4390 Project - Final

Katie Dionne

May 9, 2021

1 Abstract

In my plant classification project, I trained a deep learning model to identify what type of plant appears in an image. I began with an overfitting model, adjusting size, filters, and layers. Next, I split the data to train and evaluate models, examining changes in performance as the number of filters and layers varied. The best model from this phase was mod8, and this was my starting point in the next phase. I then created eight models using different combinations of data augmentations and compared them. The best was mod81, which I began with in the next phase, regularization. I used three types of regularization, individually and in combination, to decrease overfitting. The best model with mod816, which I then compared to several pretrained models and architectures.

I found that increasing filter size decreased the accuracy of the model, while increasing the number of filters increased the accuracy of the model. I also found that when deciding the number of filters there was an ideal middle ground where the accuracy was highest. When I applied data augmentation, I found that some augmentations and combinations of augmentations increased accuracy while others did not. Similarly I found that some regularization methods and combinations of regularization methods improved accuracy while others did not.

2 Introduction

My dataset consists of hundreds of photos I have taken of plants in my home. I took photos in different lighting conditions and from a variety of angles. Some photos are of the entire plant, others focus on the leaves. The varieties are discussed below:

2.1 Plant Types

Amarylis is a flowering bulb. I have only one, so all of the amarylis photos are of the same individual.

Avocado: refers to small deciduous trees. I have multiple in my home so the images labeled Avocado are of several individuals.

Kale is a short leafy plant. The photos are of many individual plants of differing ages, but all in the same large planter.

Paperwhite is a flowering bulb. I have a large number of them in a single pot.

Pothos is a small leafy plant which tends to climb from its pot. I have several pothos plants in several locations so there is a wide variety of photos of them.

Seedling is how I classified a variety of young plants less than an inch tall.

Snake Plant is a plant with tall thin variegated leaves. I have several individuals in different locations in my home.

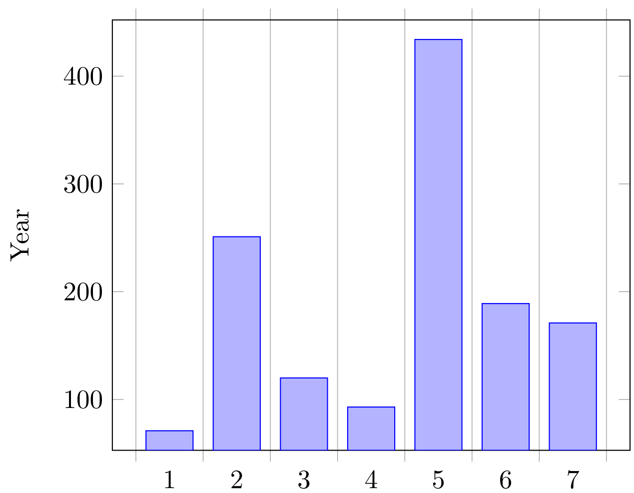
Zebra Plant is a leafy plant with striped leaves. I have only one, so all of the zebra plant photos are of the same individual.

2.2 Image Distribution

The distribution of images was dependent upon the number of each plant that I have in my home. I arranged all of my photos into a set of eight folders, labeled by plant type.

Rather than normalize, I rescaled my images by dividing by maximum. That is, for each image, find the maximum value across pixels and divide all pixels by that maximum value; pixels will take values between 0 and 1. Rescaling improves the efficiency and speed of my model.

1	2	3	4	5	6	7	8
Amarylis	Avocado	Kale	Paperwhite	Pothos	Seedling	Snake Plant	Zebra Plant
71	251	120	93	434	189	171	94

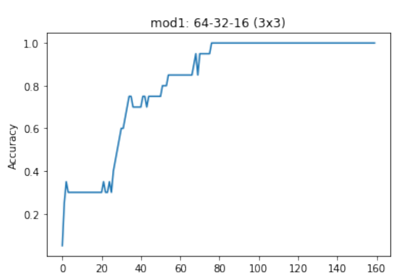


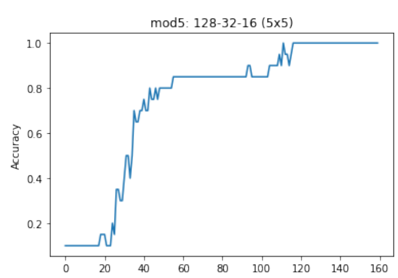
3 Overfit Model Performance

For phase 2 of my plant classification project, I built an overfitting model. In this process I created a variety of models, adjusting the size and number of filters and the number of layers.

3.1 Filter Size

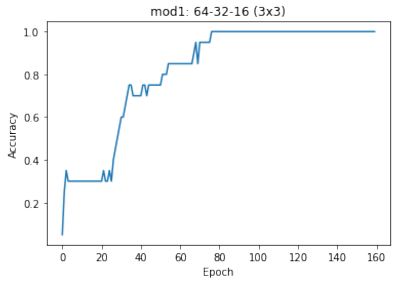
Increasing the filter size appears to decrease the accuracy of the model, or increases the number of epochs needed to reach 100% accuracy. Increasing the filter size also decreases the precision and recall of the model. To illustrate, compare mod1 (3x3) to mod5 (5x5).

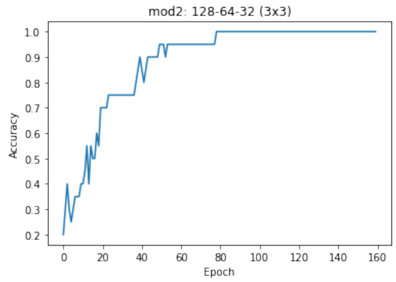




3.2 Number of Filters

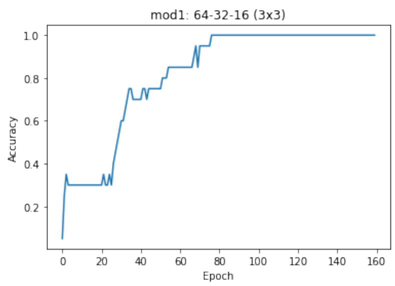
Increasing the number of filters increases the accuracy of the model, or decreases the number of epochs needed to reach 100% accuracy. To illustrate, compare mod1 to mod2, which has twice as many filters.

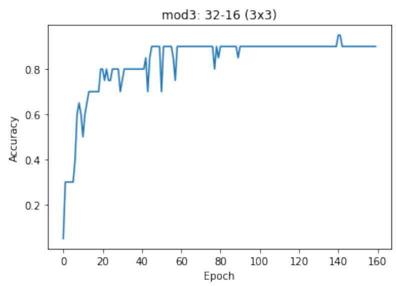




3.3 Number of Layers

Decreasing the number of layers decreases the accuracy of the model. Decreasing the number of layers also decreases the precision and recall of the model. To illustrate, compare mod1 to mod3, which has twice as many filters.





4 Split Data, Train and Evaluate Models

For phase 3 of my plant classification project, I split my data into training, validation, and testing data. To do so, I manually moved my images into three sets of folders, approximately 60% training, 20% validation, and 20% testing.

I then trained my model using the training set, using earlystop with the validation set. Then I evaluated against the test set. I made several different models to demonstrate the changes in performance as the number of filters and layers varied.

I first made 5 models, the best two were models 2 and 3. Models 6 through 9 were variations on models 2 and 3, of these the best was model 8. Models 10 and 11 were variations on model 8, but did not perform better than model 8. Models 12 and 13 were similar to models 2 and 8 respectively.

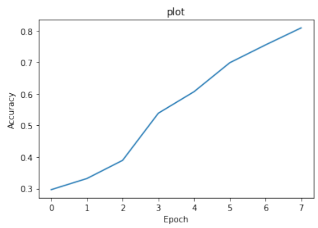
I made models with and without maxpool layers, with more or fewer filters, with more or fewer layers, and with larger or smaller filter sizes.

4.1 Number of Filters

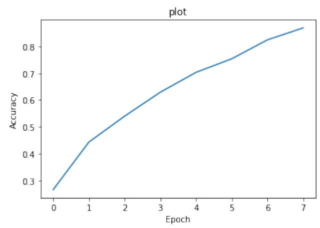
Based on the models I tried, accuracy is highest when there are neither too many nor too few filters.

For example, compare mod 7, mod 2, and mod 6. Mod 6 has twice the filters as mod 2, which has twice the filters as mod 7. From mod 7 to mod 2, increasing the number of filters increases accuracy, but from mod 2 to mod 6 as the number of filters is doubled again accuracy decreases.

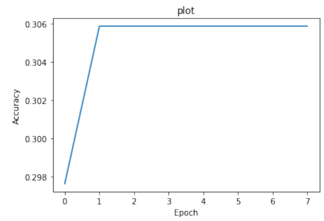
-	mod7	mod2	mod6
training accuracy	.79	.87	.30
validation accuracy	.54	.59	.30



(a) mod7



(b) mod2

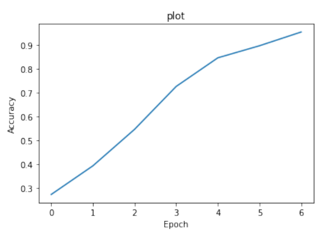


(c) mod6

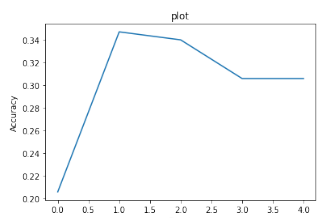
4.2 Number of Layers

In the models I created I found that increasing the number of layers decreased the accuracy. For example, compare mod 8 and mod 10. Mod 10 is the same as mod 8 but has an additional layer.

-	mod8	mod10
training accuracy	.95	.30
validation accuracy	.59	.30



(a) mod8

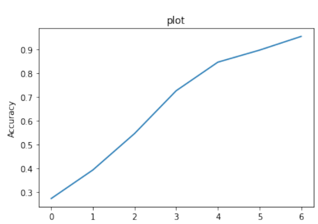


(b) mod10

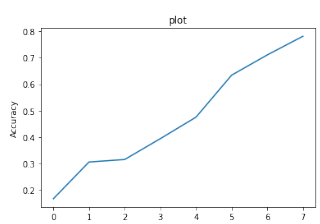
4.3 Filter Size

In the models I created I found that increasing filter size decreased the accuracy of the model. For example, compare mod 8 to mod 13. Mod 13 has a larger filter size but is otherwise the same as mod 8.

-	mod8	mod13
training accuracy	.95	.77
validation accuracy	.59	.38



(a) mod8



(b) mod13

4.4 Phase 3 Conclusion

Based on the models I tried and the learning curves I got, mod8 appears to be the strongest, and in the end it had a testing accuracy of 59.93%.

5 Data Augmentation

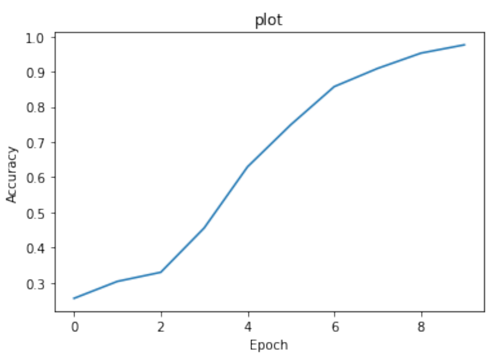
For phase 4 of my plant classification project, I applied data augmentation techniques. Using augmentations helps keep the model from overfitting. This is evident when looking at the accuracy. Without augmentation, I had training accuracy much higher than validation accuracy. But for the models with augmentation, training accuracy was similar to validation accuracy. So although the training accuracy was reduced significantly in the models with augmentation, because of the increase in validation accuracy some of the augmented models can be considered better models.

I began with model 8 from phase 3 as it was my best model from that phase. To examine the results of data augmentation, I created five separate sets of data generators. Each of the first four used a different augmentation or two. The fifth used all of the augmentations together. I then assessed the accuracy of these five models to determine which were improvements over the non augmented models, only the first two (models 81 and 82) were. I used this information to make three more models, different sets of augmentations from these first two successful models. Two of them improved over the second model, but none of them improved over the first model. Of the eight total sets of augmentations, the best was the first model, model 81.

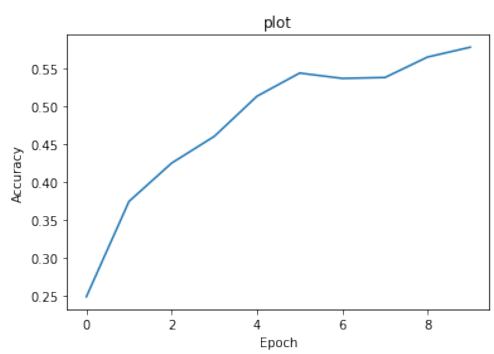
5.0.1 Augmentations

The first model used one augmentation, rotation_range=40. It is an improvement over model 8.

-	mod8	mod81
training accuracy	.98	.60
validation accuracy	.37	51



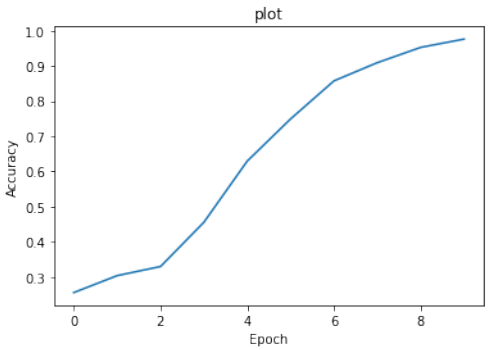
(a) mod8



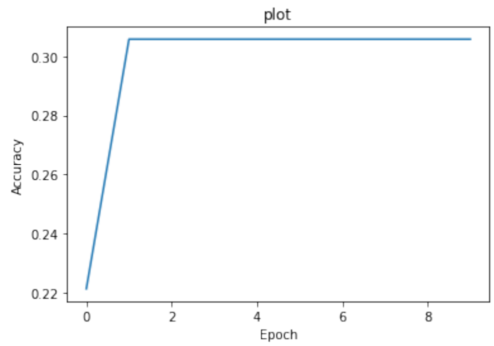
(b) mod81

The fourth model used two augmentations, horizontal_flip=True and fill_mode='nearest'. It is not an improvement over model 8.

-	mod8	mod84
training accuracy	.98	.29
validation accuracy	.37	.30



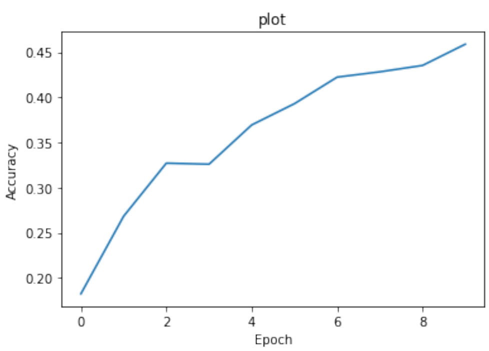
(a) mod8



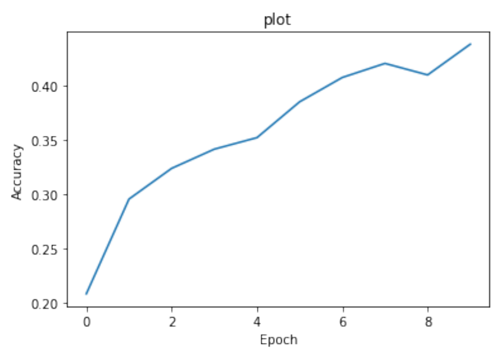
(b) mod84

The eighth model used the three augmentations from the sixth and seventh models, rotation_range=40, width_shift_range=0.2, and height_shift_range=0.2. It is an improvement over model 82 but not over model 81.

-	mod8	mod81	mod82	mod88
training accuracy	.98	.60	.47	.43
validation accuracy	.37	.51	.46	.47



(a) mod82



(b) mod88

5.0.2 Phase 4 Conclusion

It turned out that my best model was the first one, model 81. It had a training accuracy of .60, a validation accuracy of .51, and a testing accuracy of .52.

6 Regularization

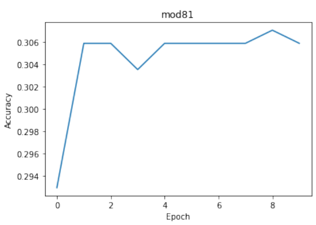
For phase 5 of my plant classification project I applied regularization techniques. Regularization helps with overfitting, and its impact is evident when examining the accuracy of the models. I used three techniques, L2 regularization, dropout, and batch normalization. L2 regularization reduces the complexity of the network by keeping the weights small. Dropout is a method by which output features are randomly dropped during training (in this case 20% of them), which breaks up happenstance patterns. Batch normalization fixes the distribution of a layer before the next layer, to avoid the vanishing gradient problem.

I began with model 81 from phase 4 as it was my best model from that phase. To examine the results of regularization, I created nine separate models, trained them, and compared their accuracy. The first five used one regularization technique each, in different locations in the model. The following three used two regularization techniques per model, and the last model used all three techniques together. Of the nine models, the best was the sixth model (model 816), which used dropout and then normalization.

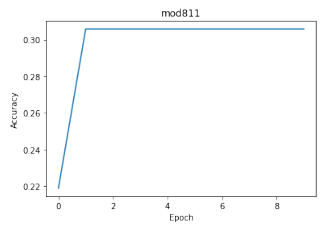
6.1 L2 Regularization

The first two models used L2 regularization. The first (model 811) used L2 early in the model and the second (model 812) used L2 later in the model. The results were similar, and both only narrowly improved over model 81.

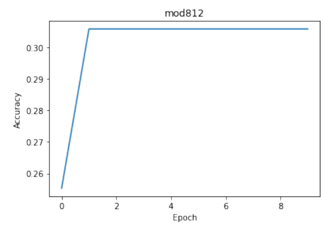
-	mod81	mod811	mod812
training accuracy	.29	.30	.31
validation accuracy	.30	.30	.30



(a) mod81



(b) mod811

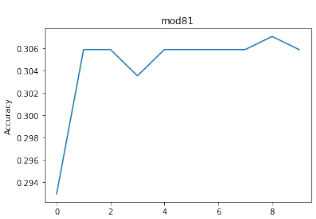


(c) mod812

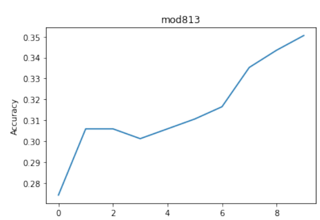
6.2 Dropout

The next two models used dropout for regularization. The third (model 813) used dropout early in the model and the fourth (model 814) used dropout later in the model. The third model had better accuracy than the fourth. Model 813 was an improvement over model 81, model 814 was not.

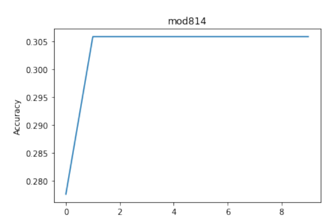
-	mod81	mod813	mod814
training accuracy	.29	.36	.29
validation accuracy	.30	.34	.30



(a) mod81



(b) mod813

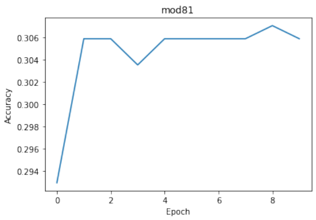


(c) mod814

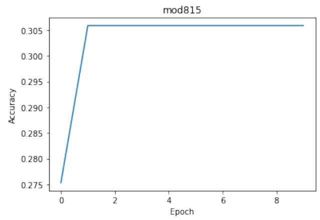
6.3 Batch Normalization

The fifth model (model 815) used batch normalization. It was an improvement over model 81.

-	mod81	mod815
training accuracy	.29	.33
validation accuracy	.30	.30



(a) mod81

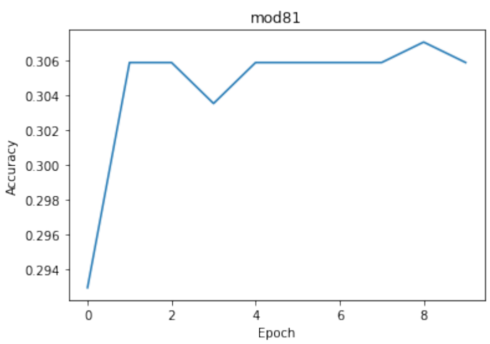


(b) mod815

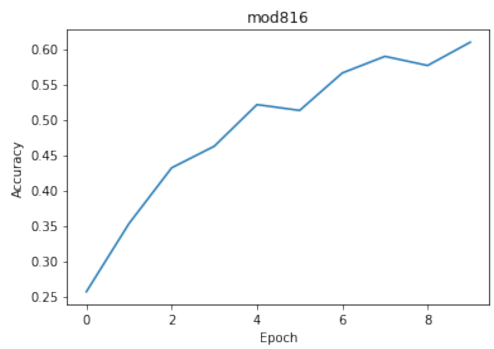
6.4 Two Techniques

The sixth through eighth models used two regularization techniques each. Model 816 used dropout early in the model and batch normalization late in the model. Model 817 used dropout early in the model and L2 regularization late in the model. Model 817 used batch normalization early in the model and L2 regularization late in the model. All three were improvements compared to model 81. Model 816 had the highest accuracy of any model in this phase.

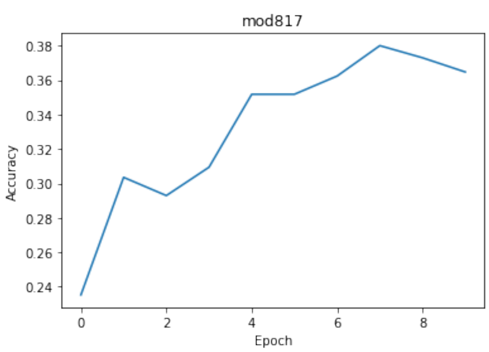
-	mod81	mod816	mod817	mod818
training accuracy	.29	.64	.37	.28
validation accuracy	.30	.48	.37	.30



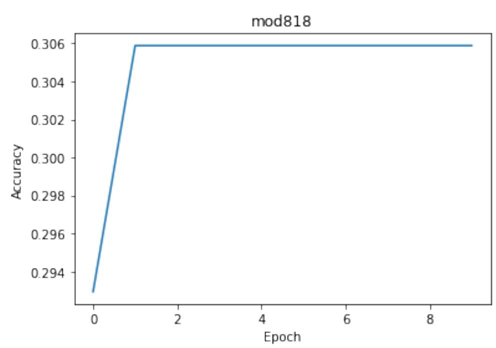
(a) mod81



(b) mod816



(c) mod817

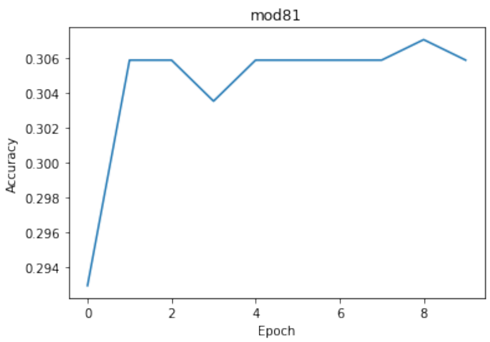


(d) mod818

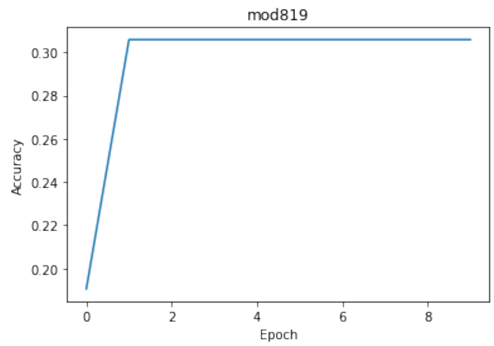
6.5 Three Techniques

The ninth model (model 819) used all three regularization techniques: Dropout and batch normalization early in the model and L2 regularization late in the model. It was an improvement over model 81 but did not outperform any of the models that employed only two techniques.

-	mod81	mod819
training accuracy	.29	.33
validation accuracy	.30	.30



(a) mod81



(b) mod819

6.6 Phase 5 Conclusion

My best model in this phase was the sixth one, model 816. It had a training accuracy of .64, a validation accuracy of .48, and a testing accuracy of .49.

-	mod81	mod816
training accuracy	.29	.64
validation accuracy	.30	.48
testing accuracy		.49

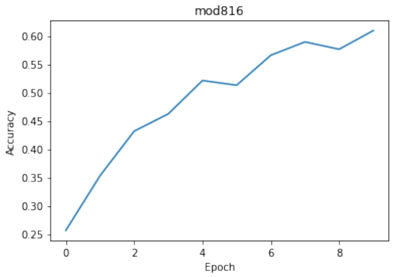
7 Pretrained Models

For phase 6 of my plant classification project I trained a variety of pretrained models and recent architectures and compared them to mod 816 which was my best model from the previous phase. I used VGG16, ResNet50, DenseNet121, and NASNetMobile. Of these the best fit was DenseNet121.

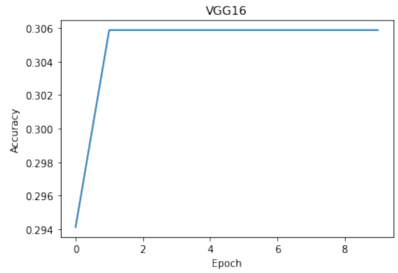
7.1 VGG16

VGG16 is a pretrained model using convolutional layers and pooling layers, which shrinks the image but increases its depth and produces a probability of what class the image belongs to.

-	mod816	VGG16
training accuracy	.64	.29
validation accuracy	.48	.30



(a) mod816

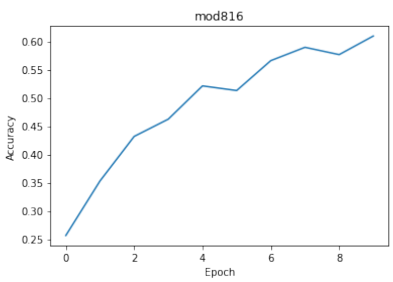


(b) VGG16

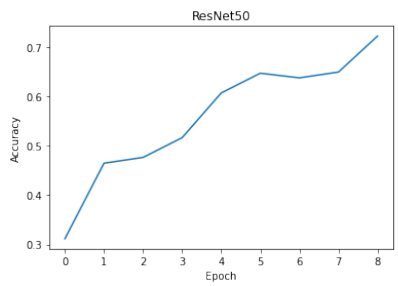
7.2 ResNet50

ResNet50 is a pretrained model using multiple residual blocks, each of which is composed of multiple convolutional layers, maintaining or improving knowledge from the previous layer. The training accuracy is much higher than the validation accuracy, implying that the model is overfit.

-	mod816	ResNet50
training accuracy	.64	.69
validation accuracy	.48	.20



(a) mod816

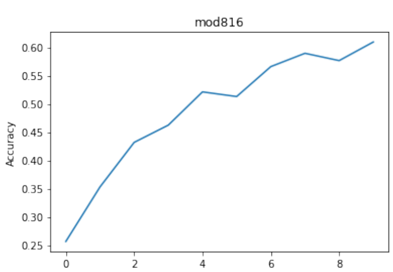


(b) ResNet50

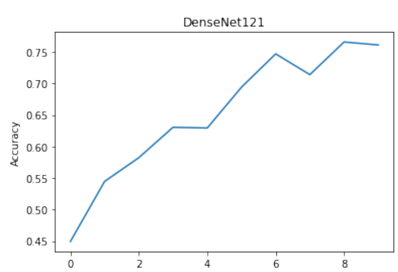
7.3 DenseNet121

DenseNet121 uses a densely connected convolutional network architecture. The training accuracy is much higher than the validation accuracy, implying that the model is overfit.

-	mod816	DenseNet121
training accuracy	.64	.77
validation accuracy	.48	.35



(a) mod816

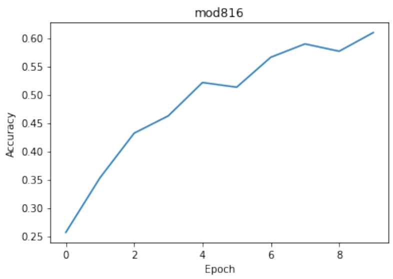


(b) DenseNet121

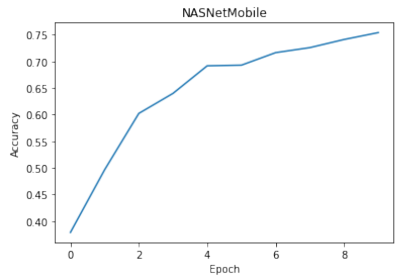
7.4 NASNetMobile

NASNetMobile uses a transferrable image recognition architecture. The training accuracy is very high but the validation accuracy is very low, implying that the model is overfit.

-	mod816	NASNetMobile
training accuracy	.64	.76
validation accuracy	.48	.05



(a) mod816



(b) NASNetMobile

8 Conclusion

There were several factors that limited the accuracy of the model I trained for this project. One of the most significant was computational; with my computer I could only reasonably train models for a few epochs each. Many of my models were also susceptible to overfitting, despite the regularization techniques that were applied. Another drawback of this project was the quality of my dataset. Although I did my best to take photos in a variety of light conditions and from different angles, the images are of a small number of plants in a small number of settings.

Even with these drawbacks, this plant classification project illustrates how powerful deep learning models can be. Despite being trained with a very limited set of images, my model achieved a testing accuracy of 49%, impressive compared to a baseline accuracy of less than 17%. This was all achieved with just a few small layers and a only small number of epochs to train over. This project is also a nice example of the democratization of machine learning; it was completed entirely on a nine year old personal computer and using freely available online software.