

## Report

### Furniture Identification using Convolutional Neural Network (CNN)

By: Piyusha More      PRN: 20070243015

**Dataset:** A multi classification data was collected from Kaggle website, which consisted of 5 classes which are furniture types that are bed(0), chair(1), sofa(2), swivel chair(3) and table(4).



**Objective:** The objective of the project is to build a CNN model which will help to identify the types of furniture.

Training data	4014 images
Testing data	423 images
Validation data	10 images

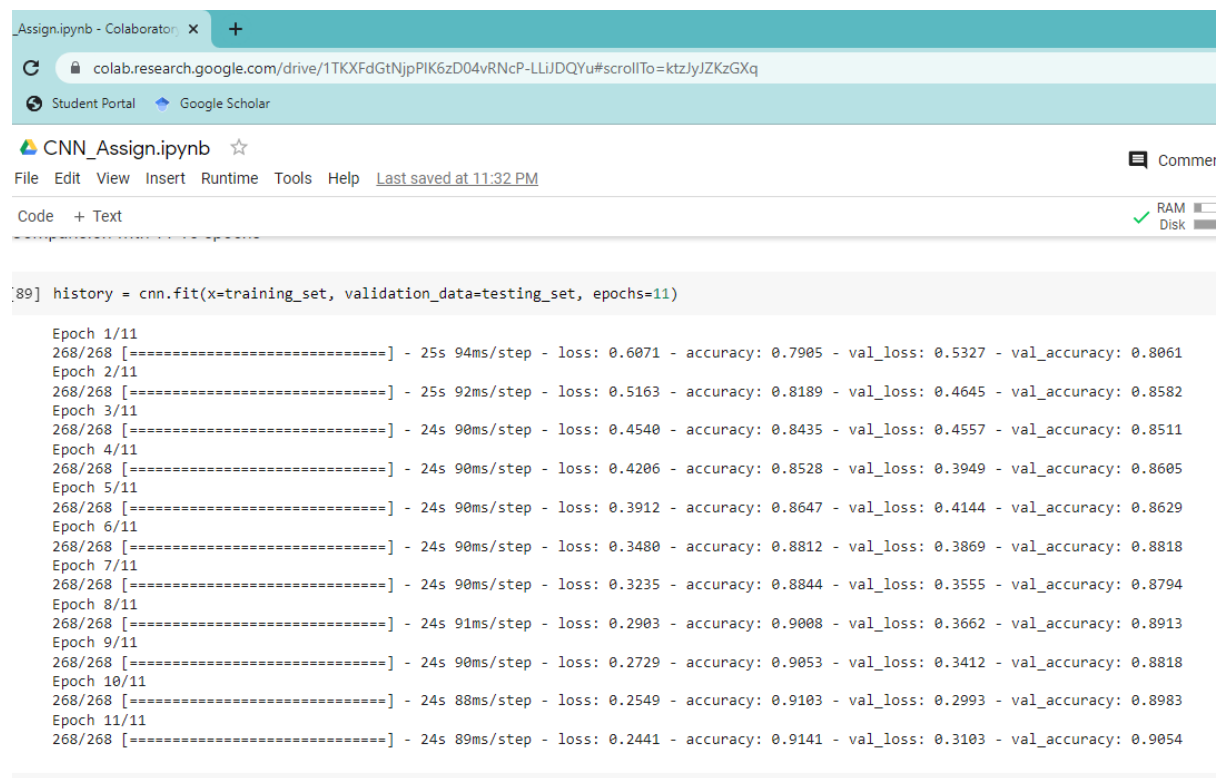
#### Steps used for building CNN model:

1. Data was loaded into training and testing data using ImageDataGenerator library.
2. Then initialise CNN model using tensorflow library.
3. Applied 2 convolutional layers with 32 filters.

4. Then applied 2 max pooling layers with size of 2 and strides of 2.
5. Applied activation function as 'relu'.
6. Applied flattening layer.
7. And finally, an output layer with 5 units because of 5 classes, by applying 'softmax' activation function.
8. The optimizer used for model is 'adam' and used categorical cross entropy for mutli-classification problem.

## Results:

I tried training the model using 3,5,11,15,25 epochs but got less cross\_entropy and higher accuracy for 11 epochs which is as follows.



```
[89] history = cnn.fit(x=training_set, validation_data=testing_set, epochs=11)

Epoch 1/11
268/268 [=====] - 25s 94ms/step - loss: 0.6071 - accuracy: 0.7905 - val_loss: 0.5327 - val_accuracy: 0.8061
Epoch 2/11
268/268 [=====] - 25s 92ms/step - loss: 0.5163 - accuracy: 0.8189 - val_loss: 0.4645 - val_accuracy: 0.8582
Epoch 3/11
268/268 [=====] - 24s 90ms/step - loss: 0.4540 - accuracy: 0.8435 - val_loss: 0.4557 - val_accuracy: 0.8511
Epoch 4/11
268/268 [=====] - 24s 90ms/step - loss: 0.4206 - accuracy: 0.8528 - val_loss: 0.3949 - val_accuracy: 0.8605
Epoch 5/11
268/268 [=====] - 24s 90ms/step - loss: 0.3912 - accuracy: 0.8647 - val_loss: 0.4144 - val_accuracy: 0.8629
Epoch 6/11
268/268 [=====] - 24s 90ms/step - loss: 0.3480 - accuracy: 0.8812 - val_loss: 0.3869 - val_accuracy: 0.8818
Epoch 7/11
268/268 [=====] - 24s 90ms/step - loss: 0.3235 - accuracy: 0.8844 - val_loss: 0.3555 - val_accuracy: 0.8794
Epoch 8/11
268/268 [=====] - 24s 91ms/step - loss: 0.2903 - accuracy: 0.9008 - val_loss: 0.3662 - val_accuracy: 0.8913
Epoch 9/11
268/268 [=====] - 24s 90ms/step - loss: 0.2729 - accuracy: 0.9053 - val_loss: 0.3412 - val_accuracy: 0.8818
Epoch 10/11
268/268 [=====] - 24s 88ms/step - loss: 0.2549 - accuracy: 0.9103 - val_loss: 0.2993 - val_accuracy: 0.8983
Epoch 11/11
268/268 [=====] - 24s 89ms/step - loss: 0.2441 - accuracy: 0.9141 - val_loss: 0.3103 - val_accuracy: 0.9054
```

Below, is table of accuracies and loss.

Training loss	<b>0.24</b>
Validation loss	<b>0.31</b>
Training accuracy	<b>0.91</b>
Validation accuracy	<b>0.90</b>

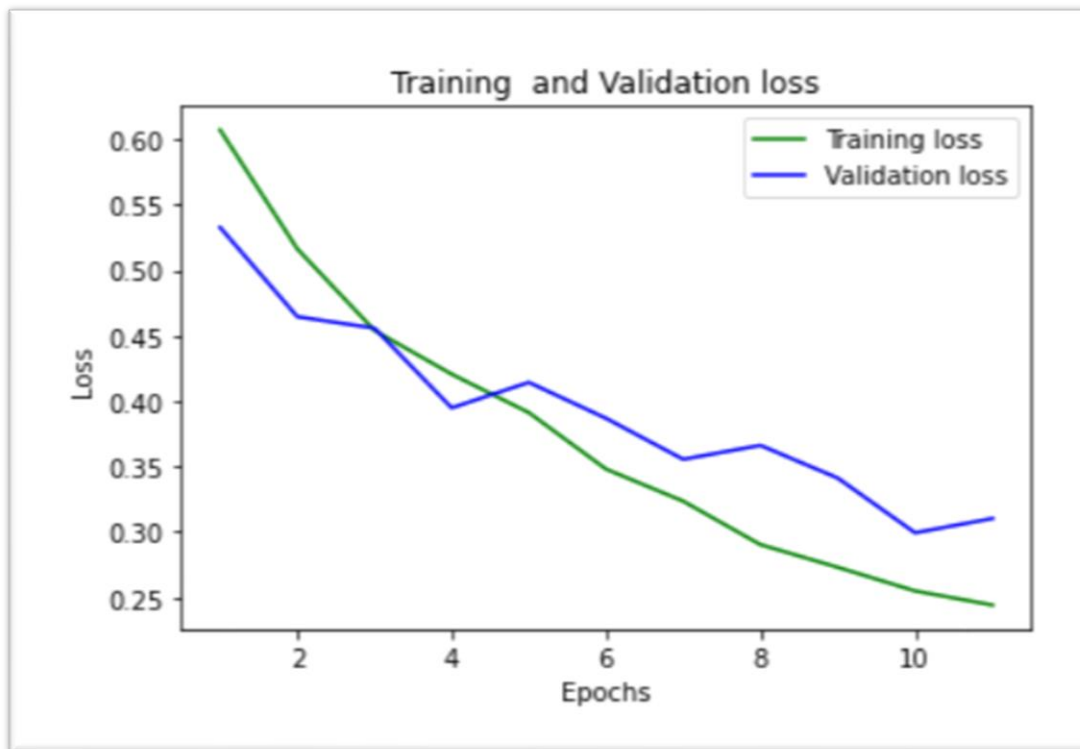


Image 1



Image 2

From above images we can get a clear idea about losses and accuracies, it could be easily interpreted from image 1 that training loss as well as validation loss are decreasing as epochs are increasing. And from image 2 this can be interpreted that training accuracy and validation accuracy are increasing along with epochs.

**Model Evaluation:**

For evaluating model 10 images were used, out of which 9 images were identified correctly and 1 image was misclassified.