The difference between the original model and the new model is very minimal. From my understanding, each layer added gives the model the chance to involve more parameters to learn.

I experimented with adding more layers and changing the settings for the existing layer. For the sake of simplicity and speed, I decided to make the minimal amount of changes that allowed me to achieve 100% accuracy. I understand that there is a very high likelihood of overfitting but since the goal was to achieve higher accuracy, I only made minimal changes.

The first layer is the same in both models. I changed the filter size to 72 for the second layer. Filters size is the number of convolutions, essentially the number of neurons is 72. It helped learn with less epoch iterations and did not slow down the model as much as higher filter values. I also changed the kernel size to (5,5). I tried different kernel sizes and (5,5) gave me the best results.

I increased MaxPooling size to (3,3) which essentially summarizes the larger set of information. This is known as down-sampling which is used to reduce dimensionality while keeping the max value. I also added strides of 2 which skips 2 values when traversing horizontally and vertically, this helps reduce the size after convolution operation. I tried different stride values but 2 seemed optimal.

I added a dense layer with relu activation. I tried using different activation functions such as sigmoid, softmax and softplus. I learned that ReLu worked well in general for most cases and is efficient.

The only other change I made was decreasing the batch size to 72. Helped the model get to 100% a bit faster and more efficiently. I also realized that I could reduce the epochs to 10 but I left it in just to show that it was part of my understanding that after a certain number of iterations the model doesn’t learn any better.