**Three classes object classifier**

## Creating an object classifier:

The diagram in figure 1 shows the steps followed for creating the classifier. It starts by creating a dataset, then preprocessing the data, after that the convolutional neural network architecture that will be used in the classifier was created and trained with the collected data. The final step is to calculate the score for both datasets and test sets.

1. Dataset:

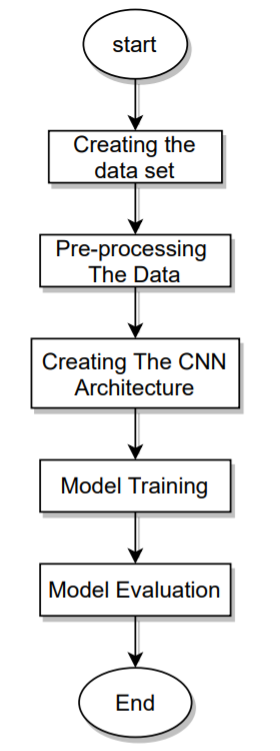
The classifier was meant to recognize two different objects, water bottle, and cup. It is formed from 1000 examples for cups, 1000 examples for bottles, and 1000 dummy example. Making in total 3000 total examples. Figure2-A shows a few samples for the cup, B shows examples used for the bottles and C shows the dummy examples. The training set is formed from 2850 examples (95%) while the test set is 150 examples (15%).

Figure 1 Flowchart shows the flow of the code



Figure 2 Sample from the dataset

1. Preprocessing:

Dataset is formed from a huge number of images in a messy was, for instance, each image has a different size. Machines in general don’t understand images or vided. In the preprocessing step, data is prepared to be inputted to the neural network to achieve the best results from the classifier.

The first step in preprocessing is image resizing. Here all the images were scaled to have width=100 pixel and height= 100 pixels before adding them to the neural network. We are using RGB images, so we have three channels for each image. The final shape of the dataset is 3000\*100\*100\*3. The second step in preprocessing was data normalization, in general, each pixel in the image has a value between 0-255 that indicates the color of the pixel. Normalization is to scale all the values to be only in range 0-1. There are several ways for normalization in general. But in image processing, this is the one always used:

Where Xi is the training example number i.

1. Creating the CNN:

Figure 3 shows the architecture used for creating CNN. It starts with 16 filters with shape 3\*3. Then a relu function is used as the activation function. after that, there is a maximum pooling layer with a 2\*2 window. The results will be flattened which will change the shape of the matrix from a 3-D volume to the 2-D array and then finally a softmax is used for classification.

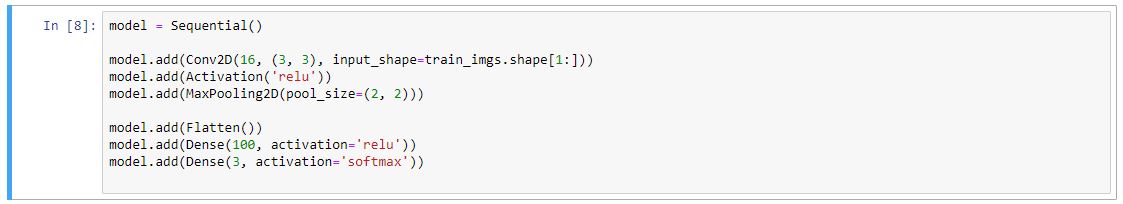


Figure 3 The architecture used

1. Model training:

Training a neural network is usually a recursive procedure, it involves three tasks. The first step is forward propagation. In this step, the training data are passed through the neural network to be classified. The second step is to calculate the cost function which is the objective function needs to be minimized. The last step is the backpropagation. In this step, the weights in the neural network get updated.

In this model, the training data set is formed from 2850 example where each batch size is equal to 32 examples. the code will have 20 epoch and the learning rate () was set to 0.05. Adam optimizer was the optimizer used and the loss function was the categorical cross-entropy.

1. Results:

The last step was to find the results. Figure 4 shows the results of the first epoch. After running the 20 epoch the classifier was found to have a training accuracy equal to 100% and a validation accuracy equal to 77.3%. the training loss is equal to 0.0016 while the validation loss is equal to 0.8219.

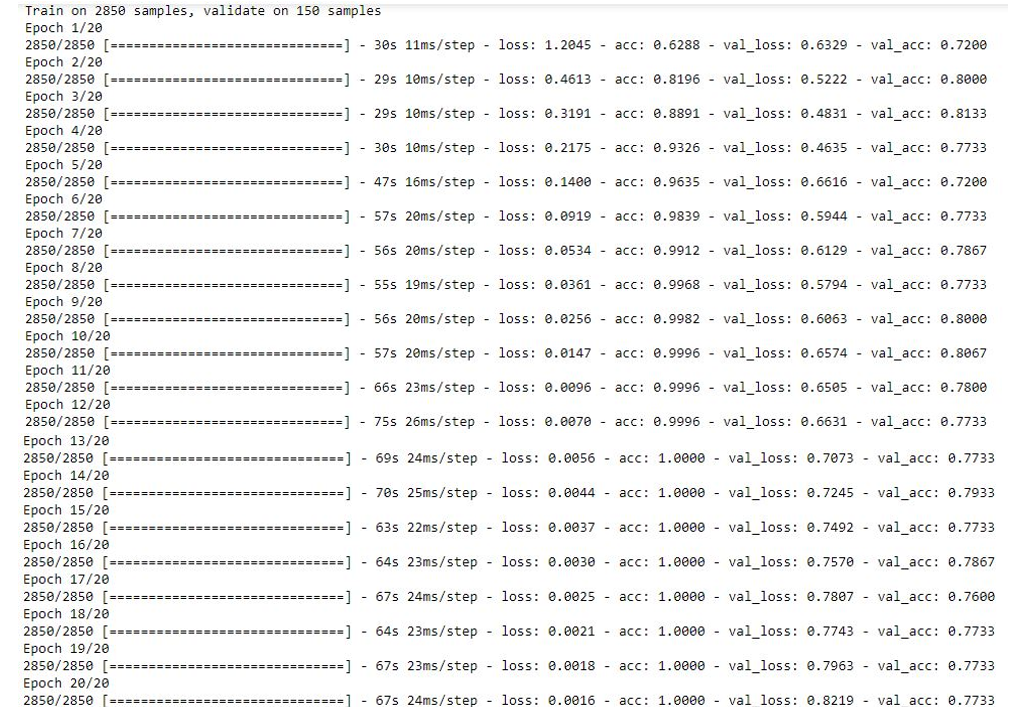


Figure 4 The results of the first epoch