Using Deep Convolutional Neural Networks to Identify Bees versus Wasps

EN 250WS

Daniel Gural and Jonas Lackey

I pledge my honor that I have abided by the Stevens Honor System

1. Introduction

1.1 Overview

There have been many computational innovations in the twenty-first century but not many have had as large of an impact as neural networks have in its first couple years since creation. Though the idea of neural networks is nothing new, the application and deployment of new deep learning models have opened a whole new door to how we can look at data. One such mountain of data to explore is in the world of biology. Neural networks have been used to aid in detecting mouse movements in experiments, determine ripeness of fruits, and help identify plant patterns in emergent marsh vegetation assemblages [1][2][3]. Species identification today is as important as ever. With many species endangered of going instinct and education on these issues rising, there is a demand for a tool to help identify species. Our group aims to create a model using deep convolutional neural networks to help identify the difference between a bee or a wasp. This could be applied to help revitalize the bee population through concentrated efforts and help educate masses on the identification of these insects.

1.2 History of Neural Networks

Artificial intelligence (AI) has dramatically improved the ability to process, analyze, and find trends in data. Tasks like object detection, speech recognition, and machine translation are now possible due to the progress in the field [7]. The study of neural networks specifically has catapulted the ability of AI from something that was almost exclusively used in academia, to a tool used not only in corporate settings, but also by the everyday man. The idea for neural networks began with the ambition to model the human brain. Although neural networks success did not necessarily come from its resemblance to the structure of the human brain, the idea to build a system that mirrored its structure helped initialize the research that led to the development of neural networks today. The idea started back in 300 B.C with Aristotle's Associationism. The theory states that "the mind is a set of conceptual elements that are organized as associations between these elements" [7]. This was further developed by philosophers like Alexander Bain and William James in 1873 by relating the processes of associative memory to the distribution of neural groupings, where every possible process required a distinct set of neurons. [7] [8]. This led Donald O Hebb to develop the Hebbian learning rule in 1949 which states that the connection between two neurons is strengthened the more they fire together. There is a metabolic change in one or both neurons where the efficiency between the two is increased [7]. Around the same time, the first model of a neuron was developed by Warren McCulloch and Walter Pitts in 1943 where they showed that simple electrical circuits connecting groups of linear threshold functions could compute logical functions [8]. Although this was a major step in the research overall, the model still had no way to learn. This changed when Frank Rosenblatt developed the perceptron in 1958..The perceptron takes a set of binary inputs and multiplies each input by a continuous valued weight and thresholds the sum of these weighted inputs to output a 1 if the sum is large enough or a 0 otherwise [9]. The work of Donald Hebb was heavily influential for Rosenblatt who used Hebb's theory of strengthening neuron connections to come up with a system of weighted inputs to allow a group of perceptrons to be able to learn [8]. He later implemented the idea on custom hardware and showed that it could be used to classify simple shapes. Throughout the 70's and 80's the research into neural networks was slow, with very few people active in the field. A paper by Marvin Minsky and Seymour Papert highlighted the limitations of the models up to that point and concluded that it was dead-end research because the current models could not have hidden layers or could not emulate an XOR [7] [8] [9]. It wasn't until there was an expansion in the algorithmic toolkit and the development of a learning rule that can be applied to multi-layered neural networks with "hidden neurons". This was based off of the work of Paul Werbos who introduced the theory of backpropagation and gradient descent which allowed neural networks to adjust the weights of the inputs in hidden layers in the neural network. These theories served as the foundation for neural network research up to the present. Many new types of neural networks were developed like recurrent neural nets or convolutional neural nets [7] [8]. With the new research came new issues. Backpropagation stopped working as the networks became more complicated until around 2006. At this point researchers found that backpropagation can work with the right activation functions and the discovery of the sigmoid. In many ways neural networks up until the 2000's were ahead of their time to the point where only recently can the computational power of the computers match what is needed to properly train neural networks [9]. Now large companies have successfully created working neural networks such as AlexNet GoogLeNet, VGGNet and ResNet. The common theme among these successful neural networks is that they are all convolutional neural networks and all of them are well suited for image classification. They are now widely used in many fields that require data analytics.

1.3 Neural Networks in Biology

Biology is a field also adapting to the neural network explosion. Scientists over the years have collected massive amounts of data to study and analyze. One interesting set of data that is most expansive in biology is photography. The model we are specifically studying, convolutional neural networks has been applied in many facets of biology. The most prevalent being species identification. Many use cases can be developed for species identification for both commercial and academic purposes. Researchers showed how using pictures of ocean sediment to identify plankton species to serve as paleo-enviromental indicators can be done much more accurately by a machine now when paired up against experts [4]. Another use is identifying animals in camera-trap images. Hunters or researchers could use this information to help identify which species are in the area. However, as pointed out in [5], there are limitations and drawbacks to deep neural networks. For example, quality and lighting in the picture play a large role. If pictures are taken by an automatic sensor in the camera, pictures could have lens flares, poor focus, or tearing if the animal was moving too fast. When training the dataset, these hindrances can greatly affect the accuracy of the model and lead to misidentifications. Identifying specifically insects with neural networks can be seen gaining momentum around 2008 when Banerjee et al. published results on identifying mosquito species through neural networks. The model was still behind modern models in the way that it still used perceptrons in a binary fashion and did not have pooling on the images like previously. The results were good for the time but left much room for improvement. Overfitting was a large issue in the past and can be seen in the article previously stated [6]. More modern papers can be seen using convolutional neural networks to better solve this problem. In [7], the authors identify bee species using wing patterns from up close pictures. The models post amazing results with top-1 accuracies reaching near 94% and top-5 reaching 99.7%. The researchers surveyed several models and posted accuracies from each. In our project, the group would like to include more common pictures of bees as wing pictures can be difficult to obtain by the normal photographer.

2. Methods

2.1 DataSet

The dataset that was chosen was called Bees vs Wasps and was found on the website Kaggle. The dataset contained images of bees, wasps, various insects, and other random images. The dataset was broken down into two categories, high quality photos and any quality photos. The high quality dataset contained 5628 pictures for training and 1047 pictures for validation. The any quality dataset contained 8452 pictures for training and 2112 pictures for validation. The main difference between the two categories is that the high quality dataset contained pictures that were both high definition and had the target in the middle of the picture. The any quality data set had blurry images, high quality images, and images with the target in any position on the picture.

Figure 1 shows examples of the high quality dataset while Figure 2 shows examples of the any quality dataset.

Figure 1: High Quality Dataset

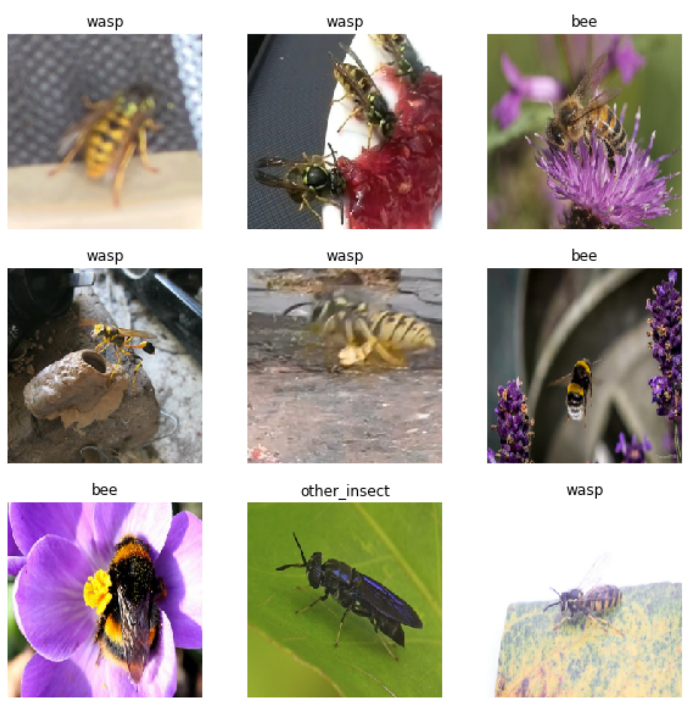

Figure 2: Any Quality Dataset

2.2 Implementation

Four different types of neural networks were trained to recognize the difference between bee's, wasp's and other miscellaneous insects with varying degrees of success. The first neural network that was trained and tested was a convolutional neural network. This CNN was constructed using Python 3 in Jupyter Notebooks because of its accessibility to data visualization tools and clean environment. While creating the CNN, the Tensorflow library was used. More specifically, the Keras library contained within Tensorflow was used. This allowed for rapid prototyping and training of the CNN. This CNN contained 10 different layers that consisted of a rescaling layer, multiple 2D convolutional layers, multiple 2D max pooling layers, a flattening layer, and two dense layers. The rescaling layer is used in the beginning of the neural network to make sure that the image statistics are within the proper ranges for the latter layers. The convolutional layers summarize the presence of specific features in an input range by applying learned filters to input images to create feature maps [10]. The problem with this is that they are often overly sensitive to the location of the data within the pictures. To rectify this, the max pooling layer down samples the feature maps that were previously created to reduce its dimensionality and allow for more assumptions to be made about specific regions. This combination of a convolutional layer followed by a max pooling layer can be repeated multiple times to create a more robust input for the neural network. After the convolutional and max pooling layers, a flatten layer converts all the data into a 1 dimensional array for the final layers to work with. The dense layers are the actual neural networks that will detect whether the image is a wasp or a bee or another insect. They will feed all the inputs from one neuron to the next with specific weights between the neurons. Figure 3 shows the order of the layers that were used for the first CNN of this project. The high quality dataset was used on this particular CNN.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_1 (Rescaling)      (None, 180, 180, 3)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 180, 180, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 90, 90, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 90, 90, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 45, 45, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 45, 45, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 22, 22, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 30976)             0
_________________________________________________________________
dense (Dense)                (None, 128)               3965056
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
=================================================================
Total params: 3,989,027
Trainable params: 3,989,027
Non-trainable params: 0
```

Figure 3: CNN Layers

There were three professionally trained neural networks that were used in addition to the CNN that was explained previously. The three that were used for this project were Resnet, InceptionV3, and Mobile net. Each of these neural networks were trained with the any quality dataset because of the larger sample size it possessed when compared to the high quality dataset. Each of these networks have their own benefits and drawbacks but will in general perform better than any ameuateur neural network. The beauty of transfer learning, or taking these pretrained models and applying to your a different problem, is that they have been fine tuned one the best datasets to be the best model they can be. By just chopping off the last layer of these models, we can harvest the elite feature extraction that these models provide. The ResNet model we used had over 150 layers and probably took hours if not days to train on even the best computers. However, with our dataset and the pretrained model, it only took about 15 minutes to train to our problem.

3. Results

3.1. High Quality vs Any Quality

To test the effect the quality has on the accuracy of the model, we tested both the high and any quality datasets on our simple CNN model. We extracted model accuracy as well as cross entropy loss as well. The models were tested on a 85/15 test validation split that was randomly split in the data set to ensure no bias between models. Both models were trained in five epochs with batches of 32 images at once. Below are the received results.

Figure 4: High Quality Data

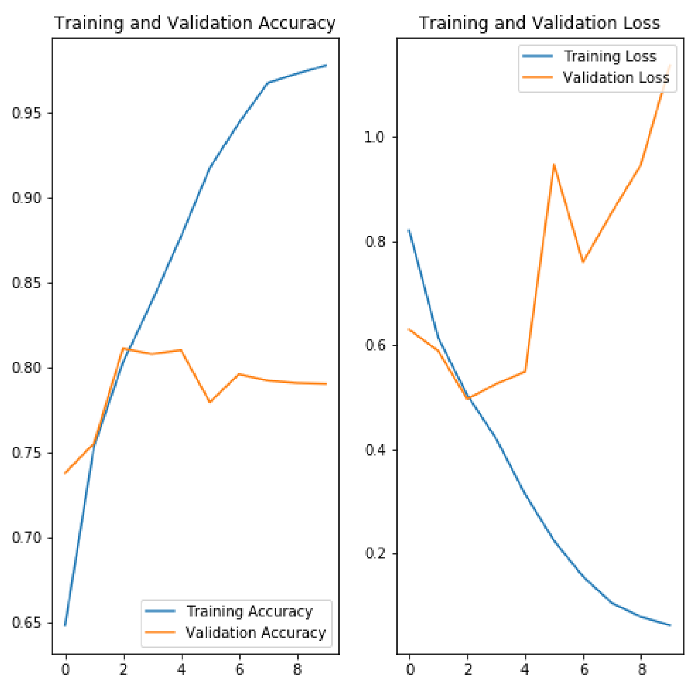

Figure 5: Any Quality Data

It can be observed that the any quality datasets was able to perform better than high quality. The high quality dataset topped out around 75% while the any quality topped out around 83%. This can be attributed to the increase in the size of the dataset, giving the data set more information to learn from and be less prone to overfitting. Overfitting most likely occurred in these models based on the curves shown. When the blue line tends to have very high accuracy and low loss but the orange line trends in the opposite direction, this is a clear indication that the model is overfitting the training data and is no longer accurately predicting unknown data. These problems can be solved using more data, regularization techniques such as dynamically lowering the learning rate, or using more complex models.

3.2 Transfer Learning Models

Each of the pretrained models had a recommended amount of epochs to train for. Mobile net and Resnet were trained for 5 epochs each while Inceptionv3 was trained for only two epochs. The batch size was kept constant from the previous models at 32. After observing the previous results, we decided to train the transfer models on the any quality dataset only as it performed better than the rest. Training the transfer learning models was much more complex and took several hours to complete on some of them. After receiving the results, the accuracies found were 92.38% for ResNet, 92.57% for MobileNet, and 91.1% for InceptionV3. Below are the graphs for learning.

Figure 6: ResNet Training

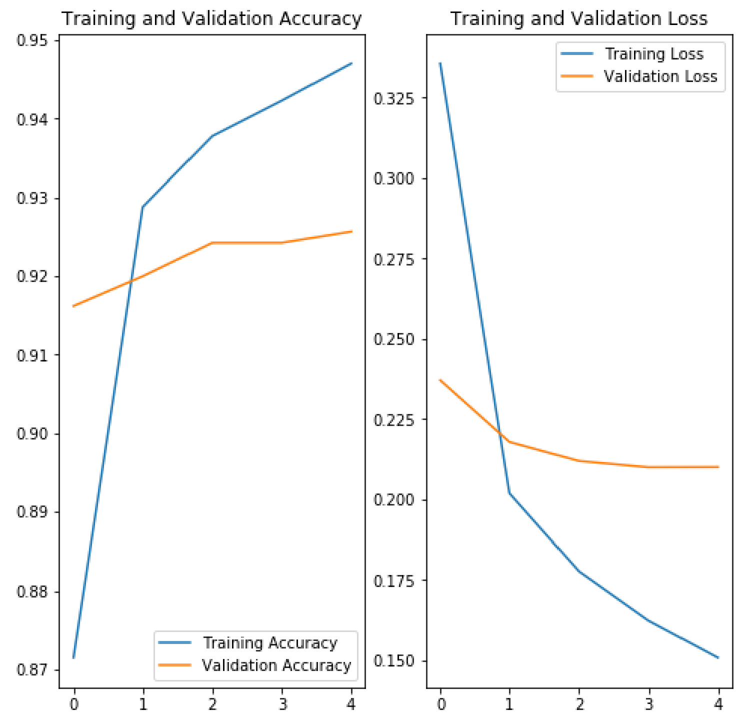

Figure 7: Mobile Net

Figure 8: InceptionV3

The graphs showed much better curves than the previous CNN model we created. It can be observed that the curves for accuracy both tend up and curves for loss both tend down. There was probably little overfitting or training error that occurred and around 93% is the best attainable accuracy with the data we currently have.

4. Conclusions

Our highest attained accuracy at predicting bees vs wasps vs other insects was 92.57% from the MobileNet model. Several other key insights were extracted from our experimentation. The first was that more data tends to help the model, even if it isn't of the highest quality. Our data supports that with more training data, the accuracy will rise. Next was that the transfer learning models will significantly outperform the standard CNN models. The idea that transfer learning surpasses just normal CNN is a well supported idea in academia. However, we were surprised at just how large the margin was. Without any additional preprocessing or regularization techniques, the models were able to remove the overfitting problem and jump accuracy up by over 15%. Finally, after surveying the popular transfer learning models, we were able to discover that MobileNet works best for our dataset and performed the best. This model could theoretically be used in species identification with minimal errors occurring. Future work could be done to identify the misclassifications the model had to see if there was a large bias and to correct that bias. Also, the model could be reasonably expanded to identify other crucial insects that could be used for education or for farming as well. With neural networks, species identification on a grand scale is a problem that becomes more realistic with each passing day.

Citations

[1] Geuther, B.Q., Deats, S.P., Fox, K.J. *et al.* Robust mouse tracking in complex environments using neural networks. *Commun Biol* 2, 124 (2019). https://doi.org/10.1038/s42003-019-0362-1

[2] Hossein Azarmdel, Ahmad Jahanbakhshi, Seyed Saeid Mohtasebi, Alfredo Rosado Muñoz, Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM), *Postharvest Biology and Technology,*Volume 166, (2020). https://doi.org/10.1016/j.postharvbio.2020.111201.

[3] Snedden GA, Feilhauer H. Patterning emergent marsh vegetation assemblages in coastal Louisiana, USA, with unsupervised artificial neural networks. *Applied Vegetation Science*. (2019). doi:10.1111/avsc.12425

[4] R. Mitra, T.M. Marchitto, Q. Ge, B. Zhong, B. Kanakiya, M.S. Cook, J.S. Fehrenbacher, J.D. Ortiz, A. Tripati, E. Lobaton, Automated species-level identification of planktic foraminifera using convolutional neural networks, with comparison to human performance, *Marine Micropaleontology*, Volume 147, (2019), Pages 16-24, https://doi.org/10.1016/j.marmicro.2019.01.005.

[5] Alexander Gomez Villa, Augusto Salazar, Francisco Vargas, Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, *Ecological Informatics*, Volume 41, (2017), Pages 24-32, https://doi.org/10.1016/j.ecoinf.2017.07.004.

[6] Amit Kumar Banerjee, K. Kiran, U.S.N. Murty, Ch. Venkateswarlu, Classification and identification of mosquito species using artificial neural networks, *Computational Biology and Chemistry*, Volume 32, Issue 6, (2008), Pages 442-447, https://doi.org/10.1016/j.compbiolchem.2008.07.020.

[7] Bishop, M. (2014) History and Philosophy of Neural Networks. In: Ishibuchi, H. (ed). Computational Intelligence. Eolss Publishers, Paris,

[8] Wang, H., & Raj, B. (2017). On the Origin of Deep Learning On the Origin of Deep Learning.

[9] Kurenkov, A. (2020). A Brief History of Neural Nets and Deep Learning. Retrieved October 10, 2020, from http://www.skynettoday.com/overviews/neural-net-history

[10] Brownlee, Jason. “A Gentle Introduction to Pooling Layers for Convolutional Neural Networks.” *Machine Learning Mastery*, Machine Learning Mastery, 5 July 2019, machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.