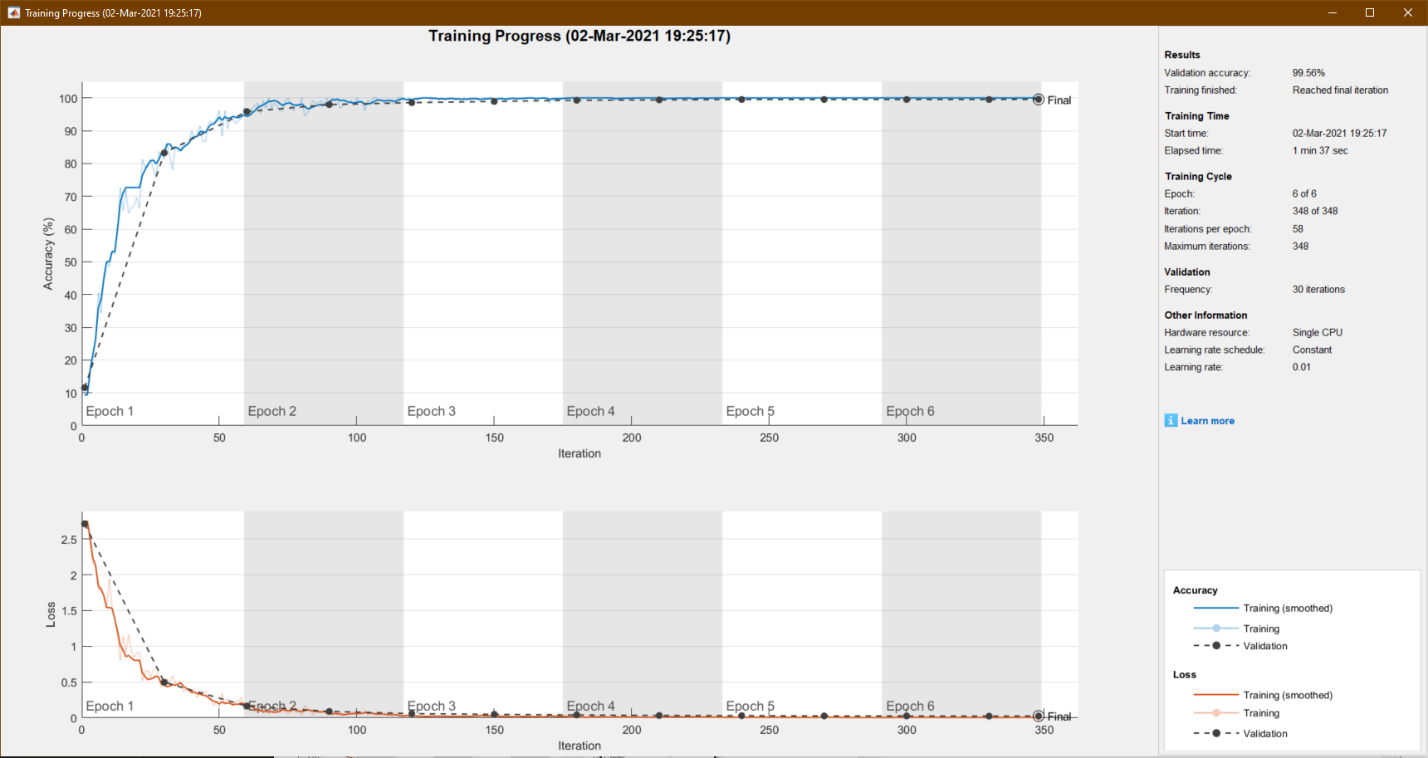
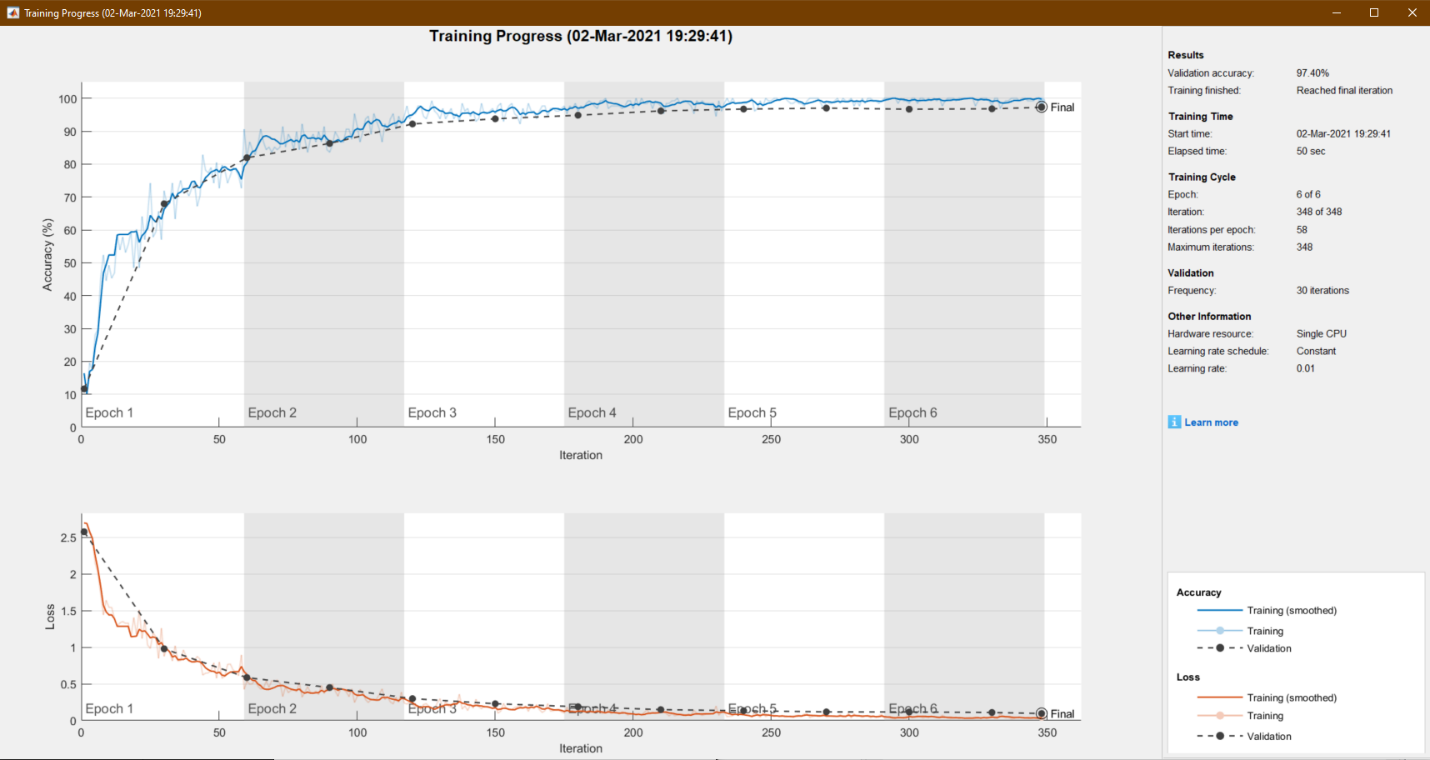
**Rex PSet8 Q3**

**Training results using 2 convolutional layers without pooling (99.56%):**



**Decreasing number of convolutional layers used:**

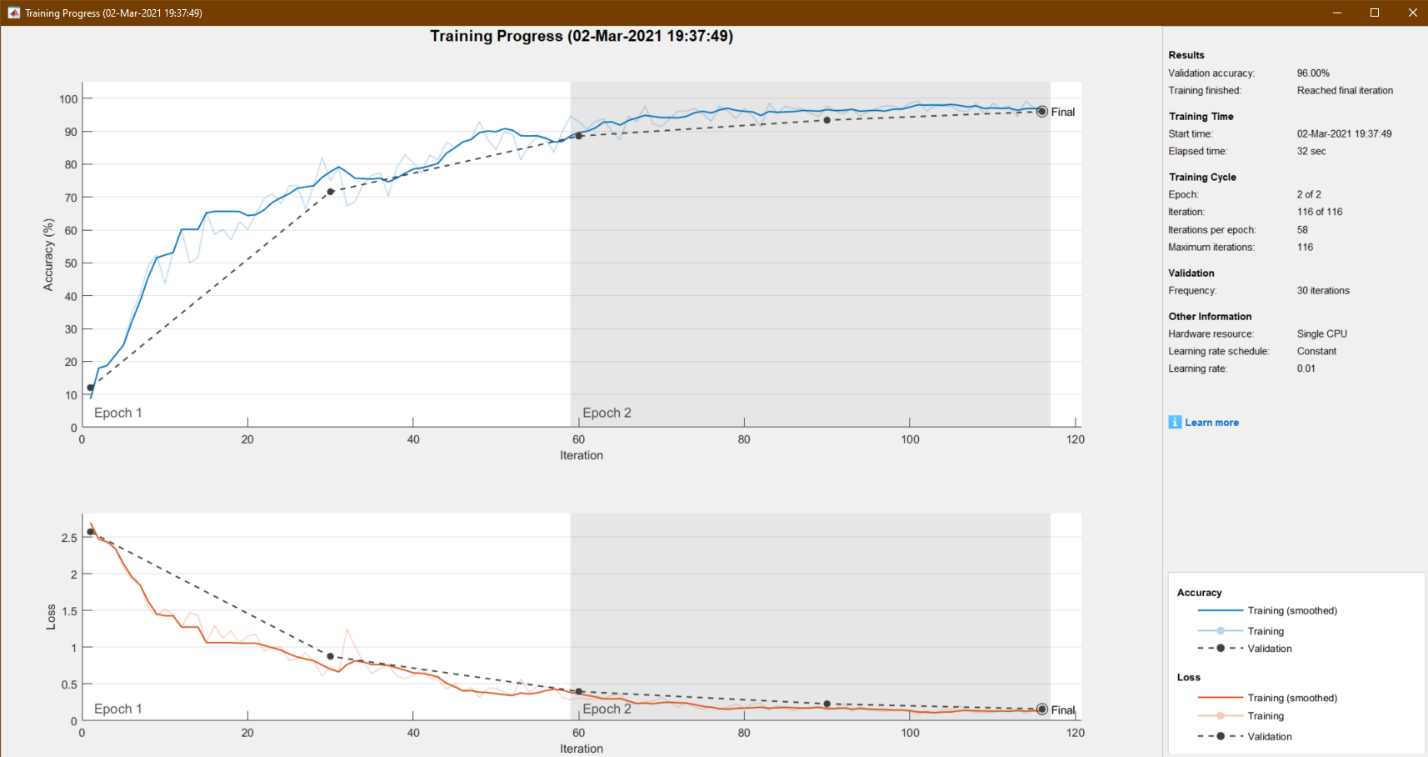
Training results using 1 convolutional layer without pooling (97.40%):



As seen from the above training session, even though only one convolutional layer was used, the model's training accuracy still greatly exceeds that of a model built with fully connected layers.

**Decreasing number of epochs ran:**

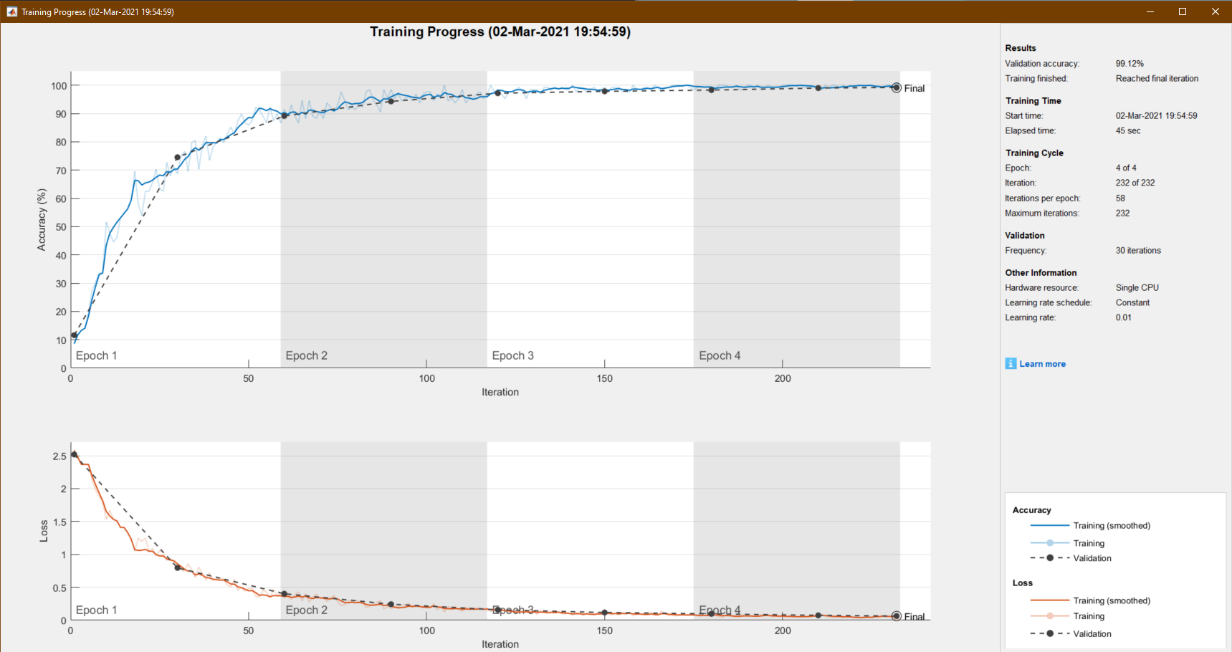
Training results using 2 convolutional layers without pooling running 2 epochs \*normally 6 (96.00%):



As seen from the above training session, limiting the number of epochs ran from 6 down to 2 for a convolutional layered model still nets a training accuracy that is much higher than its fully connected layer counterpart. As shown in the training progress above, even after 1 epoch, the training accuracy already reaches around 90%; compared to 85.20% for a model with two fully connected layers after running 6 epochs.

**Using MaxPooling2D to improve accuracy + lower training time:**

Training results using 3 convolutional layers with pooling running 4 epochs (99.12%):



**Analysis:**

**Why are convolutions better suited for object recognition in images?**

Convolutions are better suited for object recognition in images because they are able to highlight specific features in an image. This is accomplished by multiplying all the pixels in an image by a filter and replacing all the original pixels with new ones. Extracting these features in an image allows the model to better recognize the differences between two differing objects (e.g.: a circle shape in 0,8,9; compared to straight lines in other numbers, etc.)

**Which combination results in the least amount of time to train for a given accuracy?**

The least amount of training needed to reach around a 99% accuracy requires a combination of 3 convolutional layers with pooling running 3-4 epochs. (Around 45 seconds of training time) This is because every time we use a convolution, it is effectively creating multiples of our original dataset by multiplying all the pixels with different filters, creating new images. Integrating pooling allows us to compress these images down by effectively throwing away 75% of our pixels and significantly reducing training time. When using pooling along with convolutions, not only does the quality of the image not decrease, the vital features are actually enhanced as pooling only selects the pixels with the highest values in a group of pixels in order to highlight the contrasting features in an image. Thus, by utilizing a combination of convolutional layers and pooling, we can achieve a higher training accuracy while also minimizing training time. In this particular case, we can also lower the number of epochs used during training down to around 3 because the validation accuracy begins to plateau after 2 epochs, further lowering the time elapsed during training.