harbach, C. L. Culverwell, and I. J. Kitching. Phylogeny of the nominotypical subgenus of culex (diptera: Culicidae): insights from analyses of anatomical data into interspecific relationships and species groups in an unresolved tree. *Systematics and biodiversity*, 15(4):296–306, 2017.
[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
[15] T. L. Johnson, U. Haque, A. J. Monaghan, L. Eisen, M. B. Hahn, M. H. Hayden, H. M. Savage, J. McAllister, J.-P. Mutebi, and R. J. Eisen. Modeling the environmental suitability for *Aedes* (*Stegomyia*) *aegypti* and *Aedes* (*Stegomyia*) *albopictus* (diptera: Culicidae) in the contiguous united states. *Journal of medical entomology*, 54(6):1605–1614, 2017.
[16] P. Jupp. Mosquitoes of southern africa: Culicinae and toxorhynchitinae. hartebeespoort, 1996.
[17] F. Kehinde, G. Dedeke, I. Rasaq, and P. Isibor. The potential of visible light spectra as control measure of mosquito, the vector of plasmodium. In *IOP Conference Series: Earth and Environmental Science*, volume 210, page 012009. IOP Publishing, 2018.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] M. U. Kraemer, R. C. Reiner, O. J. Brady, J. P. Messina, M. Gilbert, D. M. Pigott, D. Yi, K. Johnson, L. Earl, L. B. Marczak, et al. Past and future spread of the arbovirus vectors aedes aegypti and aedes albopictus. *Nature microbiology*, 4(5):854, 2019.

[20] P. Leandro-Reguillo, T. Panaou, R. Carney, and B. G. Jacob. Fuzzification of multi-criteria proxy geoclassifiable vegetation and landscape biosignature estimators to predict the potential invasion of aedes aegypti in barcelona, spain.

[21] J. D. Lee, B. Recht, N. Srebro, J. Tropp, and R. R. Salakhutdinov. Practical large-scale optimization for max-norm regularization. In *Advances in neural information processing systems*, pages 1297–1305, 2010.

[22] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[23] M. Minakshi, P. Bharti, and S. Chellappan. Leveraging smart-phone cameras and image processing techniques to classify mosquito species. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 77–86. ACM, 2018.

[24] M. Moreno, O. Marinotti, J. Krzywinski, W. P. Tadei, A. A. James, N. L. Achee, and J. E. Conn. Complete mtdna genomes of anopheles darlingi and an approach to anopheline divergence time. *Malaria journal*, 9(1):127, 2010.

[25] J. P. Munoz, R. Boger, S. Dexter, R. Low, and J. C.-H. Li. Image recognition of disease-carrying insects: A system for combating infectious diseases using image classification techniques and citizen science. In *HICSS*, 2018.

[26] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martínez, and J. Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 3:314–317 vol.3, 2000.

[27] K. R. Reidenbach, S. Cook, M. A. Bertone, R. E. Harbach, B. M. Wiegmann, and N. J. Besansky. Phylogenetic analysis and temporal diversification of mosquitoes (diptera: Culicidae) based on nuclear genes and morphology. *BMC evolutionary biology*, 9(1):298, 2009.

[28] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[29] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[33] A. B. B. Wilke, R. de Oliveira Christe, L. C. Multini, P. O. Vidal, R. Wilk-da Silva, G. C. de Carvalho, and M. T. Marrelli. Morphometric wing characters as a tool for mosquito identification. *PloS one*, 11(8):e0161643, 2016.