

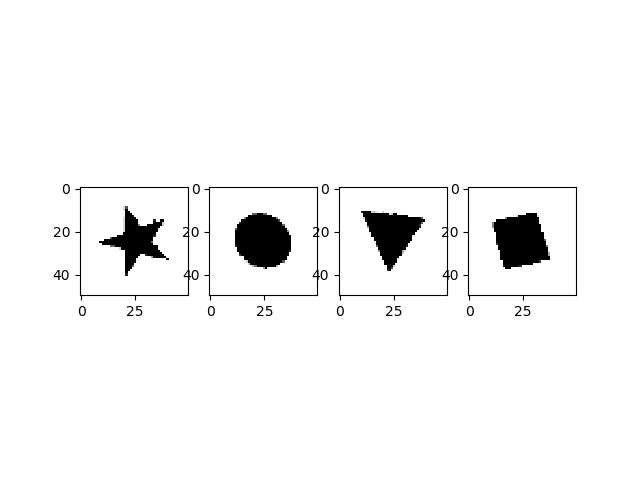
**Problem #3:**

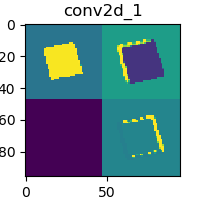
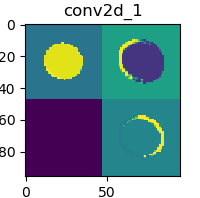
* For problem #3 I have designed three separate networks that all meet the minimum requirement of 90% accuracy across training, validation, and test data sets. However, they all vary in terms of network architecture and dimensions. “ShapeNet1” consists of two convolution layers using four kernels each with average pooling (2X2) following each respective layer, with a fully connected layer of eight neurons and a softmax output layer consisting of four neurons to appropriately classify the four shape categories. “ShapeNet2” reduces the number of convolution layers down to a single layer with 16 kernels 3x3 in size, a single max pooling layer, dropout with a 50% probability and as before, a dense layer of eight neurons followed by the output softmax layer. This network demonstrated difficulty in consistently converging to a solution. Often, the network diverged to random classification with an accuracy of 25%. However, when convergent conditions were found, it provided competitive accuracy. The third network, “ShapeNet3” is the smallest network designed but, it also has the lowest performance. This network is constructed of two convolution layers with four and two kernels respectively, it also employs Max pooling after each convolution layer. The major difference in this network is the removal of the dropout and first dense layer. Now when the data is flattened it is directly passed to the softmax layer and classified. We were able to meet the minimum accuracy thresholds in all four categories. However, the network accuracy dipped by nearly 4.5%. For the purpose of this assignment “ShapeNet3” is the minimum network in terms of number of kernels per layer as well as total learnable parameters by a factor of 3.78 from the ShapeNet1 and a factor of 68 fewer than ShapeNet2. A summary of the network architectures can be found in the table below.

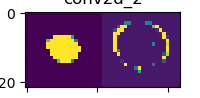
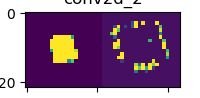
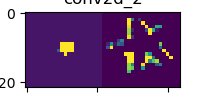
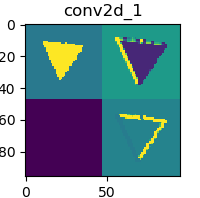
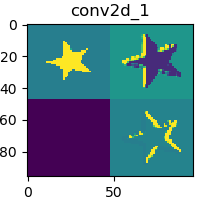
|  |  |
| --- | --- |
| **Network** | **Architecture Summary** |
| **ShapeNet1** | Conv2D(4,(3,3)) 🡪 AvgPool2D(2,2)🡪 Conv2D(4,(3,3)) 🡪 AvgPool2D(2,2)🡪  Dropout(50%)🡪 Dense(8)🡪 Softmax(4) |
| **ShapeNet2** | Conv2D(16,(3,3)) 🡪 MaxPool2D(2,2)🡪 Dropout(50%)🡪 Dense(8)🡪 Softmax(4) |
| **ShapeNet3** | Conv2D(4,(3,3)) 🡪 MaxPool2D (2,2)🡪 Conv2D(2,(3,3)) 🡪 MaxPool2D (2,2)🡪 Softmax(4) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Network** | **Training Accuracy**  **(%)** | **Validation Accuracy**  **(%)** | **Test Accuracy**  **(%)** | **Trainable**  **Parameters** | **Learning Rate** | **optimizer** | **Epochs** |
| **ShapeNet1** | 93.63 | 99.31 | 99.84 | 4104 | 1.00E-03 | RMSProp | 25 |
| **ShapeNet2** | 94.78 | 99.27 | 99.73 | 73932 | 1.00E-03 | RMSProp | 15 |
| **ShapeNet3** | 90.54 | 94.37 | 95.32 | 1086 | 1.20E-03 | RMSProp | 25 |

* Above are shown the tabulated results of all three tested networks, their parametric sizes, optimizers, and necessary hyper parameters. As can be seen, ShapeNet3 is able to achieve the desired minimum accuracy with the least parameters at a cost of accuracy.







* Four examples of images from the test data set are shown in black and white above. Using the activated outputs of the network, normalizing, and scaling them in to the standard pixel range of 0 to 255 we are able to visualize the activation maps of each convolution layer. The images above are from ShapeNet3 (the minimum network), and as would be expected the activation maps in the first row show development of low level features (which are most prevalent in these simple images). Some of these features appear to be the general internal area of each shape, as well as the edge schemes of each shape style (where yellow is the areas in which the activations are largest). However, the second convolution layer becomes slightly more abstract but maintains a similar pattern, discovering the internal areal of the shapes as well as a more combined version of the edges rather than their initial disjoint versions. Note that all shapes show no recognizable activation in the third map, this likely means that the network could be reduced to use one less kernel, however this would need to be empirically tested for confirmation.
* Below, the plots of loss vs epoch and accuracy vs epoch are shown for the minimum network. The envelope of both plots behave as would be desired, with the loss showing a decreasing trend towards zero, and the accuracy approaching a maximum of 100%. The gap between the training and validation accuracy may indicate some overfitting, which is not unexpected due to the small data set being used to train the network. The accuracy plot also shows that at around the 8th epoch the network begins the process of plateauing and only achieves slow progress beyond here and would take a much larger number of epochs to approach 99% like its larger siblings, if it is even possible to achieve this without overfitting the given dataset.

