Machine learning assignment - 2

Akarsh Somani,

IIIT Kalyani

**Data Information**:- Dataset contains 13 classes and each class having approximate 155 images.

**Task:-**Train a CNN base Classifier, you can use any deep learning framework and any CNN architecture.

**Submit:**- Training and Testing code with model and a small report of your approach.

Dataset Link: <https://drive.google.com/file/d/1QkAWDrnpLuMs3XfQab9XGNGaJQwEetIq/view>

My approach-

1. First, load the data set in a NumPy array and assign a label to it accordingly.
2. As can be seen that all the images have almost different dimensions, I have used cv2 to resize the images to 250x150 size matrix and stored it in to different variable.
3. Now I will split the data into two parts that is test and train. Size of test will be 20% of the total data size.
4. As 13 labels are possible, I will one hot encode the labels for both y\_test and y\_train. For x values in range [0,255] I will convert it into [0,1].
5. Now I will use ‘Keras’ to declare the CNN model.
   1. Convolution layer with 16 filters and ‘relu’ activation function.
   2. MaxPooling layer with pool size 2
   3. Convolution layer with 32 filters and ‘relu’ activation function.
   4. MaxPooling layer with pool size 2
   5. Dropout layer with value as 0.2
   6. Convolution layer with 64 filters and ‘relu’ activation function.
   7. MaxPooling layer with pool size 2
   8. Dropout layer with value as 0.2
   9. Flatten Layer
   10. Dense layer with 500 neurons with ‘relu’ activation function.
   11. Dropout layer with value as 0.2
   12. Final dense layer with 13 neuron and ‘SoftMax’ activation function
6. Compiling the model with loss='categorical\_crossentropy', optimizer='adam',

metrics=['accuracy']

1. Training the model with batch size 100 for 20 epochs.
2. Predicting the model accuracy, we get 24%

Points to focus-

1. The deep learning model I am using is ‘Keras’ as it is more convenient and simpler than other models.
2. ‘Relu’ activation function is used as it is best suited in case of CNN as it ignores negative value which we do not want in any of our calculations.
3. Whenever we have categorical data we use 'categorical\_crossentropy' as the loss function.
4. The accuracy is only 24% because of several reason and but the major problem is the smaller number of the data size. If data size is increased, then may the accuracy of the model will increase.