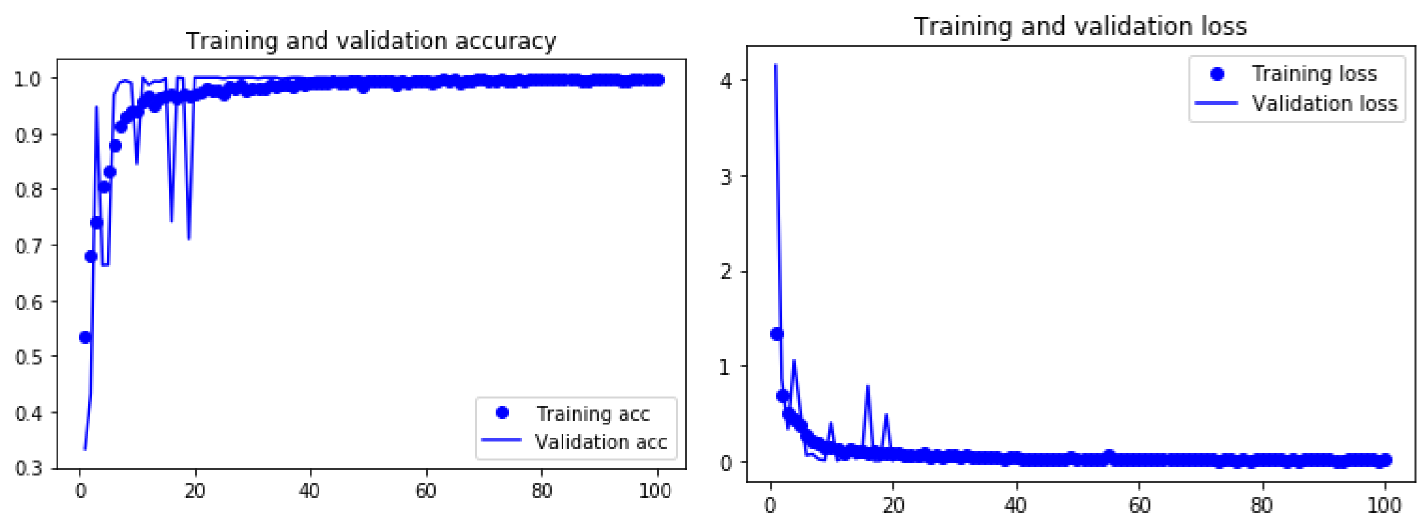
Project 2
Convoluted Neural Network Shape Detection
CSC 412
Anthony Astarita and Briana Connolly

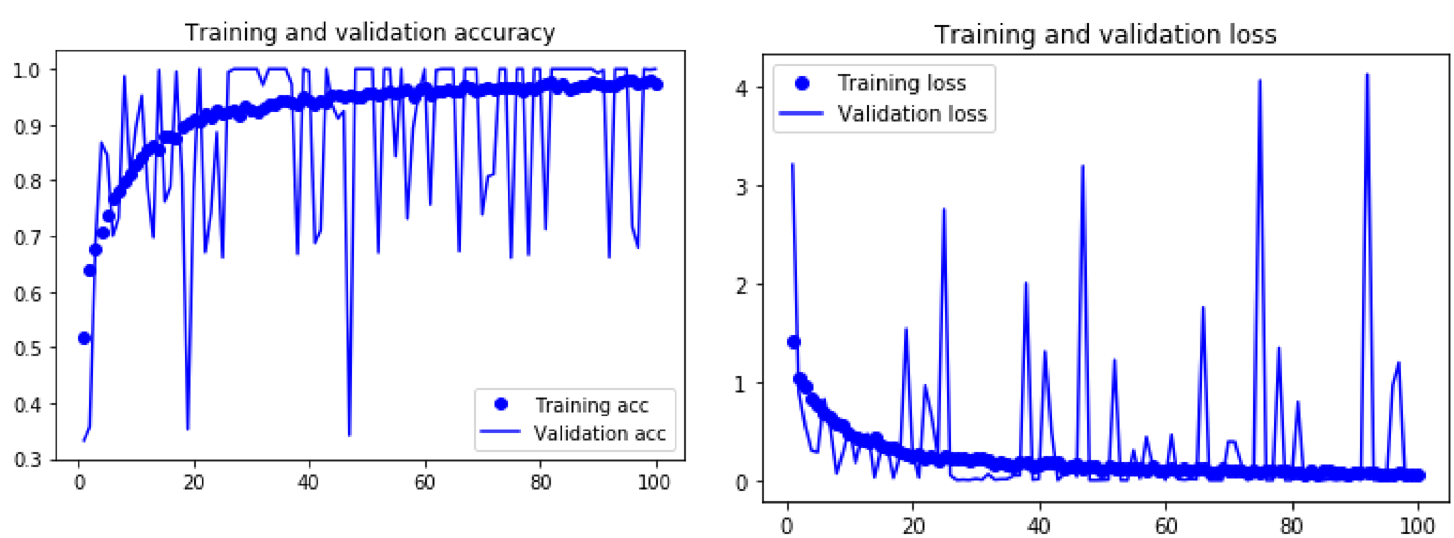
Introduction: The purpose of these experiments were to use a Convoluted Neural Network (CNN) to perform image classification for a multi-class dataset. We had trained two CNNs with different optimizers and activation functions. We used a dataset that included hundreds of pictures of shapes. The four shapes we were trying to classify were squares, stars, triangles, and circles. The dataset can be found here: https://www.kaggle.com/smeschke/four-shapes.

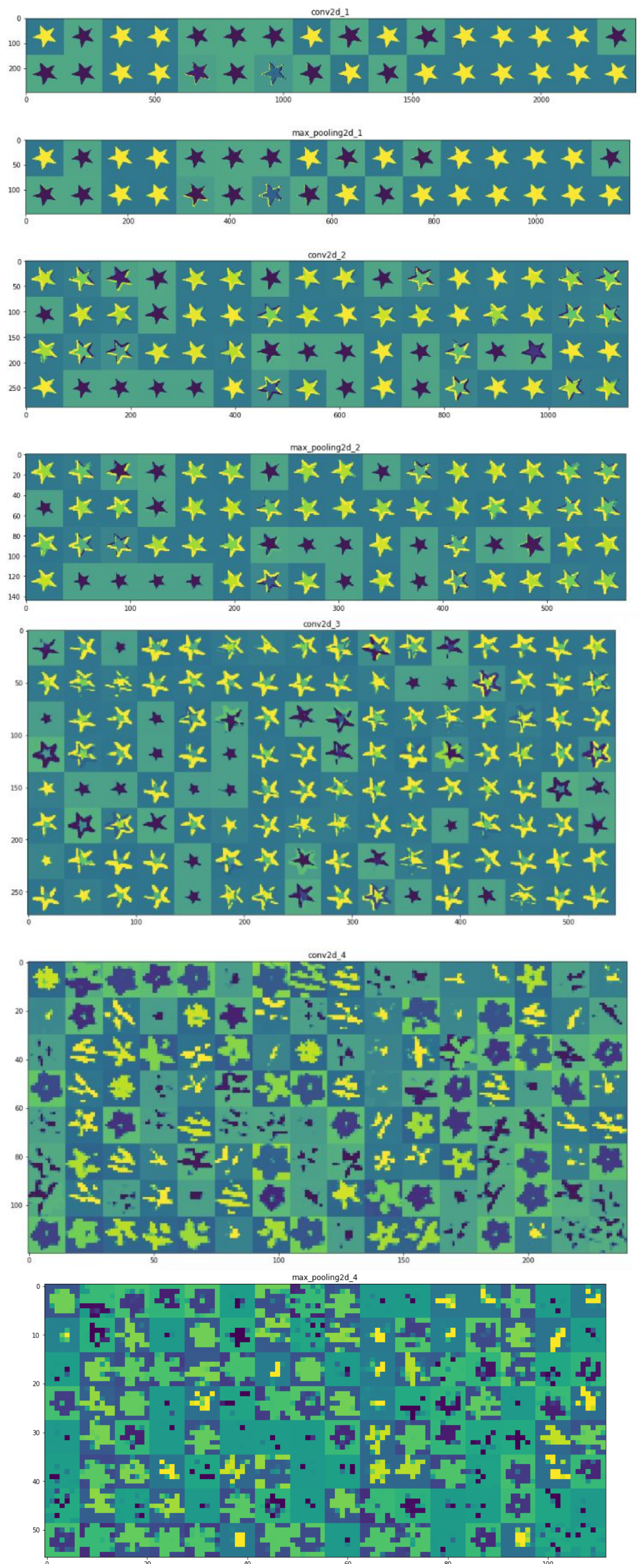
Topology Description: The CNNs included four Conv2D layers and four MaxPooling2D layers. In addition, we added a Dropout layer. The purpose for the addition of the Dropout layer is to prevent overfitting. This layer will allow our CNN to “dropout” certain nodes that tune our model too tightly to bad data. In addition, we augmented our data to further avoid overfitting. We also used normalized mini batches with size 32 in order to increase speed and performance of the CNN. We The first CNN used Adam optimizer with the relu activation function, and the second CNN used an Adadelta optimizer with tanh activation function.

Adadelta with Tanh

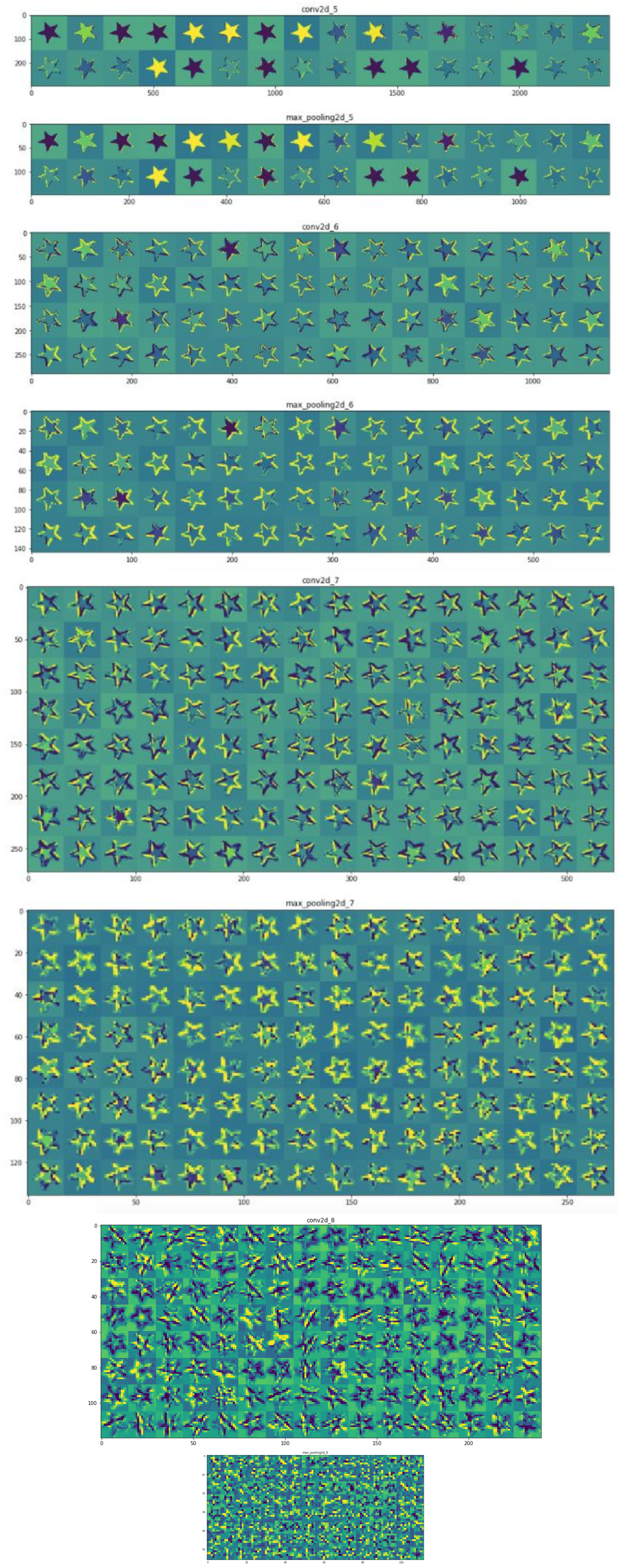


Adam with relu





Adadelta and tanh



Adam and Relu

Analysis: The first experiment, using the Adam optimizer and relu activation function yielded pretty good results. The accuracy of the model at the first epoch was at about 50 percent. The accuracy quickly increased and plateaued at about 95 percent at epoch 40. The training loss showed a similar pattern, plateauing at about epoch 40. As for validating accuracy and validation loss, there was significant variation in the model. There were many scattered noticeable relative minimas for the validation accuracy. The model ended with about a 97 percent accuracy predicting the shapes. This model was quickly able to pick up the features of each of the shapes. We were surprised that by the end they did not active a closer accuracy to 100 percent.

The second experiment seemed to work slightly better than the first. The Adadelta and tanh combination seemed to work very well. The learning rate for this model was significantly higher, at 1.0. This may have made the accuracy plateau at a sooner epoch. This experiment's accuracy had plateaued at around epoch 20. In comparison, there was very little in the validation accuracy and loss. This again can be attributed to the much higher learning rate. The Adam and relu model only had a learning rate of 0.0001. This, combined with the architecture of Adadelta may have increased the final accuracy of the model. The model ended with an accuracy of about 99.7 percent. It might also be the case that because of the lower learning rate, the model might have eventually reached an accuracy of 99 percent, but at a later epoch.

It was important to visualize every channel in every intermediate activation. The activation layers show how the model recognizes different features of the image. For example, the activation layers shown are showing the features of a star, and at every point in the CNN, what the model is "seeing". The first layer is retaining the full shape of the star, retaining all information present in the original image, not applying many filters. As the layers go deeper and deeper, the intermediate activations become less and less understandable. By the sixth layer, it is getting hard to recognize the original shape. The deeper the layer, the more information is related to the class of the image.

Conclusion: The two experiments had ended with very good accuracy, but the Adadelta and tanh combination model seemed to reach a higher accuracy and at and earlier epoch. In addition, there was less consistency with the Adam and relu model. This might have been because of the lower learning rate. A lower learning rate allows the model to not overfit to the data, at the drawback of learning at a slower rate. Ways to improve the experiment would be to increase the learning rate, that way it wouldn't have taken more epochs to reach ninety nine or a hundred percent. Another option would be to increase the number of layers. The addition of layers might change the feature detection so it would be easier and faster to detect the features of the shapes.