A Comparative Analysis of Different Convolutional Neural Network Architectures for Mosquito Genera Classification

Dhilan Shah, Suhani Shukla, Michael Squeri, Sameen Ahmad, Ankhi Banerjee,
Adriana Talianova, Saurav Bavdekar, Arav Sachdeva
Arnav Deol

2022

Abstract

Vector-borne diseases, such as Dengue Virus, Zika Virus, Malaria, and West Nile virus, cause an estimated 700,000 annual deaths. Aedes, Culex, and Anopheles, three common mosquito genera, carry and transmit these diseases. Climate change is worsening the frequency and severity of infection, and currently, there is a limited number of universally accessible cures. Mosquito genera's disease-carrying capacities vary due to their unique adaptations, making classification valuable in understanding the susceptibility of some viruses. Additionally, identifying mosquito genera can inform preventative measures to reduce disease transmission and help health professionals determine proper mitigation measures. With this information in mind, our group built a research project centered around the following question: which convolutional neural network (CNN) architecture can most effectively distinguish between Aedes, Anopheles, and Culex mosquito larvae? First, we extracted data from the GLOBE API. Then, using a Python algorithm and metadata containing mosquito genera, we sorted the images by genus. Finally, we trained four CNNs and compared their ability to identify mosquito genera with image classification. The architectures used for larvae classification were LeNet-5, AlexNet, VGG-16 Net, and ResNet-50. Each model was built in Google Colab in Python using Tensorflow and Keras libraries, trained using the larvae images, and analyzed to identify the best overall network. After the trials, ResNet-50 was the most effective model due to its low loss of 0.60. Understanding the accuracy of CNN architectures when identifying mosquito genera can support further research in the scientific community regarding computer vision techniques in biological fields. Most importantly, combining this research with other mosquito tracking algorithms can save lives, specifically in areas most sensitive to mosquito-vectored diseases.

Key Words:
Convolutional Neural Networks, Mosquito Larvae, Image Classification, Artificial Intelligence, Data Science

Research Question and Hypothesis

The main question guiding our research was: which convolutional neural network (CNN) architecture can most effectively distinguish between Aedes, Anopheles, and Culex mosquito larvae? Initially, we hypothesized that LeNet-5 would be the most successful model, due to its simplicity, which could prove advantageous with such a small dataset. This question can help us structure our project around a singular cause: finding the most effective CNN architecture. To do this, we will extract GLOBE data, build the CNNs using Keras, and run multiple trials to find the most effective model.

Introduction and Review of Literature

With vector-borne diseases accounting for 17% of global infectious diseases (Omodior, et al 2018), Mosquitoes are responsible for millions of deaths yearly. Diseases are spread to humans and animals alike through the bite of a disease-carrying mosquito. The *Anopheles, Culex,* and *Aedes* mosquito genera are mainly responsible for transmitting the most dangerous pathogens. *Culex* mosquitoes are vectors of West Nile fever, St. Louis encephalitis, and Japanese encephalitis. *Aedes aegypti* can transmit the Dengue virus and the Zika virus. Finally, *Anopheles Albinamus* can carry Malaria. Though environmental preferences differ among the genera, mosquitoes typically populate shaded woods near areas of standing water, such as wetlands, and are also attracted to increased temperatures. Additionally, these diseases are becoming increasingly dangerous, largely due to the changing environment, globalization, and population growth. Although many treatments are being developed for vector-borne diseases, they are still far from being universally accessible. Also, over time, insects have developed a resistance to insecticides, further complicating the search for a means of effective mosquito control and increasing the spread of vector-borne diseases.

In 2020 alone, Malaria still caused approximately 620,000 deaths (96% of which were in Africa) across 87 countries and territories (CDC, 2021*)*. Infants and pregnant women are the most susceptible to Malaria (WHO, 2022*)*. In addition to Malaria, other diseases including Dengue, which threatens about 40% of the world's population, and Lyme disease, the most commonly reported vector-borne infection in the United States (Tibayrenc, 2017), continue to be a problem due to mosquitoes. As more people suffer from these diseases, there is increasing pressure on institutions to implement preventative measures that may start with classifying mosquito larvae using CNNs or other image classifiers. Image classifiers can inform how governments and organizations invest their resources and focus their efforts. Determining mosquito larvae type is important to help prevent vector-borne diseases and thousands of deaths yearly can focus efforts to eliminate the breeding habitats of certain mosquitoes.

Global Learning and Observations to Benefit the Environment, also known as GLOBE, is a program that utilizes citizen science and involves the public with data collection and classification to help us better understand Earth and its processes. Data about clouds, land cover, mosquito habitats, and trees are collected through the GLOBE Observer app and are publicly available. The mosquito habitat data collection process entails taking geospatial pictures of mosquito larvae and classifying their species based on specific features such as siphon shape and the presence of pecten (The GLOBE Program, n.d.). However, many people are unable to classify the images because they are untrained to do so, which can result in missed data opportunities. Since the species of mosquito is an important factor in understanding and predicting disease outbreaks, the mosquito imagery classification data must be available. Our team aims to automate the classification process to reduce errors and allow for an easier data collection process.

Image classification can be achieved using GLOBE Observer Mosquito Habitat Mapper Images and Neural Networks, which are machine learning algorithms that consist of an input layer, some number of hidden layers, and a calculated output layer. The hidden layers consist of weighted nodes that initially contain a linear function. This function is then given the inputs and calculates an output and an activation function. This process repeats, with the nodes in the hidden layers determining future actions until a final output is calculated. Convolutional Neural Networks (CNNs) are Neural Networks specifically applied to Computer Vision tasks. Each Convolutional Layer in a CNN contains multiple feature detectors, which are groups of filters that detect patterns in an image based on location and a set of image matrices formed from RGB values. As more filters in a feature detector are processed, nodes in its layer are activated and their outputs are passed on to the next layer (Shin, 2021).

Current research is being done to make CNNs more accurate for image data with varied angles, backgrounds, and lighting (Liu et al., 2021). Research is also being done to increase the training speed and precision of CNNs, which would likely be useful for further investigation of our research (Bhuiya, 2021). The classification of mosquitoes based on the genus is becoming a more sought-after research goal. However, the classification of larvae specifically has received minimal attention in the scientific community. Our project focuses on the implementation of four different CNN architectures, AlexNet, LeNet-5, VGG-16 Net, and ResNet-50.

1.1 AlexNet:

AlexNet, a deep learning architecture, was proposed by Alex Knrizhevsky along with his colleagues. It consists of eight layers with over 60 million parameters. Five of them perform max pooling while the remaining three layers are fully connected and utilize ReLU activation. The use of ReLU non-linearity depicts that saturating activation functions can quicken the training speed of deep CNNs. It also consists of two normalization layers and one softmax layer. To prevent overfitting, dropout layers were employed. To ensure that the size of feature images wasn't significantly skewed, padding was introduced. AlexNet also uses GPU to enhance performance and was the first CNN to do so. It holds a batch size of 128 and performs heavy data augmentation with actions, including mirroring and cropping. There is a split in the architecture because this model was originally trained on a GTC 580 GPU that couldn't fit the network with 3 GB of memory. So, it was divided evenly between 2 GPUs. During its development, the model was also initially trained by the authors using an Imagenet dataset. *AlexNet* rose in popularity after becoming the first CNN to win the ImageNet Large Scale Visual Recognition Challenge in 2012. During the challenge, it achieved an error of 15.3%, significantly lower than that of competing models (Great Learning Team, 2020).

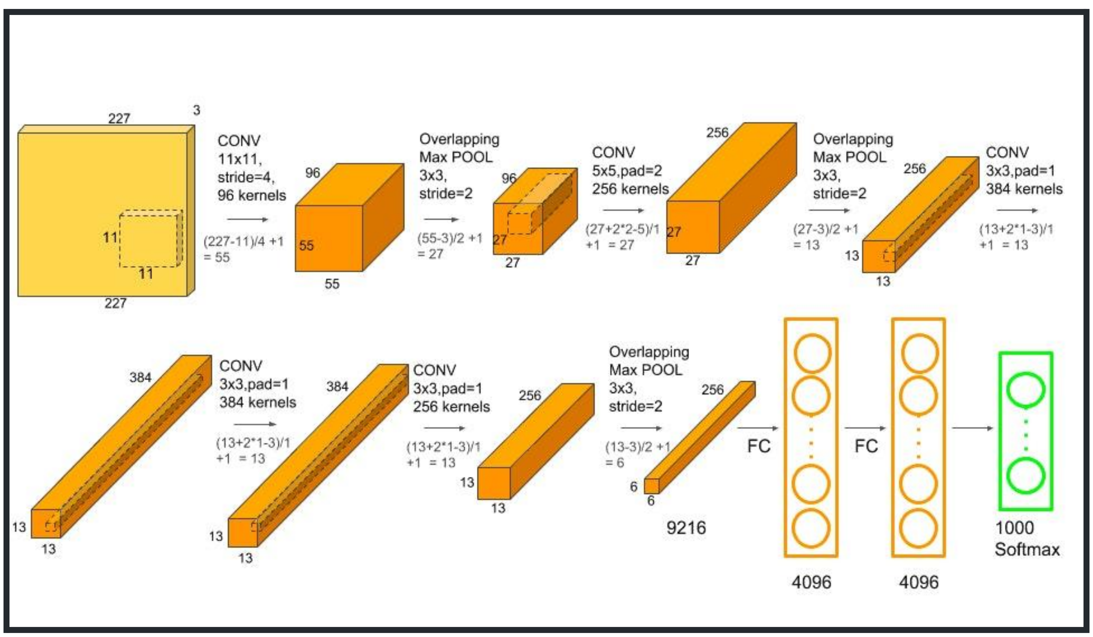

Figure 1: AlexNet, LearnOpenCV

1.2 LeNet:

LeNet was designed by Yann LeCun and associates in 1998. The author used backpropagation algorithms to train a convolutional neural network to create the model. It is utilized to recognize printed, machine, or by-hand, characters. It has 7 layers consisting of three convolutional, two sub-samplings, and two fully connected layers. A combination of convolution, pooling, and nonlinear activation functions, make up each convolutional layer. To classify and categorize images, it also uses a Softmax classifier. Further, it uses a tanh activation function and MLP as the final classifier. It has 60000 trainable parameters available and typically 32 x 32 grayscale images serve as the input to the model. There are rare connections that exist between layers, which serve to decrease the computational complexity. LeNet is one of the earliest pre-trained models and is notable for its simplicity. It was a novel architecture at its time that greatly removed the need of performing character recognition by hand (Saxena, 2021).

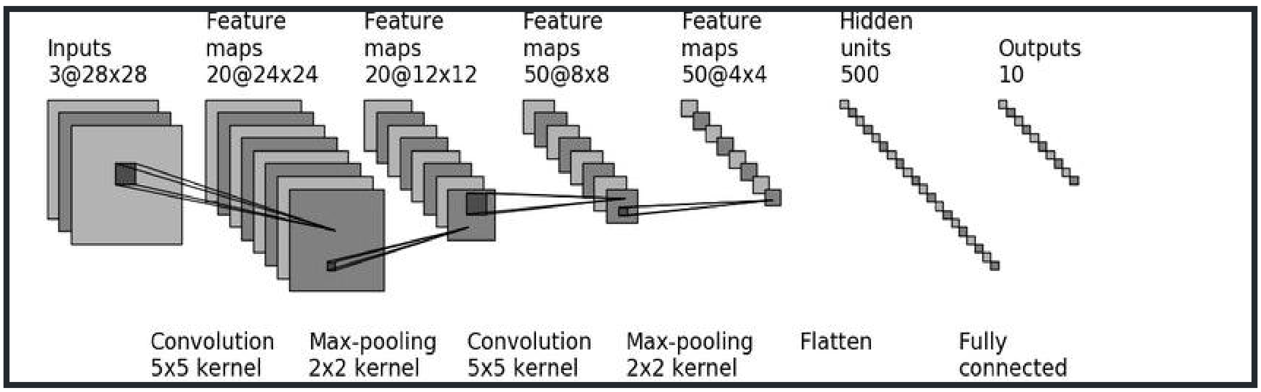

Figure 2: LeNet-5, AnalyticsSteps

1.3 VGG-16:

VGG-16 Net was invented by Simonyan and Zisserman from the Visual Geometry Group (VGG) at the University of Oxford in 2014. This convolutional neural network is 16 layers deep and was the 1st runner-up of the ImageNet Large Scale Visual Recognition Competition in 2014 in the classification task. For the competition, the pre-trained network was trained on the ImageNet ILSVRC dataset which is made up of 1000 classes split into three sets: 1.3 million images for training, 100,000 images for testing, and 50,000 images for validation. This network has learned to extract the features of an image which can be used to distinguish between objects. The network was invented to increase the accuracy of classification by increasing the depth of the CNNs. VGG Net typically takes the input of 224 x 244 RGB images and then passes them through a stack of convolutional layers, which have a fixed filter size of 3 x 3 and a stride of 1. Between the convolutional layers, there are five max pooling filters.

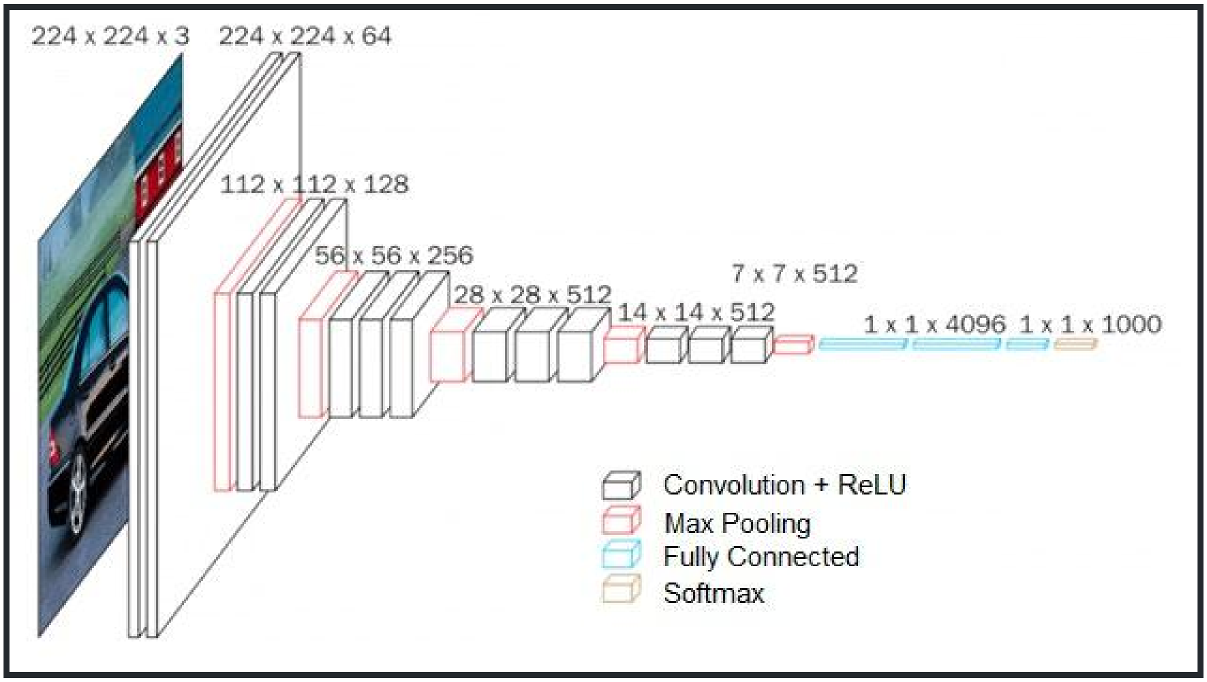

Figure 3: LeNet-5, VGG16, MyGreatLearning

1.4 ResNet:

First introduced in 2015 by Microsoft researchers, ResNets, or residual networks, were able to achieve an impressive performance at the ILSVRC 2015 performance test. Residual networks make use of identity shortcut connections that enable the flow of information across layers without attenuation, which multiple stacked non-linear transformations would cause. (Targ et al., 2016*)* This unique process allows for increased optimization by eliminating the chance of a vanishing/exploding variant. ResNet-50 is a CNN developed from the ResNet structure that is 50 layers. The pre-trained version of the network was trained on more than a million images using the ImageNet database. The image input size of the network is typically 224 x 224. ResNet was made with the aim of tackling the degradation problem, caused by stacking more layers on the network. Deep residual nets are able to improve the accuracy of models using residual blocks, essentially the concept of "skip connection" (Boesch, 2022)

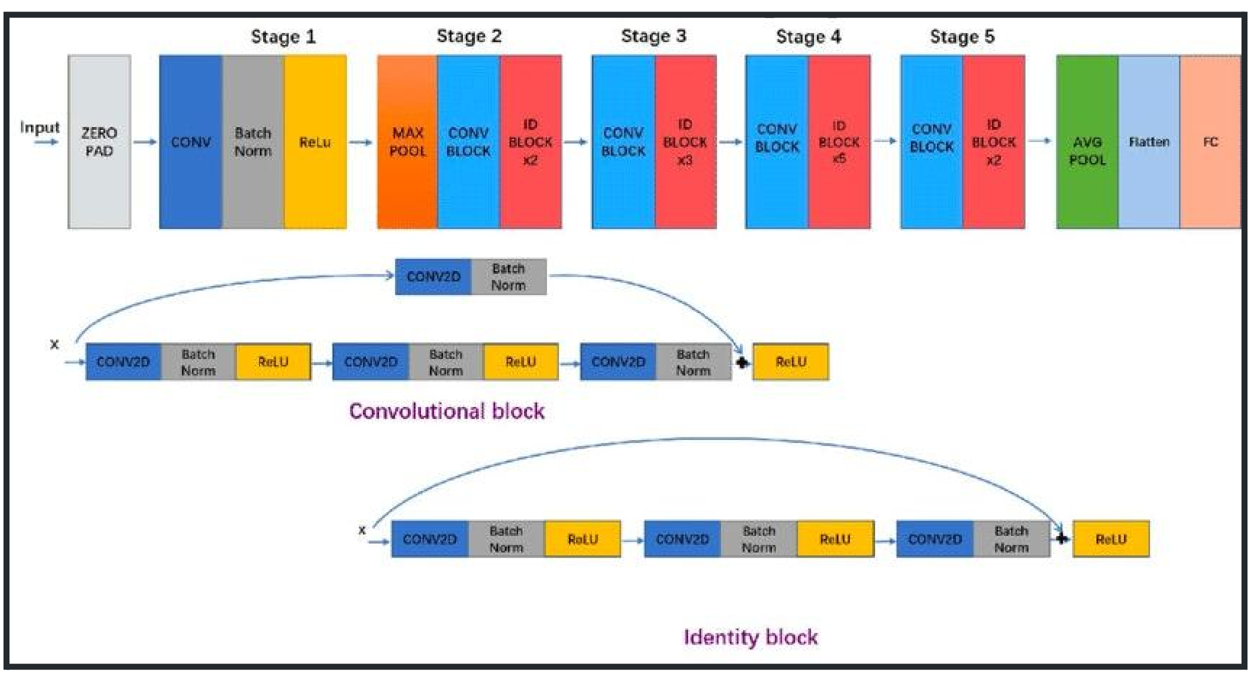

Figure 4: ResNet, GeeksForGeeks

Research Methods and Materials

To train and evaluate convolutional neural networks, a Python algorithm collected images from the GLOBE Observer Mosquito Habitat Mappers database. The data collection dates ranged from 1995 to 2022 and from study sites across the globe. After the CSV file was downloaded from GLOBE Observer's website, a Python script filtered the data down to images that were accepted by GLOBE and labeled by genus type. Then, we used the requests library to download the images through the associated URLs. We implemented the naming scheme, *imageID-genus.jpg,* to efficiently sort the images into three folders by genus using the shutil python library and string splitting logic. Lastly, the downloaded images were scanned to check if there were any invalid entries, such as an image of a mosquito habitat instead of a mosquito larva. The refined dataset consisted of 2880 Aedes larvae images, 1996 Anopheles larvae images, and 1760 Culex images—a sizable amount for the training of a representative CNN. However, only 100 images from each classification (300 images in total) were used in the training dataset to ensure each genus was equally represented and manage computing time.

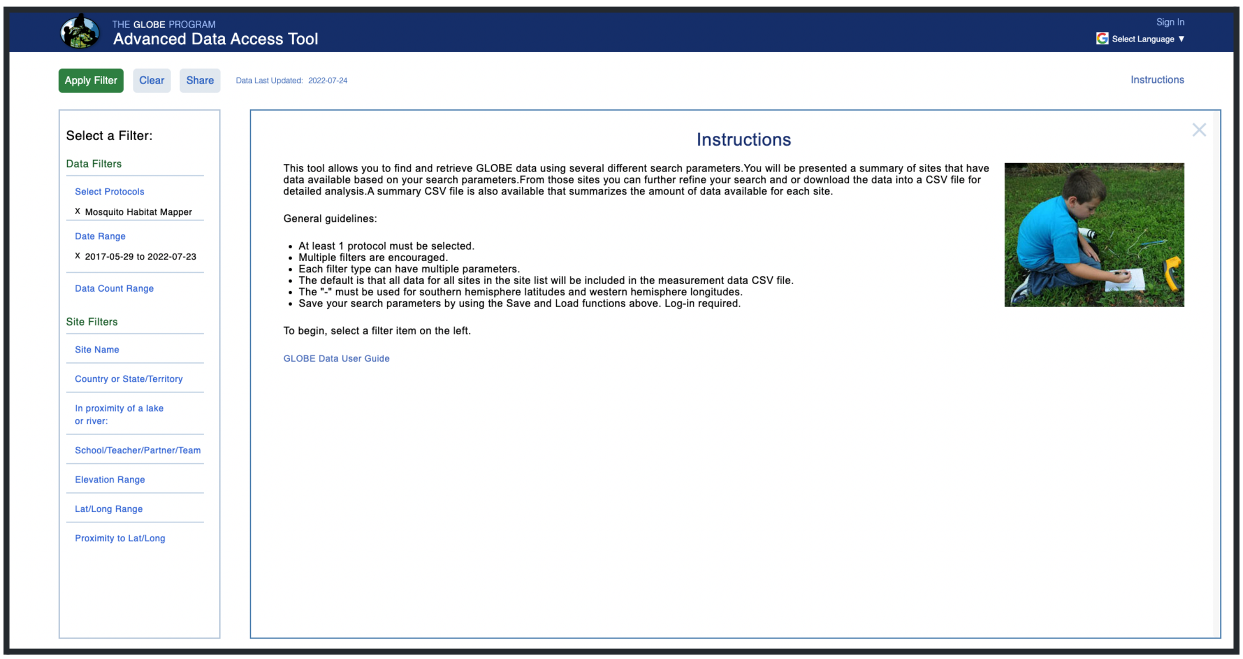

Figure 5: Print screen of the data entry web page from GLOBE

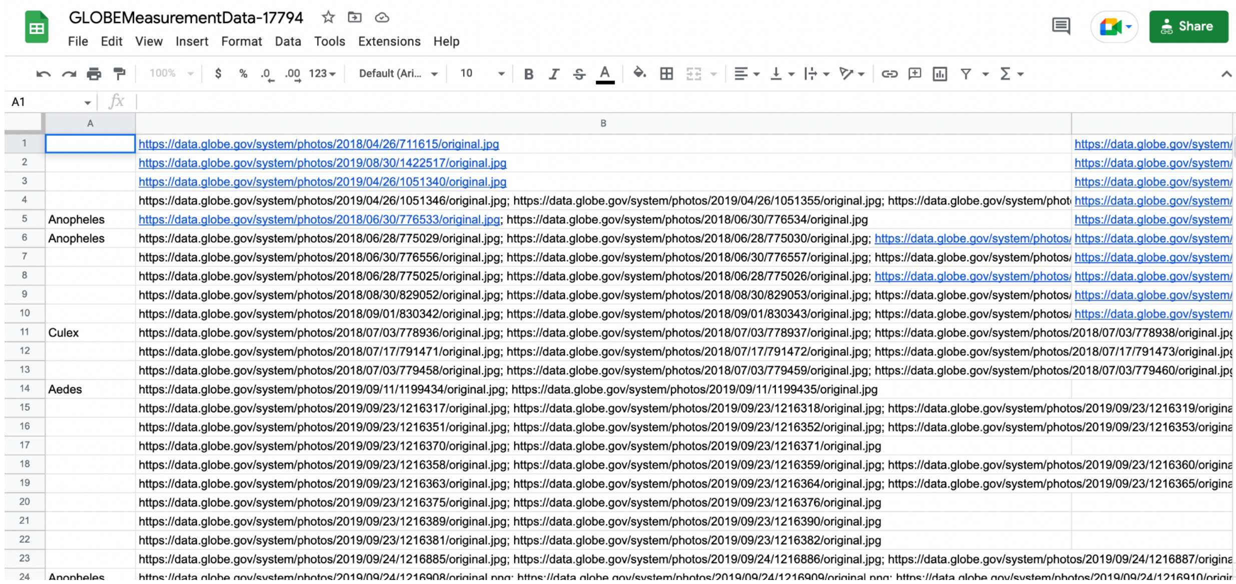

Figure 6: Image of unsorted data collection

After the data was sorted into three different genus classes, the network construction began. First, using the cv2 and NumPy Python libraries, we transformed the larvae dataset in a script that resized and vectorized the images based on RGB values. Resizing and vectorizing the dataset allowed the CNN models to interpret the images effectively. Then, the complete dataset of images was split into training and testing data using the scikit-learn library's train_test_split function. With these new datasets, our team constructed two different data generators using the Keras Python library, one for testing and one for training data. The data generators allowed multiple cores to process our dataset in real-time and fed the data into the deep learning models. (Amidi et al, 2022) Data generators also allow for image augmentation, making them useful tools for image classification projects.

After these data generators were assembled, the first CNN architecture could be built. Initially, we planned to implement transfer learning using PyTorch, which involves taking a machine learning model already used and repurposing it for a new and different task (Nelson, 2020). However, transfer learning makes it harder to exactly visualize each model's architecture in code, so we decided to use the Keras library to create a line-by-line reconstruction of each Neural Network.

AlexNet was the first architecture built. First, the model was compiled using the categorical cross-entropy loss function to predict accuracy. The model also implemented the adam optimization algorithm, which is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. (Brownlee, 2017*)* Then, the model was fit using the training generator and run through 50 epochs. Additionally, the testing generator was used as validation data. The next architecture built was LeNet-5. After the data was reshaped to fit the Keras code rendition of LeNet-5, the model was compiled and fit in the same way as AlexNet. This process of first reshaping the data, then compiling and fitting the model was also implemented for the training of the VGG-16 network. For ResNet-50, most of the same measures were taken, except instead of the Adam Optimizer Algorithm, Stochastic Gradient Descent (SGD) was used. Each model was retrained and tested ten times, with the runtime being restarted each trial.

Results

To compare CNN performances, 10 trials were conducted to gather evidence of each architecture's mean performance accuracy. After the conclusion of the trials, the mean accuracy and loss were calculated. The most accurate model was AlexNet, with an average accuracy of 89%. Next was LeNet-5 with an average of 87.3% accuracy. In third place was VGG-16 Net, with an average of 87% accuracy. The CNN with the lowest accuracy was ResNet-50, which had an average of 84% accuracy. Considering the small size of the sample dataset, these averages exceeded most of our group's initial expectations. The model with the lowest error was ResNet, which had an average loss of 0.60. Next was LeNet-5, which had an average loss of 0.87. LeNet-5 was followed closely by AlexNet, which had an average loss of 0.89. The CNN architecture with the largest loss was VGG-16 Net, which had a mean loss of 1.73

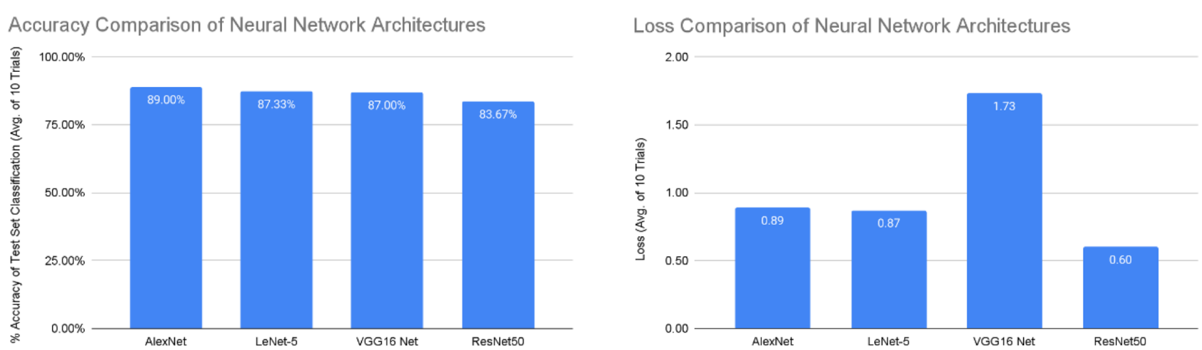

Figure 7: Accuracy comparison of neural network architectures (left) and Loss comparison of neural network architectures (right)

	AlexNet		LeNet-5		VGG16 Net		ResNet50	
	Accuracy	Loss	Accuracy	Loss	Accruacy	Loss	Accuracy	Loss
Trial 1	93.30%	0.38	85.00%	1.22	86.70%	0.38	83.30%	0.58
Trial 2	80.00%	1.84	86.70%	2.05	90.00%	0.27	86.70%	0.36
Trial 3	83.30%	1.10	81.67%	1.95	85.00%	0.52	85.00%	0.52
Trial 4	93.40%	0.03	88.33%	1.33	81.67%	0.42	85.00%	0.51
Trial 5	91.70%	0.42	95.00%	3.23	86.67%	0.65	83.33%	0.42
Trial 6	88.30%	0.37	88.33%	0.77	90.00%	0.99	81.67%	0.53
Trial 7	86.70%	1.51	88.33%	1.53	91.67%	0.56	85.00%	0.51
Trial 8	90.00%	0.53	86.70%	2.22	86.67%	0.71	83.33%	0.64
Trial 9	86.67%	1.04	88.33%	0.82	88.33%	0.91	76.67%	0.74
Trial 10	91.67%	0.39	85.00%	2.18	83.33%	0.62	86.67%	0.27
Averages	89.00%	0.89	87.33%	1.73	87.00%	0.60	83.67%	0.51
Std. Deviation	4.42%	0.58	3.44%	0.75	3.12%	0.23	2.92%	0.14

Figure 8: Accuracy and loss for each trial

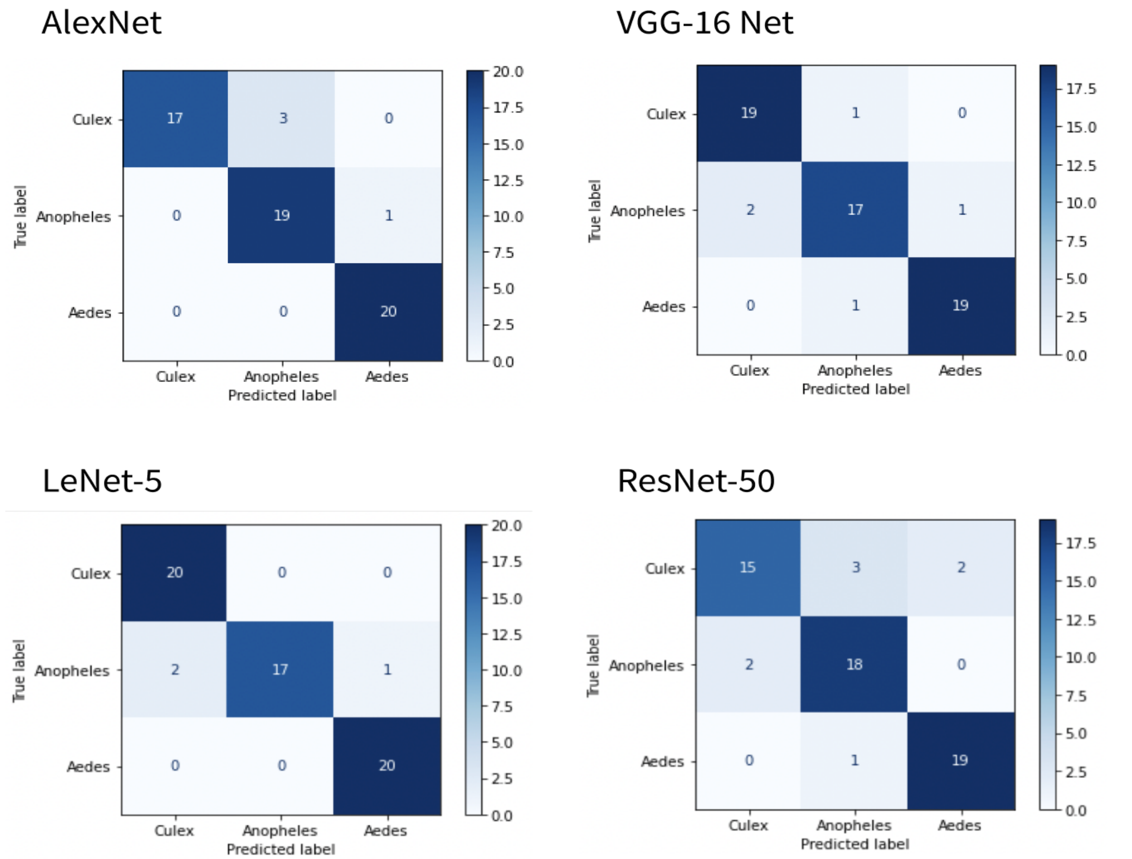

Figure 9: Confusion Matrices for each Model's Most Accurate Trial

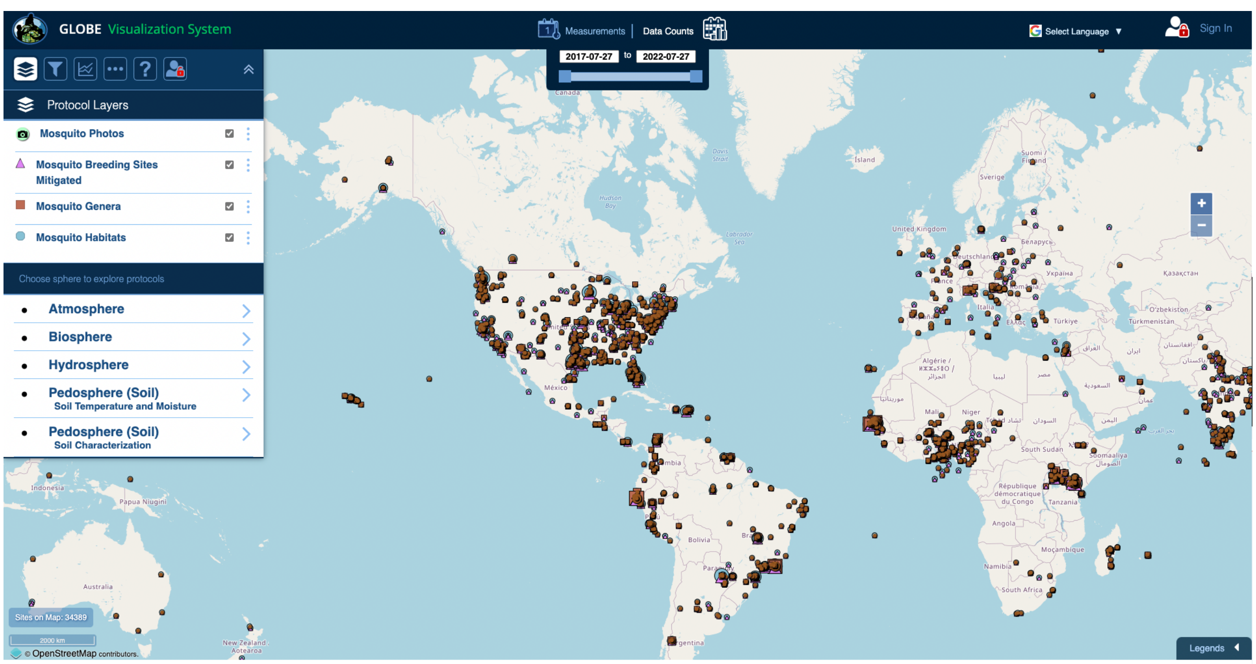

Figure 10: GLOBE Visualization Page: MHM Data

Discussion

When considering only the accuracy of each model, AlexNet was the most successful model overall. The reason for AlexNet's accuracy with this dataset is likely due to its revolutionary architecture. Alex Krizhevesky was the first to introduce Local Response Normalization (LRC) to CNNs in 2012 when he was building AlexNet. LRC can be applied in two different ways: first applying on single channel or feature maps, where an N×N patch is selected from the same feature map and normalized based on the neighborhood values. Second, LRN can be applied across the channels or feature maps (Alom et al., 2018) Krizhevesky's quest to build a broader CNN for large-scale image recognition was a massive success, and undoubtedly will be implemented in future classification research projects.

However, when looking at these accuracies, it becomes clear that there are very small differences between the models, with the highest and lowest accuracies being separated by only 6%. So, instead of accuracy, our team decided to turn to average loss to determine the most effective model overall. Between the models, ResNet50 had a much lower average loss than every other model. As mentioned earlier, ResNet50 has the unique ability to eliminate attenuation using shortcut connections, which may have been the reason for its low loss. So, despite ResNet-50 having the lowest accuracy, it actually is the most effective model, due to its outstandingly low loss. The final results of this study didn't support our hypothesis that LeNet-5 would be the most effective model since ResNet-50 was deemed the most effective model. We can now come to the reasonable conclusion that model complexity has no correlation to the amount of data given to a CNN.

Compared to other studies focused on the evaluation of different CNN architectures, our research had both differences and similarities. In a research study published in the Journal of Ambient Intelligence and Humanized Computing, where researchers attempted to develop a new land following method for autonomous vehicles (Lee et al, 2019), VGG-16 had a slightly lower validation loss than ResNet50. In our model, VGG-16 had a very large loss, while ResNet50 had a relatively low loss. This difference may be due to differences in the number of epochs in our study and Lee's: Lee used 500 epochs, while we used 50. Another study that compared CNN architectures was published in Scientific Reports, where researchers compared different deep learning architectures for the classification of chest radiograms (Bressem et al, 2020). In this investigation, ResNet-50 was one of the top-performing models, with an Area Under the Receiver Operating Characteristic Curve (AUROC) of 0.881.

One shortcoming of using Citizen Science data is that there was little way for us to check that all user inputted labels for larvae genus were 100% accurate, leading to possible errors in our prediction models. Another possible cause for the error is that we extracted only the first 100 images from the complete dataset, which could lead to location bias. Finally, due to the model's errors when being fit with the adam optimization function, our team decided to use SDG for ResNet-50. This difference in optimizer algorithms could have impacted loss and accuracy. Another important thing to note is that throughout this whole process is that we only used approximately 1/17 of the available GLOBE dataset. The reason for this is that we don't currently have the computation power to work with every image value collected. Google Colab does have a rather powerful GPU accelerator for a cloud-based environment. However, image classification with such a large dataset requires extraordinary computation power. For a better idea of how much power would be needed to run through the entire dataset, we attempted to train AlexNet using every image collected. In the end, it took 2 hours to train one epoch. After this test, we concluded that Colab simply didn't have the computation abilities to work with so much data.

Conclusion

After the testing process, ResNet-50 was determined to be the most successful CNN model for larvae classification based on genus due to its outstandingly low loss and the lack of deviation between model accuracies. In the future, an accurate mosquito larvae classification can be used in tropical and subtropical areas to prevent the spread of life-threatening vector-borne diseases. It is an essential step in understanding the viruses which are a threat to a considerable part of humanity as well as steps that need to be taken to further prevent their spread. It provides a guide for professionals to determine what actions need to be taken and will help them allocate their resources more effectively. Additionally, the insight into the accuracy of different CNNs can be implemented in different biological fields when using and in the further development of computer vision techniques.

If possible, our group intends to integrate our model into an IOS application, which could be added as a tool in the GLOBE Observer app. Also, our group will search for a Jupyter environment with higher computing capabilities, so that we can run each model with the entire dataset.

Acknowledgments

The authors would like to acknowledge the support of the 2022 Earth Explorers Team, NASA STEM Enhancement in the Earth Sciences (SEES) Virtual High School Internship program. The NASA Earth Science Education Collaborative leads Earth Explorers through an award to the Institute for Global Environmental Strategies, Arlington, VA (NASA Award NNX6AE28A). The SEES High School Summer Intern Program is led by the Texas Space Grant Consortium at the University of Texas at Austin (NASA Award NNX16AB89A), or The SEES High School Summer Intern Program is in partnership with NASA Cooperative Agreement Notice NNH15ZDA004C Award NNX16AB89A

Contribution Statement

MS, SB, and SS collected and filtered evaluation data. AB and AT searched for training and testing datasets. DS performed the data analysis and built the CNN architectures. DS, SA, SS, SB, MS, AB, AS, and AT were collaborating authors of the article AD reviewed and supported the preparation of the article. RL contributed to sections of the paper describing the GLOBE Observer citizen science app and the SEES Mosquito Mappers Summer High School Research Internship. The science and education team at the Institute for Global Environmental Strategies, led by RL, conceived of the diversity-driven, virtual outreach model for the SEES Mosquito Mappers. RL designed and directed the place-based summer internship. PN and CS served as co-mentors.

Also, we would like to share that working with Arnav Deol as a project mentor throughout this project helped us stay organized and practical in our research. Additionally, having access to resources from previous research experiences helped us better define our project timeline and, eventually, create an impactful research project.

Bibliography/Citations

About GLOBE. (n.d.). GLOBE. Retrieved July 26, 2022, from https://www.globe.gov/about

Advanced-Data Access Tool (ADAT). (n.d.). The Globe Program. Retrieved July 26, 2022, from https://www.globe.gov/globe-data/retrieve-data

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van esesn, B. C., Awwal, A. A. S., & Asari, V. K. (2018, September 12). The history began from alexnet: A comprehensive survey on deep learning approaches. *Arxiv*. Retrieved July 26, 2022, from https://arxiv.org/abs/1803.01164

Amidi, A., & Amidi, S. (n.d.). *A detailed example of how to use data generators with Keras*. Retrieved July 26, 2022, from https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly

Bhuiya, S. (2020, October 31). *Disadvantages of CNN models*. Medium. Retrieved July 26, 2022, from https://sandeep-bhuiya01.medium.com/disadvantages-of-cnn-models-95395fe9ae40

Boesch, G. (n.d.). *Deep Residual Networks (ResNet, ResNet50) – Guide in 2022*. Viso.ai. Retrieved July 26, 2022, from https://viso.ai/deep-learning/resnet-residual-neural-network/

Bressem, K. K., Adams, L. C., Erxleben, C., Hamm, B., Niehues, S. M., & Vahldiek, J. L. (2020). Comparing different deep learning architectures for classification of chest radiographs. *Scientific Reports*, *10*(1). https://doi.org/10.1038/s41598-020-70479-z

Brownlee, J. (2017, July 3). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Machine Learning Mastery. Retrieved July 26, 2022, from https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Brownlee, J. (2019, April 26). *How to Develop VGG, Inception and ResNet Modules from Scratch in Keras*. Machine Learning Mastery. Retrieved July 26, 2022, from https://machinelearningmastery.com/how-to-implement-major-architecture-innovations-for-convolutional-neural-networks/

Christine. (2021, March 24). *How Do Convolutional Neural Networks Work?* WriteMe.ai. Retrieved July 26, 2022, from https://writeme.ai/blog/how-convolutional-neural-networks-work/

Dorn, P.l., Justi, S., Krafsur, E.s., Lanzaro, G.c., Cornel, A.j., Lee, Y., & Hill, C.a. (2017). Genetics of major insect vectors. *Genetics and Evolution of Infectious Diseases*, 341-382. https://doi.org/10.1016/B978-0-12-799942-5.00015-9

Gazar, M. (2018, November 26). *LeNet-5 in 9 lines of code using Keras*. Medium. Retrieved July 26, 2022, from https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086

GLOBE Data Entry App [Computer software]. (n.d.). https://www.globe.gov/globe-data/data-entry/data-entry-app

GLOBE *Visualization Page* [Computer software]. (n.d) https://vis.globe.gov/GLOBE/

Great Learning Team. (2020, June 24). *AlexNet: The First CNN to win Image Net*. Great Learning. Retrieved July 26, 2022, from https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net/

Great Learning Team. (2021, October 1). *Introduction to VGG16 | What is VGG16?* Great Learning. Retrieved July 26, 2022, from https://www.mygreatlearning.com/blog/introduction-to-vgg16/

IBM Cloud Education. (2020, October 20). *Convolutional Neural Networks*. IBM. Retrieved July 26, 2022, from https://www.ibm.com/cloud/learn/convolutional-neural-networks

Lee, M., Han, K. Y., Yu, J., & Lee, Y.-S. (2019). A new lane following method based on deep learning for automated vehicles using surround-view images. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-019-01496-8

Liu, J., Li, P., Tang, X., Li, J., & Chen, J. (2021). Research on improved convolutional wavelet neural network. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-97195-6

Malaria Fact Sheet. (2022, July 26). World Health Organization. Retrieved July 26, 2022, from https://www.who.int/news-room/fact-sheets/detail/malaria

Malaria's Impact Worldwide. (2021, December 16). Centers for Disease Control and Prevention. Retrieved July 26, 2022, from https://www.cdc.gov/malaria/malaria_worldwide/impact.html#:~:text=Nearly%20half%20the%20world's%20population,in%20the%20WHO%20African%20Region.

Mohan, S. (2020, December 26). *Keras Implementation of ResNet-50 (Residual Networks) Architecture from Scratch*. Machine Learning Knowledge. Retrieved July 26, 2022, from https://machinelearningknowledge.ai/keras-implementation-of-resnet-50-architecture-from-scratch/

Nayak, S. (2018, June 13). *Understanding AlexNet*. LearnOpenCV. Retrieved July 26, 2022, from https://learnopencv.com/understanding-alexnet/

Nelson, D. (2020, October 17). *What is Transfer Learning?* Unite.ai. Retrieved July 26, 2022, from https://www.unite.ai/what-is-transfer-learning/

Omodior, O., Luetke, M. C., & Nelson, E. J. (2018). Mosquito-borne infectious disease, risk-perceptions, and personal protective behavior among U.S. international travelers. *Preventive Medicine Reports*, *12*, 336-342. https://doi.org/10.1016/j.pmedr.2018.10.018

Rawat, A. S. (2021, May 7). *LeNet-5 Tutorial: Architecture, Features and Importance*. AnalyticsSteps. Retrieved July 26, 2022, from https://www.analyticssteps.com/blogs/lenet-5-tutorial-architecture-features-and-importance

Residual Networks (ResNet) – Deep Learning. (2022, June 15). GeeksforGeeks. Retrieved July 26, 2022, from https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/

Riva, M. (2021, January 19). *Interpretation of Loss and Accuracy for a Machine Learning Model*. Baeldung. Retrieved July 26, 2022, from https://www.baeldung.com/cs/ml-loss-accuracy

Saxena, S. (2021, March 18). *The Architecture of Lenet-5*. Analytics Vidhya. Retrieved July 26, 2022, from https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/

Shin, T. (2020, June 2). *A Beginner-Friendly Explanation of How Neural Networks Work*. Towards Data Science. Retrieved July 26, 2022, from https://towardsdatascience.com/a-beginner-friendly-explanation-of-how-neural-networks-work-55064db60df4

Singh, K. (2019, February 17). *Training AlexNet with tips and checks on how to train CNNs: Practical CNNs in PyTorch(1)*. Medium. Retrieved July 26, 2022, from https://kushaj.medium.com/training-alexnet-with-tips-and-checks-on-how-to-train-cnns-practical-cnns-in-pytorch-1-61daa679c74a

Targ, S., Almeida, D., & Lyman, K. (2016). Resnet in Resnet: Generalizing Residual Architectures. *The International Conference on Learning Representations*. https://openreview.net/pdf/lx9l4r36gU2OVPy8Cv9g.pdf

Thakur, R. (2019, August 6). *Step by step VGG16 implementation in Keras for beginners*. Medium. Retrieved July 26, 2022, from http://Step by step VGG16 implementation in Keras for beginners

VGG Net. (2020, August 21). EverybodyWiki. Retrieved July 26, 2022, from https://en.everybodywiki.com/VGG_Net

Visualization System. (n.d.). The Globe Program. Retrieved July 26, 2022, from https://www.globe.gov/globe-data/visualize-and-retrieve-data

Badge Descriptions/Justifications

I am a collaborator: As a research team, we worked together to build the majority of the project. By splitting up the tasks, we were able to use each of our individual skills to make this research project an effective comparative analysis of multiple CNNs.

I make an impact: Our work with classifying mosquito larvae based on genus can help health professionals make more informed decisions on how to manage the spread of vector-borne diseases.

I am a STEM Professional: Not only did we work with multiple STEM professionals in the field while building our research project, but we also followed all of the correct research procedures taken by official STEM researchers.

I am a data scientist: Our group extracted, filtered, and analyzed GLOBE Mosquito Mappers data for our project using Python-built CNNs.