Categorizing Equus Members with Deep Learning

Miguel Corona

March 2020

1 Introduction

The objective of the project is to create a convolution neural network that can distinguish the members belonging to the Equus genus. The animals belonging to the Equus genus are horses, donkeys, and zebras. The network shall support the categorization of each pure breed i.e. hybrids such as mules are omitted due to being part horse and part donkey.

2 Data Preparation

2.1 Image Selection

Approximately 1044 total images were collected of horses, donkeys, and zebras. The images were selected such that each one only contains the animal associated with the image's label i.e. there are no images of hybrids nor images that contain that contain more than on class. Images that contain multiple instances of the same class were permitted as entries to the dataset.

2.2 Preprocessing Data

The images were selected to emphasize their label. Any images that contained more than one class were cropped to ensure that a single class is present in the image. All samples were adjusted to become square images of dimensions 256 x 256 pixels with the use PIL. Landscape images were cropped to focus on the label to minimize potential information loss on the resizing of the images.

2.3 Data Distribution and Visualization

The distribution of samples is broken down in the following manner:

Donkey	392
Horse	315
Zebra	340

The reasoning for selecting more Donkey samples than those of the zebra is due to its variability. The breakdown may indicate that sufficient records are needed to distinguish between donkeys and adult horses and adult zebras, but additional donkey samples assist in determining whether a donkey or young horse or young zebra is present. Figure 1 contains a depiction of the data visualization.

Figure 1: Visualization of Equus Samples and Labels

2.4 Data Normalization

Normalization was performed on the images via rescaling of the RGB values such that the values lie in the [0,1] interval. The rescaling of the RGB values on the images may alleviate the skewing that may occur on the grayscale images that may be present in the dataset.

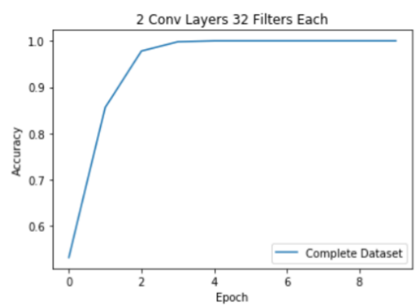
3 Building an Overfitting Model

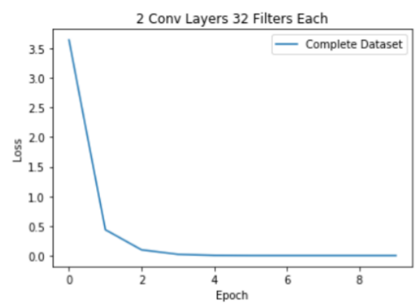
3.1 Initial overfitting model

An initial convolution neural network was developed to identify whether it was feasible to yield a model that can properly overfit the dataset. All samples were utilized for training the model and no samples were partitioned into a validation set or test set. The initial neural network is composed with the following architecture:

Type	Parameters	Kernel Size	Activation
Convolution	32	3	relu
Convolution	32	3	relu
Dense	3	N/A	softmax

The model was trained across 10 epochs and yields the learning curves:



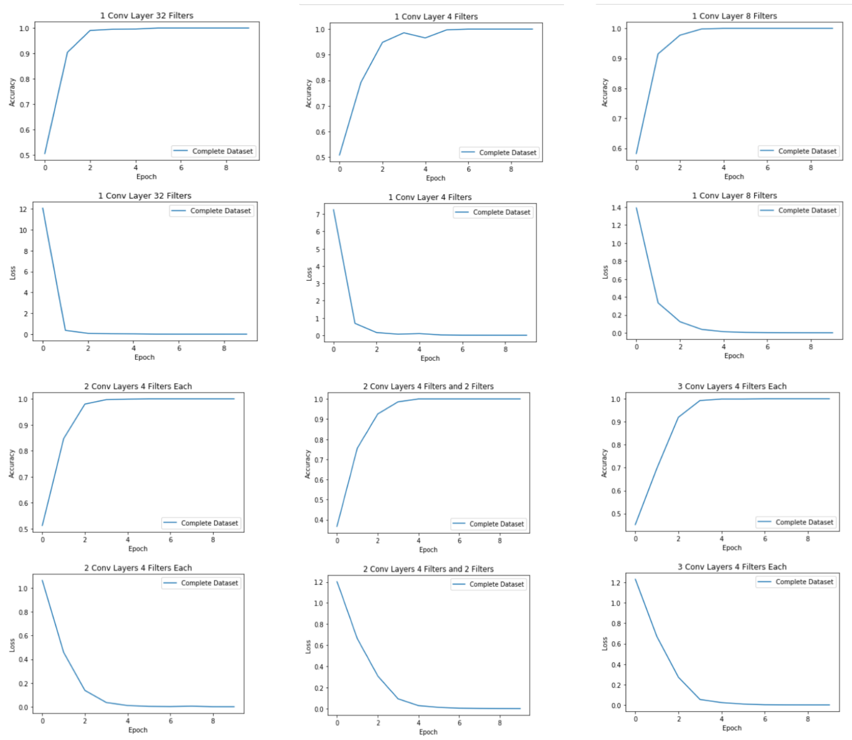


Params	Epoch (s)	Final Acc	Final Loss	Overfit Epoch
6,106,531	84s	1.0000	1.4374e-04	5

The accuracy of the model converged to an overfitted state by the midpoint of total epochs defined. Each epoch was approximately 84 seconds which indicates that the initial overfitting network may not be optimal as less complicated models discussed later yield comparable results. The largest improvement in the accuracy of the model occurred at the early stages of training and the loss converged to its minimum value.

3.2 Performance Dependence on Filters and Layers

Additional networks that differ in the number of filters and the number of layers were constructed once it was determined that overfitting was plausible. Each of the networks is described in the table by the number of convolution layers and number of filters per layer. All networks utilize relu as the activation function for each convolution layer and have a dense layer with a softmax activation as the final layer. The learning curves of the variable networks are depicted below:



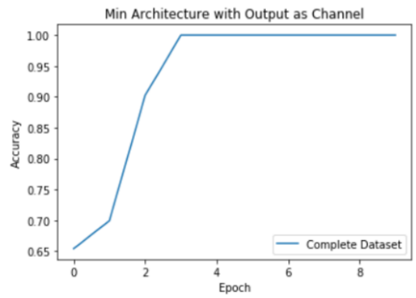
Model	Params	Epoch (s)	Final Acc	Final Loss	Overfit Epoch
1 Conv 32	6,194,435	18s	1.0000	3.7188e-04	6
1 Conv 4	774,307	11s	1.0000	0.0031	7
1 Conv 8	1,548,611	11s	1.0000	7.7929e-04	5
2 Conv 4 4	762,311	23s	1.0000	5.7619e-04	6
2 Conv 4 2	381,213	22s	1.0000	0.0017	5
3 Conv 4 4 4	750,411	35s	1.0000	0.0022	7

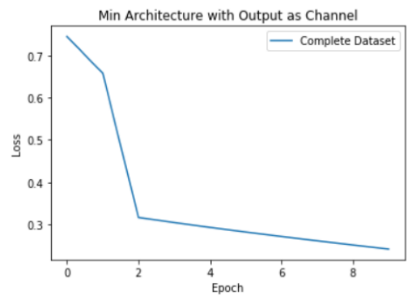
The results of the variable networks seem to indicate that the number of layers do not have a significant effect on the convergence of a model to an overfitted state if sufficient epochs are performed. All models perfectly classified each sample in the dataset albeit at different epochs and with different final loss with sufficient training. The number of layers or filters alone in the network do not seem to have a significant impact on the final loss compared to the impact of the total number of parameters as the networks with the most parameters yield smaller loss values up to a certain point. The number of layers may affect the loss by favoring the model with less layers when the number of parameters in the network are comparable. It was observed that the number of layers impact the time required to complete an epoch. The observation between the number of parameters and the number of layers are consistent when the results are compared to the initial overfitted model as the number of parameters between 2 Conv 32 32 is comparable to that of the 1 Conv 32 model; Both yield comparable loss values, but with a significant difference in the epoch completion time.

3.3 Minimum Architecture if Label as an Input Channel

An experiment was performed to identify the minimum architecture required to overfit the dataset if each sample contains its label as an additional channel. The images acquired from the ImageDataGenerator were converted from their initial shape of (256, 256, 3) to a shape of (256, 256, 6). The first three elements of the final axis represent the RGB value of the pixel and the last three elements contain the tensor representation of Donkey, Horse, or Zebra. The tensor elements in the final axis remain normalized as the RGB values were normalized via the ImageDataGenerator and the classification tensor contains a single 1.0 value to represent the label.

The network was trained utilizing model.fit() unlike the prior networks due to the transformations required to reshape a sample to shape (256, 256, 6). The network was trained for 10 epochs similarly to the prior models to yield the results:





Params	Epoch Time (s)	Final Acc	Final Loss	Overfit Epoch
387,209	13s	1.0000	0.2416	4

Providing the classification of the sample as an input channel permits a simpler network to overfit the data much sooner than the models observed earlier. An area of concern with having the label as an input channel is the resulting loss as this model overfits sooner, but yields a relatively large loss compared to the prior models. The 2 Conv 4 2 model contains a comparable number of parameters to the minimum architecture with the label as an input channel, but the loss for each respective number differs in magnitude with the 2 Conv 4 2 model yielding a smaller loss value.