ELEN 199: Directed Research

**Classification of Diseased Crop Images Using AlexNet**

Maximilian Khan mjkhan@scu.edu

**I. Abstract**

In vast fields filled with numerous crops, it is intensely laborious for farmers and workers to observe each crop and note both which ones are diseased, and are diseased in what way. It is also noteworthy that many farmers across the world have access to laptops that can perform artificial intelligence model training based on images that they can collect from their own farms, but that their computational capacities are limited due to financial restrictions. The focus of this project is to develop a trained AlexNet model that can accurately classify whether or not a crop is diseased and in what way, and do so under a low number of epochs and batches to simulate the computational restrictions of many international farmers. After modifying the number of epochs between 2 and 4, and the learning rates between 0.001, 0.0005, and 0.0001, our results showed that after running each trial variant three times, the model trained under 4 epochs with a learning rate of 0.001 was most accurate at 98.55%.

**II. Motivation**

It is important that our global food supply chain is constantly improved and optimized such that we can feed the most amount of people possible, and slowly take steps toward eliminating world hunger. Though many farmers in say India or Uganda are limited by their financial income, their overall income is increasing due to increased demands for global imports to say China or the United States. Some are able to buy webcams, 360-degree cameras, or even drones, and as such, them being able to use software that can improve the overall state of their farms through images is very powerful.

**III. Dataset**

For this project, we used a dataset containing over 87k images of both healthy and diseased crop leaves that were organized in 38 different classes, available on Kaggle. Scrolling through all of the images and their categories, I found that none were mislabeled. An initial issue was that the images were of size 235x235 where AlexNet expected sizes of 227x227. However, after importing the data into MATLAB, I created a transformation function that reduced all of the images to 227x227. For training and testing, the dataset was split between 80% train and 20% test.

|  |  |
| --- | --- |
|  |  |

*Figure 1: Healthy apple and blueberry leaves example images*

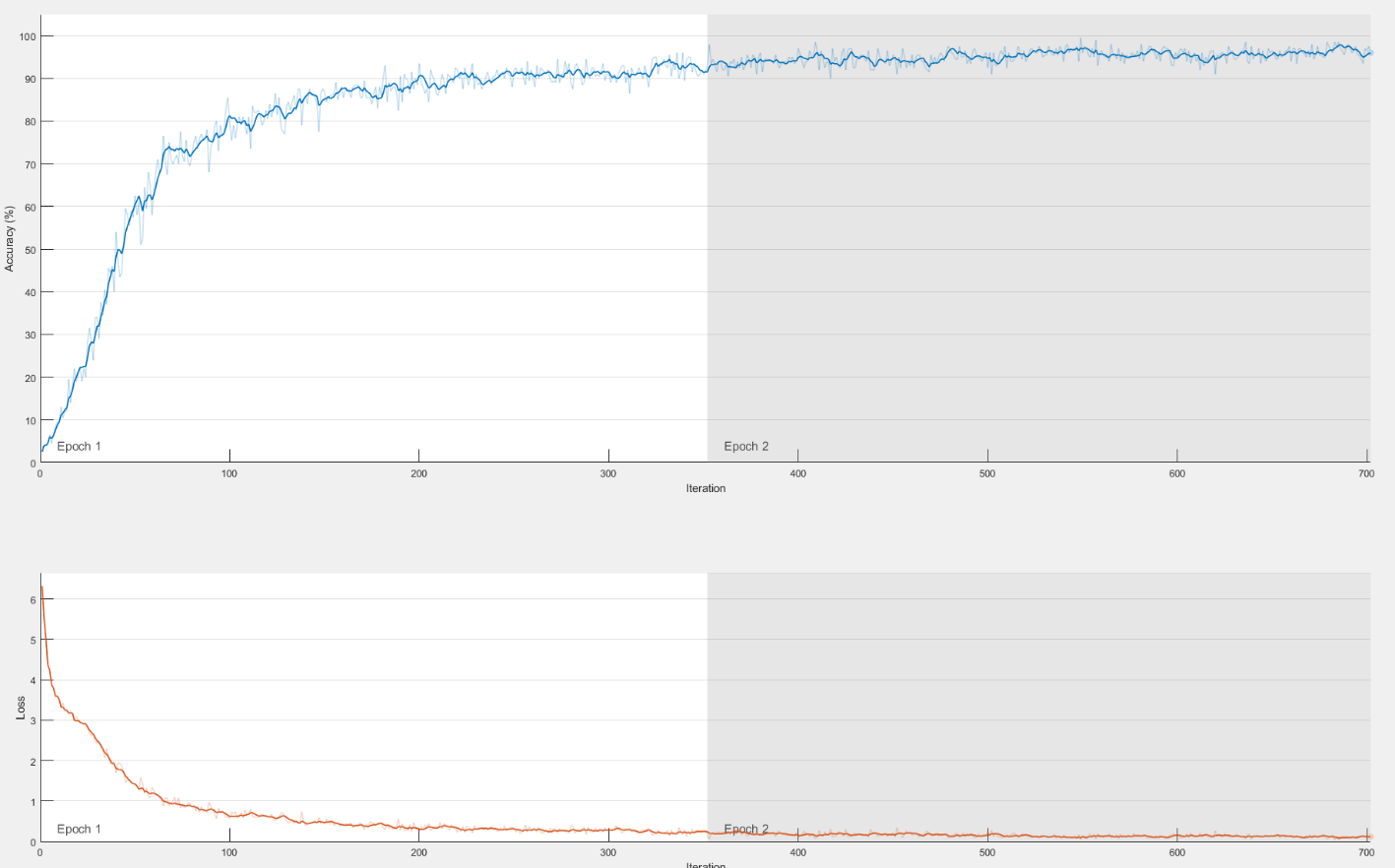
The 38 different classes included:

|  |  |
| --- | --- |
| Apple Scab | Pepper (Bell) Healthy |
| Apple Black Rot | Potato Early Blight |
| Apple Cedar Rust | Potato Healthy |
| Apple Healthy | Potato Late Blight |
| Blueberry Healthy | Raspberry Healthy |
| Cherry Healthy | Soybean Healthy |
| Cherry Powdery Mildew | Squash Powdery Mildew |
| Corn Gray Leaf Spot | Strawberry Healthy |
| Corn Common Rust | Strawberry Leaf Scorch |
| Corn Healthy | Tomato Bacterial Spot |
| Corn Northern Leaf Blight | Tomato Early Blight |
| Grape Black Rot | Tomato Healthy |
| Grape Black Measles | Tomato Late Blight |
| Grape Healthy | Tomato Leaf Mold |
| Grape Leaf Blight | Tomato Septoria Leaf Spot |
| Orange Haunglongbing | Tomato Spider Mites |
| Peach Bacterial Spot | Tomato Target Spot |
| Peach Healthy | Tomato Mosaic Virus |
| Pepper (Bell) Bacterial Spot | Tomato Yellow Leaf Curl Virus |

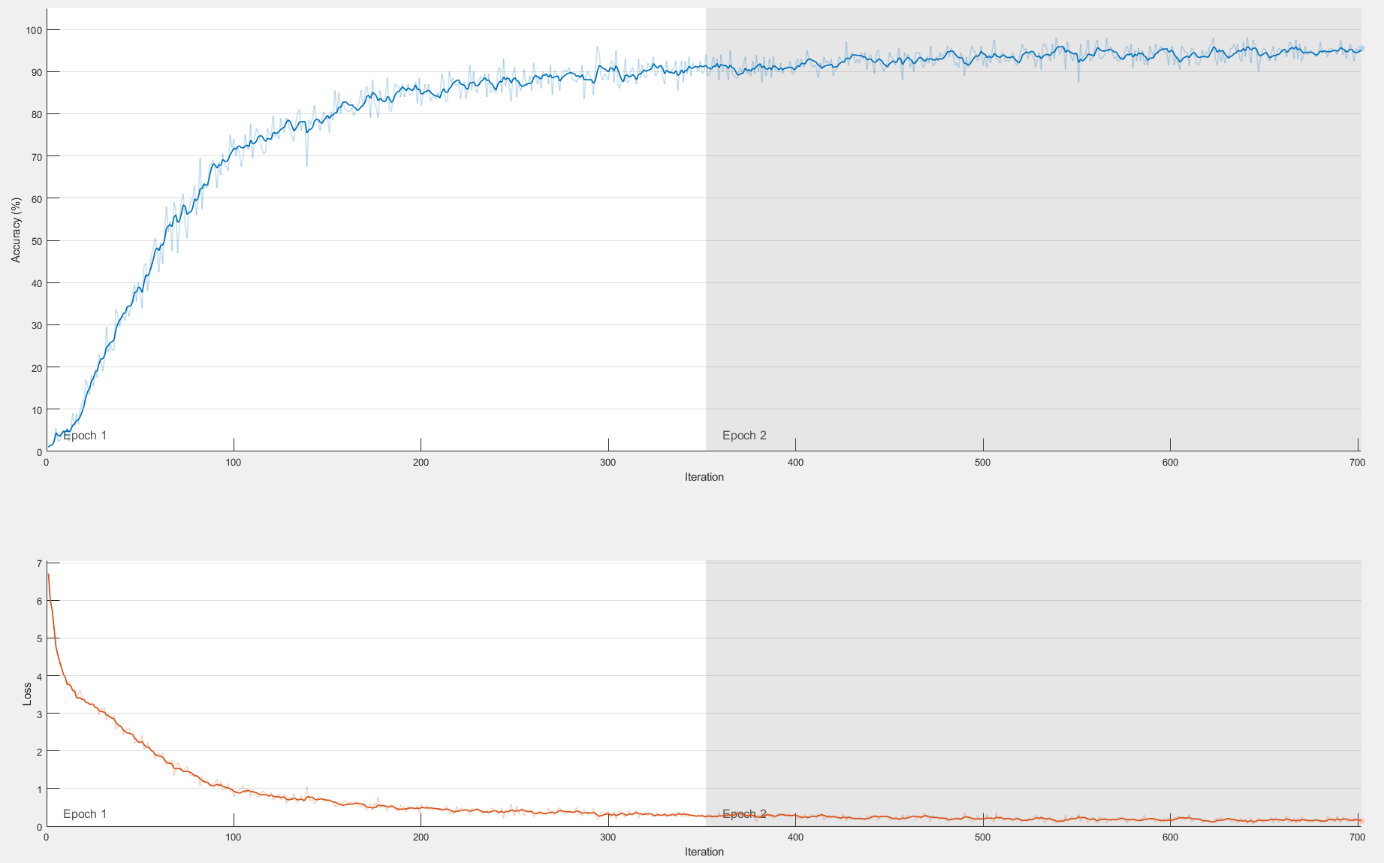
**IV. Methodology**

To simulate the computational restrictions of farmers, I limited the duration of AlexNet training. Using a constant batch-size of 200, I varied the number of epochs between 2 and 4, and the learning rates between 0.001, 0.0005, 0.0001. I averaged three tests of each variant, and the data to draw a conclusion. Below are examples from each test.

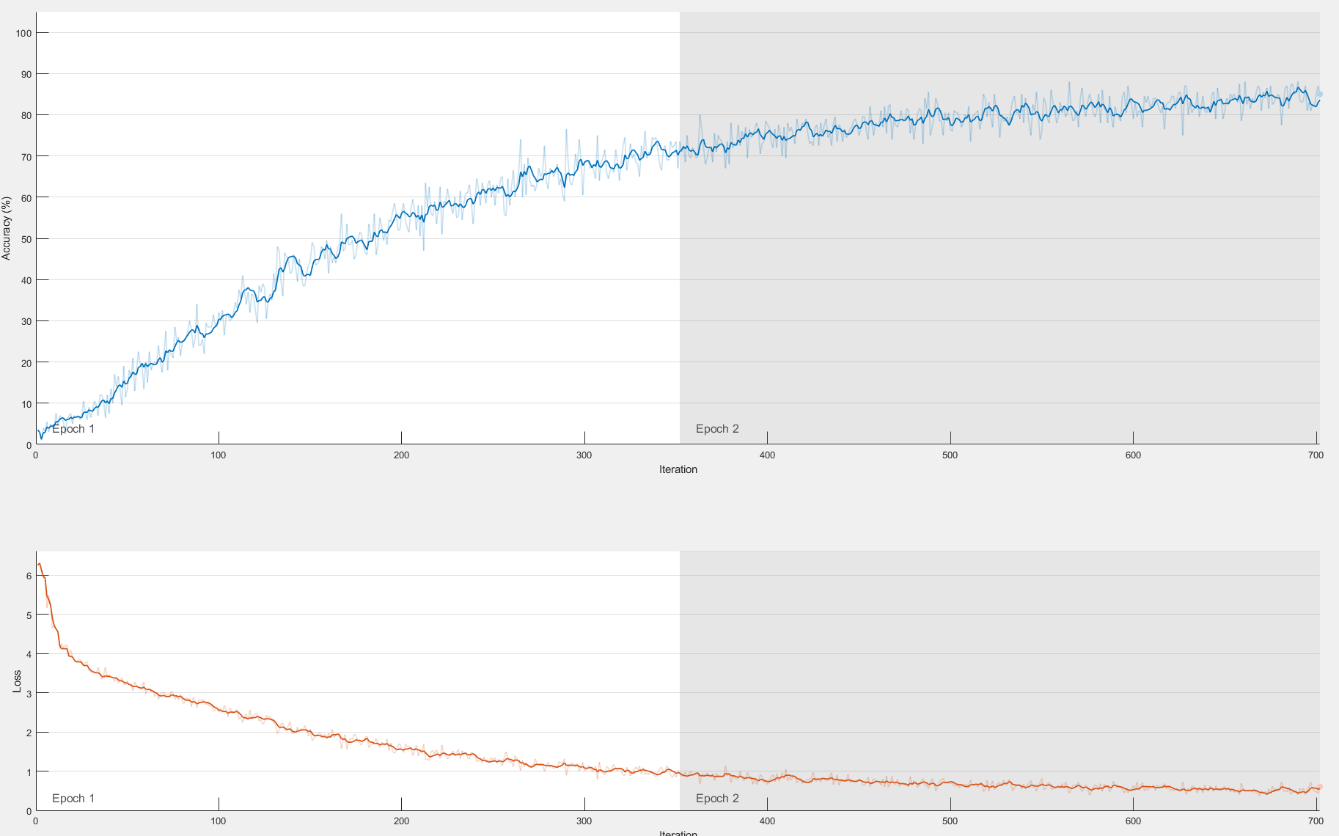
2 Epochs:



*Figure 2: Graph of training accuracy, 2 epochs @ LR = 0.001*



*Figure 3: Graph of training accuracy, 2 epochs @ LR = 0.0005*



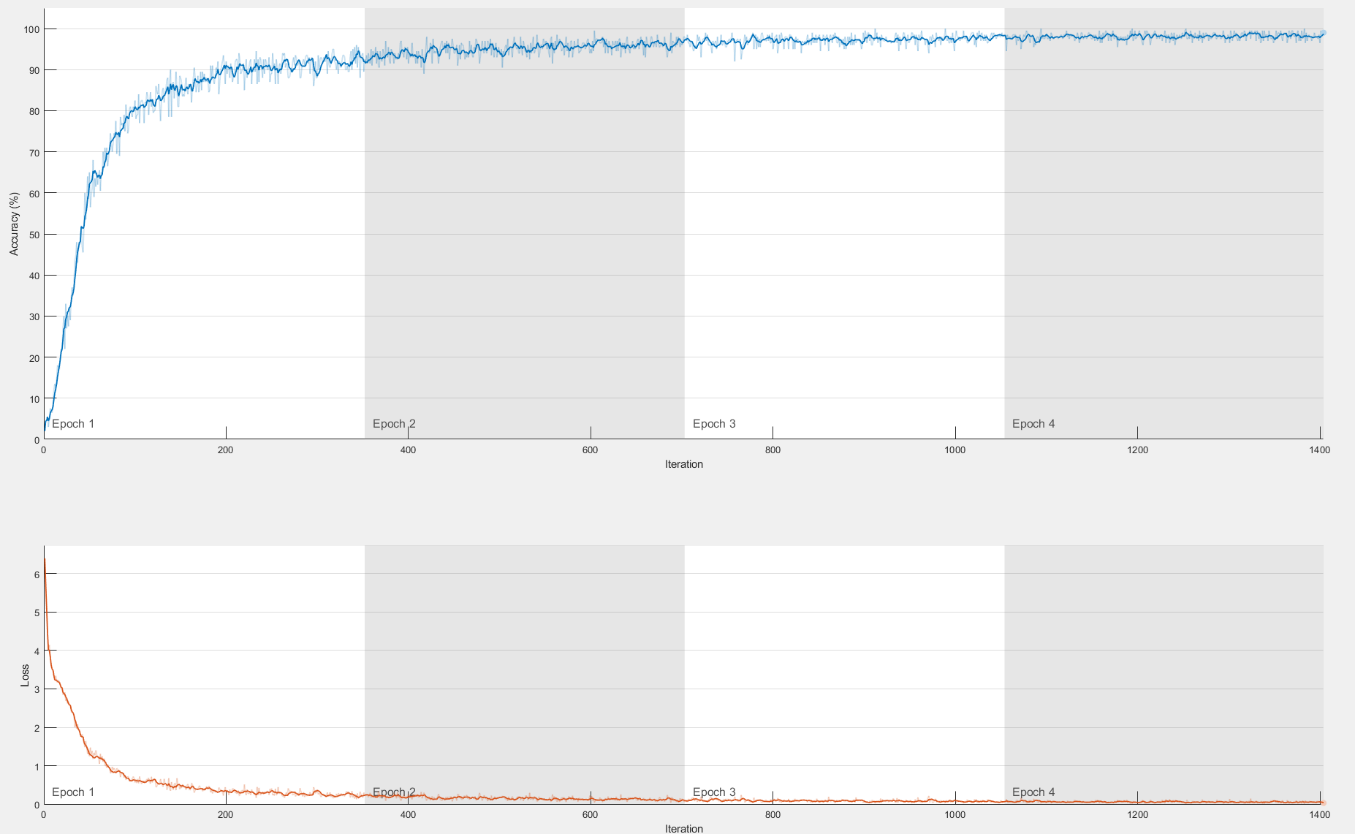
*Figure 4: Graph of training accuracy, 2 epochs @ LR = 0.0001*

As we can observe by the graphs, the model that initially learned the fastest was the one that used the learning rate of 0.001, and the one that learned the slowest had a learning rate of 0.0001. By general intuition, this makes sense. As smaller learning rates would make our model slower to converge to a higher accuracy over a lower number of epochs compared to models trained over a higher number of epochs, below is data that justifies this claim.

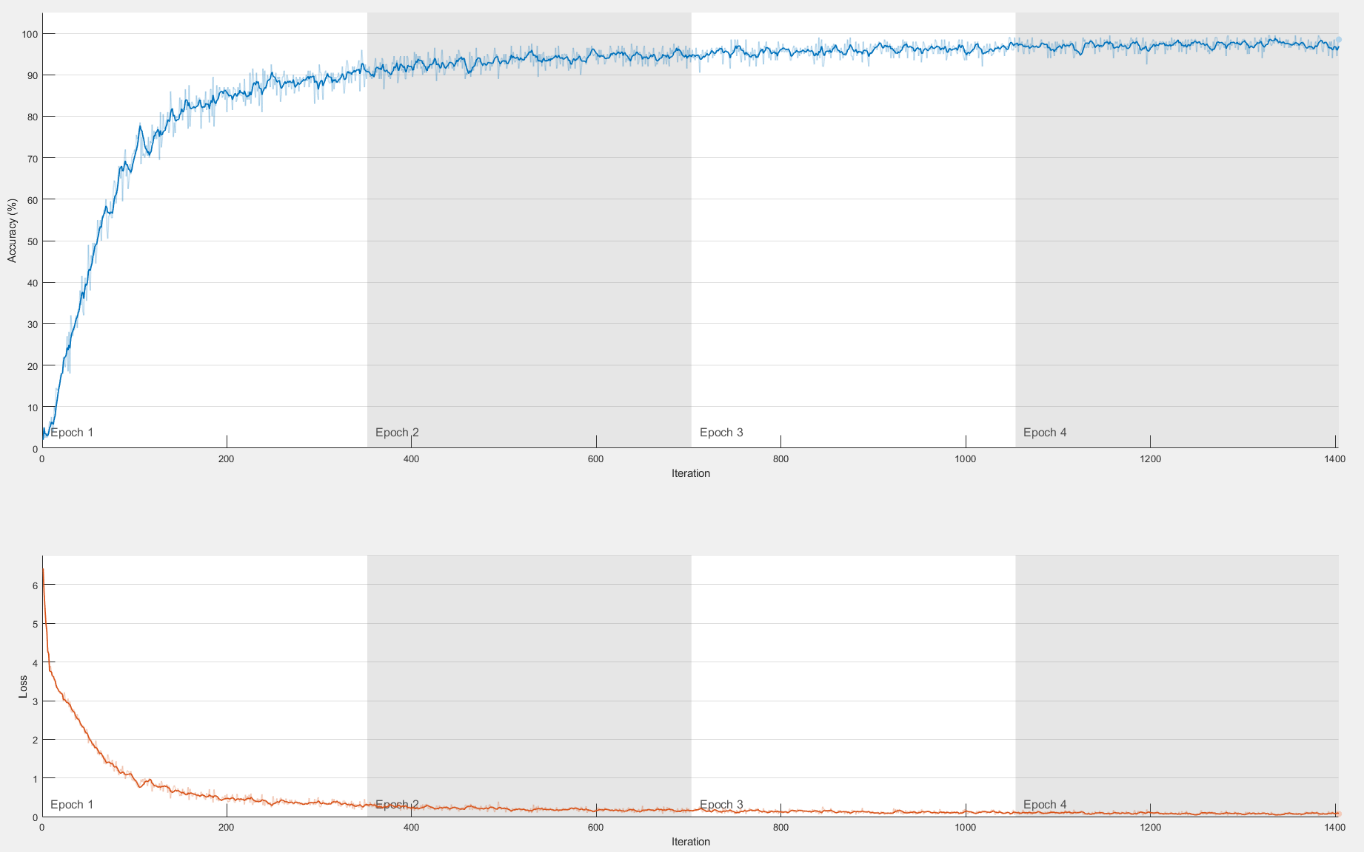
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Iteration | Accuracy 0.001 | Accuracy 0.0005 | Accuracy 0.0001 |
| 1 | 1 | 2.50% | 1.00% | 3.50% |
| 1 | 50 | 62.50% | 37.00% | 13.50% |
| 1 | 100 | 79.50% | 74.00% | 32.00% |
| 1 | 150 | 87.00% | 79.00% | 44.50% |
| 1 | 200 | 93.50% | 82.00% | 56.50% |
| 1 | 250 | 92.00% | 91.00% | 63.00% |
| 1 | 300 | 91.00% | 88.00% | 68.50% |
| 1 | 350 | 91.00% | 89.50% | 73.50% |
| 2 | 400 | 97.00% | 92.00% | 77.50% |
| 2 | 450 | 95.50% | 94.50% | 75.00% |
| 2 | 500 | 92.50% | 91.50% | 76.50% |
| 2 | 550 | 96.00% | 94.50% | 76.00% |
| 2 | 600 | 95.00% | 95.00% | 79.00% |
| 2 | 650 | 96.60% | 95.50% | 81.00% |
| 2 | 700 | 96.50% | 94.50% | 85.00% |
| 2 | 702 | 95.00% | 95.50% | 85.00% |

*Figure 5: Table of accuracy scores compared against other learning rates over epochs*

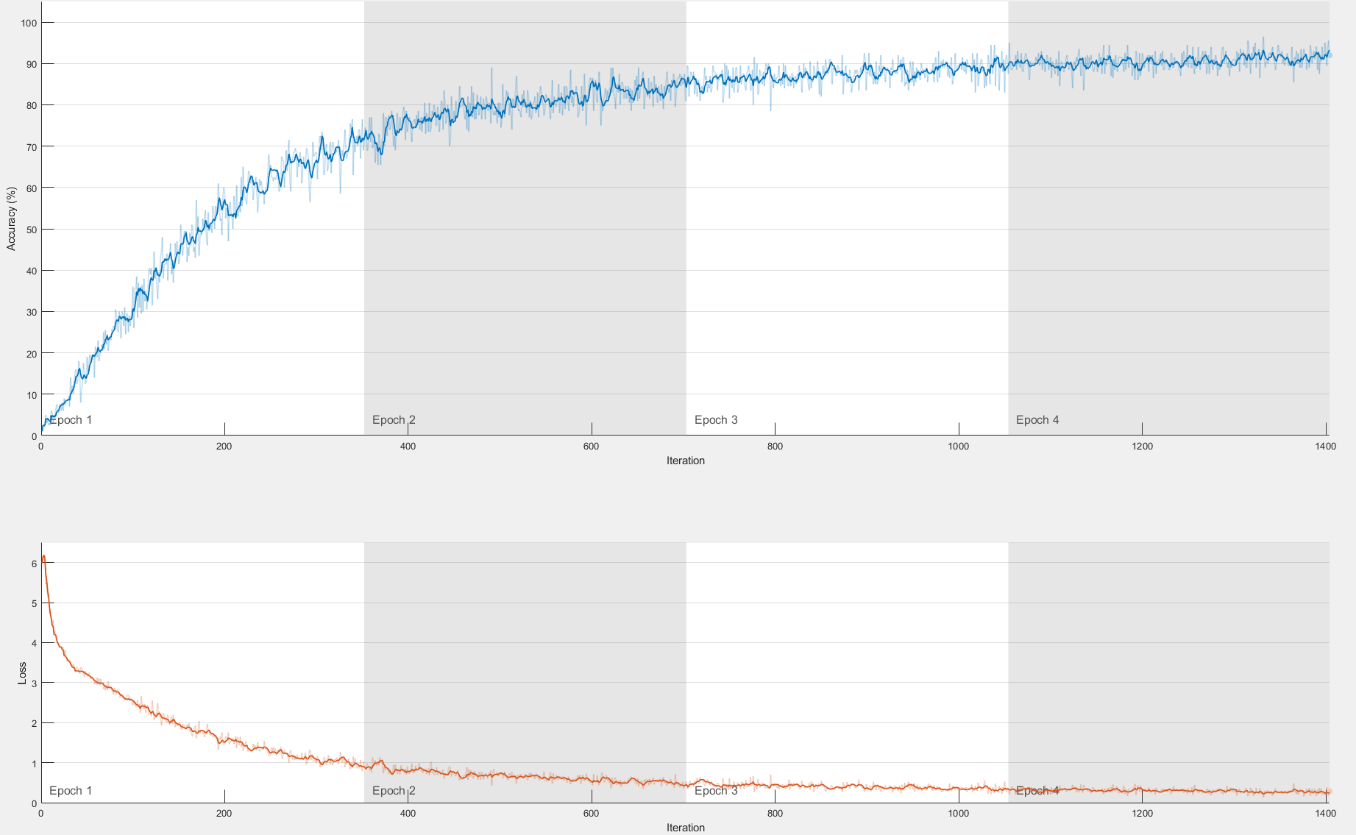
4 Epochs:



*Figure 6: Graph of training accuracy, 4 epochs @ LR = 0.001*

Fig

*Figure 7: Graph of training accuracy, 4 epochs @ LR = 0.0005*



*Figure 8: Graph of training accuracy, 4 epochs @ LR = 0.0001*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Iteration | Accuracy 0.001 | Accuracy 0.0005 | Accuracy 0.0001 |
| 1 | 1 | 2.00% | 2.00% | 1.00% |
| 1 | 50 | 65.00% | 49.00% | 19.00% |
| 1 | 100 | 82.50% | 72.00% | 36.00% |
| 1 | 150 | 88.00% | 80.50% | 41.50% |
| 1 | 200 | 92.00% | 81.00% | 58.50% |
| 1 | 250 | 89.00% | 86.50% | 65.50% |
| 1 | 300 | 88.00% | 92.50% | 68.50% |
| 1 | 350 | 94.00% | 92.50% | 76.50% |
| 2 | 400 | 91.50% | 93.50% | 75.00% |
| 2 | 450 | 94.00% | 92.00% | 78.50% |
| 2 | 500 | 94.00% | 95.00% | 80.50% |
| 2 | 550 | 96.00% | 92.00% | 82.50% |
| 2 | 600 | 95.00% | 93.50% | 85.50% |
| 2 | 650 | 96.50% | 92.00% | 81.00% |
| 2 | 700 | 96.00% | 94.00% | 85.50% |
| 3 | 750 | 97.00% | 98.50% | 81.50% |
| 3 | 800 | 95.00% | 96.00% | 85.50% |
| 3 | 850 | 98.00% | 96.00% | 82.00% |
| 3 | 900 | 97.00% | 97.00% | 88.00% |
| 3 | 950 | 97.00% | 96.00% | 90.50% |
| 3 | 1000 | 98.00% | 96.00% | 89.50% |
| 3 | 1050 | 98.00% | 96.00% | 83.00% |
| 4 | 1100 | 98.50% | 96.50% | 91.50% |
| 4 | 1150 | 97.50% | 98.50% | 91.00% |
| 4 | 1200 | 98.00% | 97.50% | 92.50% |
| 4 | 1250 | 98.50% | 95.50% | 89.00% |
| 4 | 1300 | 98.00% | 98.50% | 91.00% |
| 4 | 1350 | 97.00% | 96.50% | 92.50% |
| 4 | 1400 | 96.50% | 96.00% | 92.50% |
| 4 | 1404 | 99.00% | 98.50% | 92.00% |

*Figure 9: Table of accuracy scores compared against other learning rates over epochs*

To measure the total accuracy of each model after they were fully trained, we tested them against all test images and found their accuracies:

|  |  |  |
| --- | --- | --- |
|  | 2 Epochs | 4 Epochs |
| LR = 0.001 | 97.88% | 98.55% |
| LR = 0.0005 | 97.55% | 98.42% |
| LR = 0.0001 | 92.80% | 96.07% |

*Figure 10: Average accuracies of all variants of model trained*

**V. Conclusion**

Given the accuracy of each of the models, it is clear that the most accurate model trained used a learning rate of 0.001 over 4 epochs. What we can observe is that the models that were trained with a learning rate of 0.0001 were the least accurate due to not being allowed more time to be trained with their smaller steps. However, we must notice that over such little epochs that we achieve over 90% accuracy. With that said, it is clear that if farmers were to train images they collected of their crops to classify diseases or crop variants, AlexNet would be a very efficient model for them to use.

Using these models generated, farmers could load them onto phones, drones, or laptops, and scan their fields to have a greater insight into the state of their farm.

**VI. Future Work**

Using my 2.6 GHz quad-core CPU instead of GPU to simulate resource constriction, a 2-epoch train on my laptop took a little over an hour, and a 4-epoch train took a little over two hours. As I did three variant 2-epoch trains three times, and three variant 4-epoch trains three times, total computation time took around 27 hours. In order to efficiently train more model variants at new learning rates and epoch numbers, I should transition my work to a more powerful desktop with a GPU that MATLAB could utilize. This would allow me to figure out that, given a large dataset of greater than 50k images, at what point should farmers limit the number of epochs that they use to train to prevent diminishing returns and wasted compute time.

Though AlexNet is obviously an efficient model, it would also be interesting testing it against other well-established models and comparing performance over the same learning rates and number of epochs.

**VII. References**

1. <https://www.kaggle.com/vipoooool/new-plant-diseases-dataset>
2. <https://www.ifama.org/resources/Documents/v17i4/Lakner-Baker.pdf>
3. <https://www.mathworks.com/help/deeplearning/ref/alexnet.html>