LOVELY PROFESSIONAL UNIVERSITY

Classfication of Rock-Paper-Scissor using tensorflow

Names:

1- Mohammed Yahya Al Sultan

Registraion number: 11908184

2-Chanpreet Singh

Registraion number: 11801819

Section:

KM086

1-Introduction:

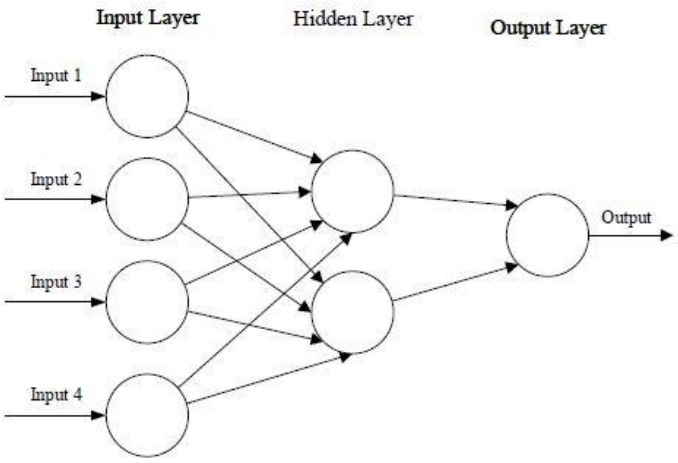
The field of deep learning has taken a dramatic twist in recent times, with the rise of the Artificial Neural Network (ANN). These biologically inspired computational models are able to far exceed the performance of previous forms of artificial intelligence in common machine learning tasks. One of the most impressive forms of ANN is Convolution Neural Network (CNN)[1].

CNN has played a critical role in today advanced techniques of classification. It has boosted the classification dramatically by mimicking the human brain. CNN consists of convolutions and pooling layers that are meant to reduce the complexity and size of the model and at the same time retaining the quality features that are desired for the seek of building the state-of-art model to classify different images.

By using the Powerful toles of CNN I could build a model that could easily classify three types of images that are Rock, Paper and Scissor. The Model architecture and further details are going to be explained in further sections.

2-Literature Review:

2.1 CNN architecture



CNN is primarily considering the input to be in the form of images. Therefore, the main focus is to design a model that could deal with this datatypes successfully . One of the key differences the neurons in the construction of the layers of CNN. These neurons are

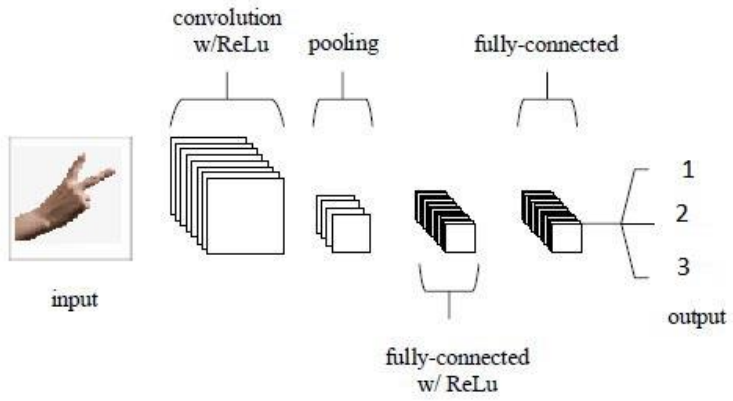
organized into three dimensions, the spatial dimensionality of the input (height and the width) and the depth. The depth does not refer to the total number of layers within the ANN, but the third dimension of an activation volume. CNN is unlike the ANNS in the essence of having an architecture of neurons in a certain layer connecting to a small region of the preceding layer.

Taking my model into consideration ,initially, the input value has the has the dimensionality of (None, 148, 148, 64)(where None represent the number of samples that could be brought into the model for training) . leading to an output the will have the dimensionality (leading

to a final output layer comprised of a dimensionality of (1 , 1 n)(where n represents the number of classes at the last layer)

2.2 Overall architecture

CNN is comprised of three types of layers. These are convolutional layers,pooling layers and fully-connected layers. When these layers are combined, a CNN architecture has been formed. A simplified CNN architecture for Rock Paper Scissor classification is illustrated in the following figure.



2.3 Classification optimizer used in the Model:

2.3.1 Gradient Descent:

Advantages:

Easy computation.

Easy to implement.

Easy to understand.

Disadvantages:

May trap at local minima.

Weights are changed after calculating gradient on the whole dataset. So, if the dataset is too large than this may take years to converge to the minima.

2.3.2 Stochastic Gradient Descent:

Advantages:

Frequent updates of model parameters hence, converges in less time.

Requires less memory as no need to store values of loss functions.

May get new minima's.

Disadvantages:

High variance in model parameters.

May shoot even after achieving global minima.

To get the same convergence as gradient descent needs to slowly reduce the value of learning rate.

2.3.3 Momentum:

Advantages:

Reduces the oscillations and high variance of the parameters.

Converges faster than gradient descent.

Disadvantages:

One more hyper-parameter is added which needs to be selected manually and accurately.

2.3.4 Adagrad

Advantages:

Learning rate changes for each training parameter.

Don’t need to manually tune the learning rate.

Able to train on sparse data.

Disadvantages:

Computationally expensive as a need to calculate the second order derivative.

The learning rate is always decreasing results in slow training.

2.3.5 Adam

Advantages:

The method is too fast and converges rapidly.

Rectifies vanishing learning rate, high variance.

Disadvantages:

Computationally costly[4].

2.4 Techniques used in the model:

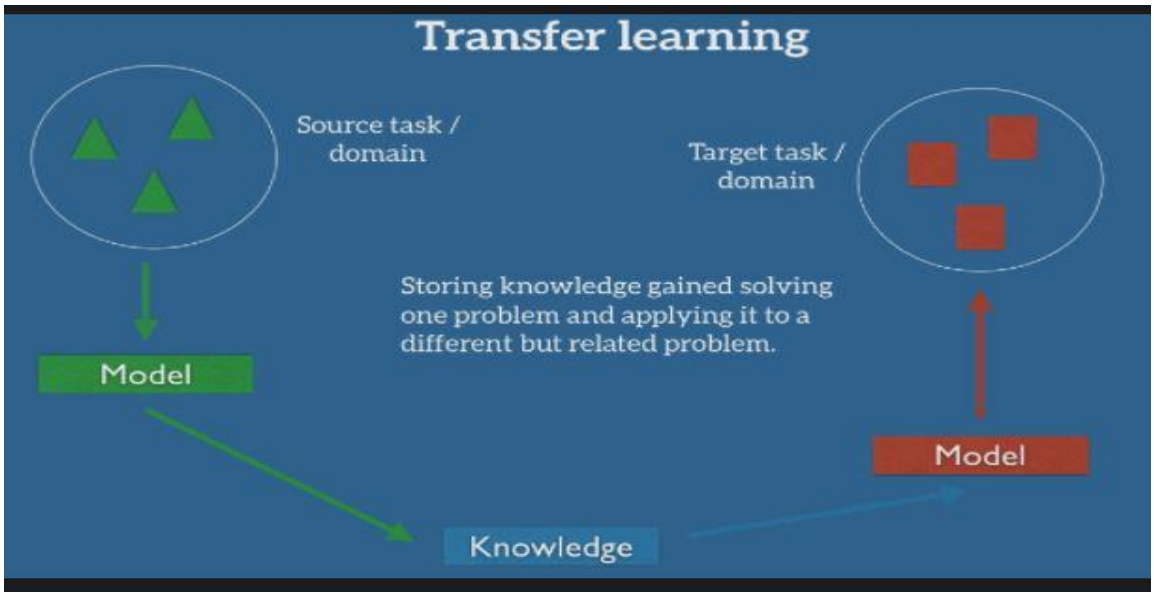
There are two main features I could further use in model to further increase the diversity and take the advantage of previous state of art model .The two techniques are Augmentation and Transfer learning:

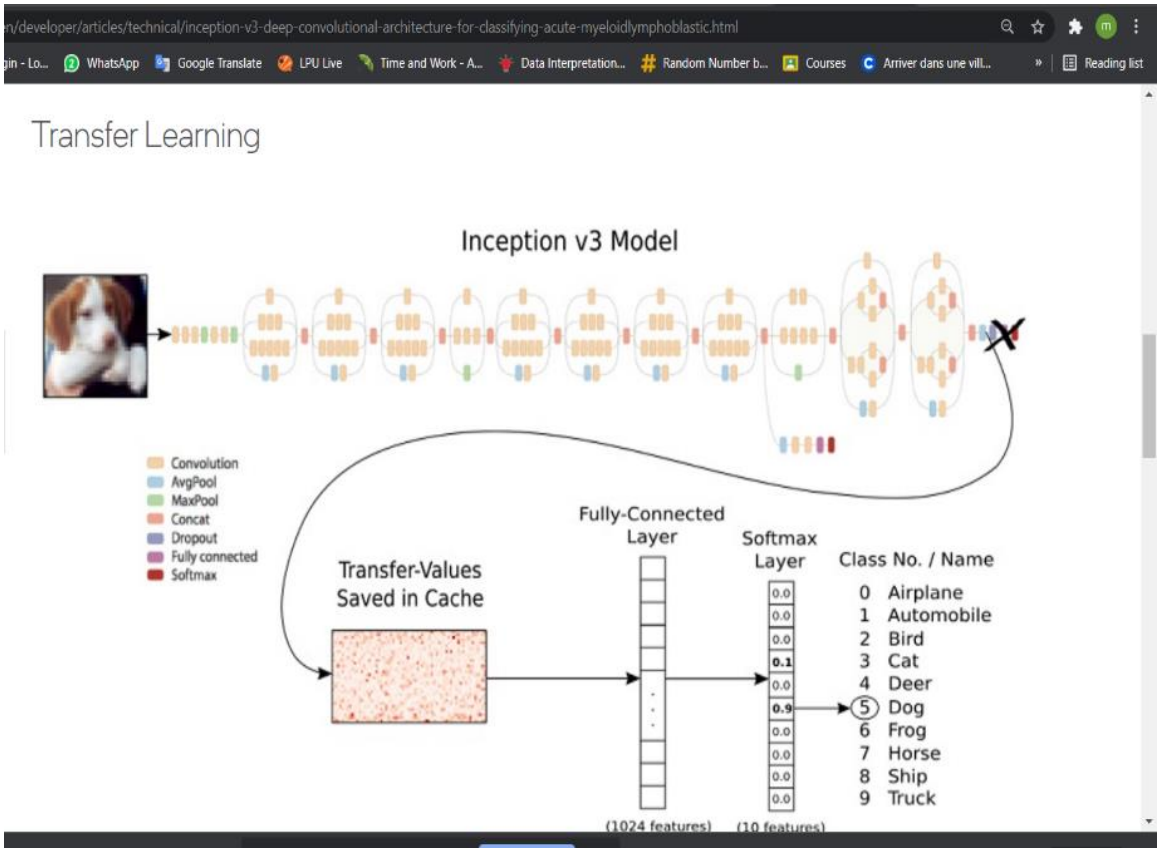
2.4.1 Augmentation:

Image class Augmentation is basically an artificially way to expand the size of the training data by creating modified versions of images in the dataset. This technique can be implemented by introducing Horizontal and Vertical Shift Augmentation, use shift, flip, brightness, and zoom image data augmentation[5].

2.4.1 Transfer Learning:

Transfer learning help us to use a previously implemented model by freezing some of its layer and add our model at the bottom layer. This can add many features that are costly computed ,also introduce some sort of diversity that will eventually lead to less overfitting. It is always recommendable to use a model that is related to the domain of the problem in order to get best possible accuracy[6].





3-Datasets:

The project that is being assigned to me is classification of images using TensorFlow. The dataset I worked with is about classification of three types of images which are rock, paper and scissor[2]

The dataset is being created with the help of CGI(Computer Generated Imagery). It is easier and more affordable to build a model using this technique, as it would be a tedious task to fine people of different categories, such as different colour of skin and different positions of hands and also different hand size, that would satisfy the model of being diversed[3].Some the images that are being created with the help of CGI are as follow:

Rock

Paper

Scissor

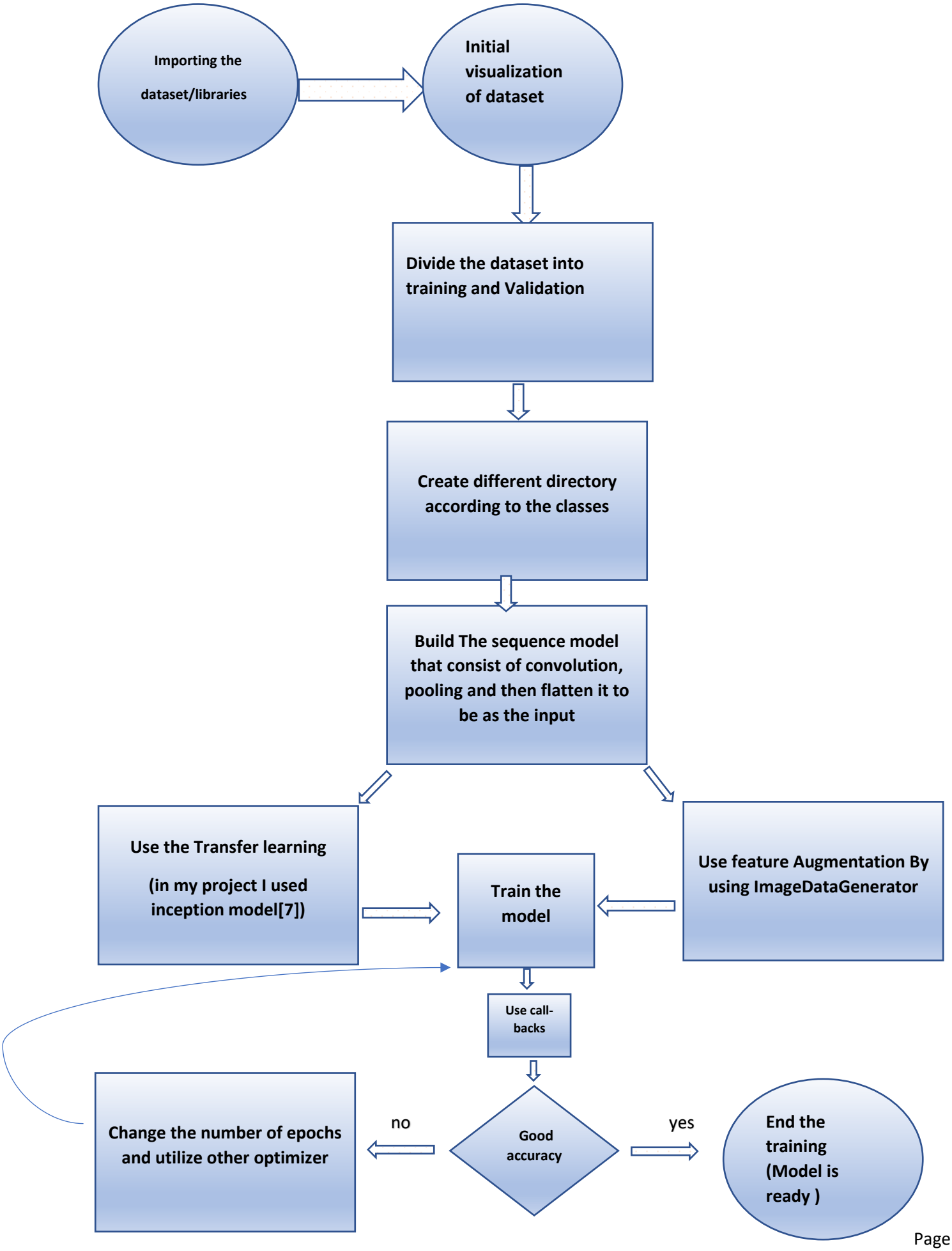
The contents of the dataset are as follow:

Split	Examples
Test	372
Train	2520

The overall size of this dataset is 220 Mb.

Here is other variation of the dataset that is being created with the help of CGI:

4-Proposed Architecture:



5-Result and experiment analysis:

1-The number of epochs was not a main triggering for boosting the performance of the model. In most of the optimization functions ,the accuracy started to saturate after the fifth epoch.

2-By using the call-backs the computational time has reduced and thus it was a useful tool in order to reduce the time for training.

3-Adam,Adammax optimizer have reached the highest accuracy and could classify the images quiet well at testing time.

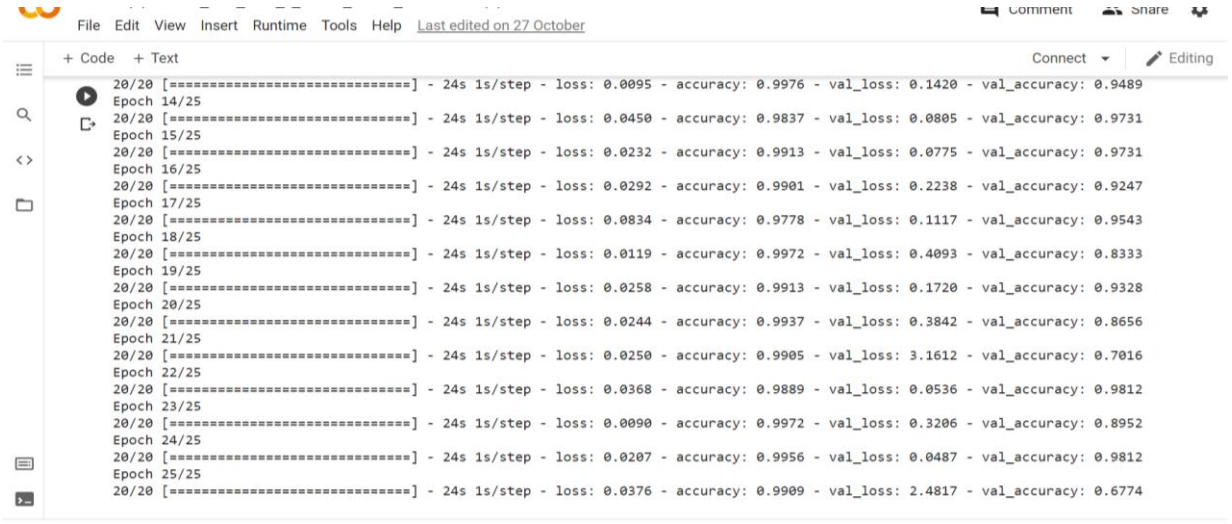
4-Stochastic Gradient Decent (SGD) was the most inferior classifier as it showed a low accuracy as well as poor classification at run time.

5-Using the inception model for Transfer learning was not a good option as this model was not related to the classes given in the classification model.

6-Screenshots:

4.1-Using inception model through tranfer learning.

The below image shows the training of the model and using the inception model through transfer learning .The training is not stable with the divergence between the two Models(Rock-Paper-Scissor and inception).



Clear overfitting is being displayed and was classifying most the images incorrectly

File Edit View Insert Runtime Tools Help Last edited on 27 October

+ Code + Text Connect Editing

```
Epoch 2/25
20/20 [==============================] - 23s 1s/step - loss: 1.0918 - accuracy: 0.3833 - val_loss: 1.6845 - val_accuracy: 0.3333
Epoch 3/25
20/20 [==============================] - 23s 1s/step - loss: 1.0897 - accuracy: 0.4210 - val_loss: 0.8006 - val_accuracy: 0.6344
Epoch 4/25
20/20 [==============================] - 23s 1s/step - loss: 0.9566 - accuracy: 0.5484 - val_loss: 0.7955 - val_accuracy: 0.7527
Epoch 5/25
20/20 [==============================] - 23s 1s/step - loss: 0.8229 - accuracy: 0.6548 - val_loss: 0.6130 - val_accuracy: 0.6667
Epoch 6/25
20/20 [==============================] - 23s 1s/step - loss: 0.6546 - accuracy: 0.7143 - val_loss: 0.3745 - val_accuracy: 0.7957
Epoch 7/25
20/20 [==============================] - 23s 1s/step - loss: 0.6456 - accuracy: 0.7298 - val_loss: 0.1314 - val_accuracy: 1.0000
Epoch 8/25
20/20 [==============================] - 23s 1s/step - loss: 0.4599 - accuracy: 0.8060 - val_loss: 0.4015 - val_accuracy: 0.7016
Epoch 9/25
20/20 [==============================] - 23s 1s/step - loss: 0.4737 - accuracy: 0.8000 - val_loss: 0.2550 - val_accuracy: 0.8548
Epoch 10/25
20/20 [==============================] - 23s 1s/step - loss: 0.3367 - accuracy: 0.8595 - val_loss: 0.1341 - val_accuracy: 0.9624
Epoch 11/25
20/20 [==============================] - 23s 1s/step - loss: 0.2618 - accuracy: 0.9008 - val_loss: 0.0603 - val_accuracy: 0.9839
Epoch 12/25
20/20 [==============================] - 23s 1s/step - loss: 0.3153 - accuracy: 0.8810 - val_loss: 0.0608 - val_accuracy: 0.9624
```

```
import matplotlib.pyplot as plt
acc = history.history['accuracy']
```

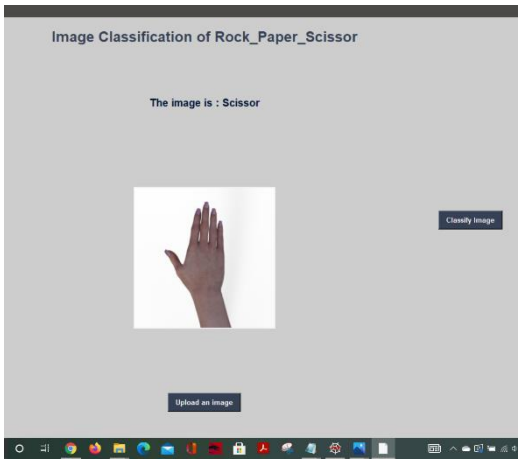
Type here to search ENG 17:02 31-10-2021

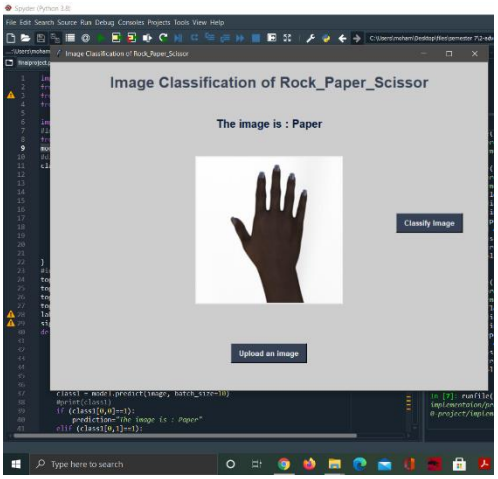
+ Code + Text Connect Editing

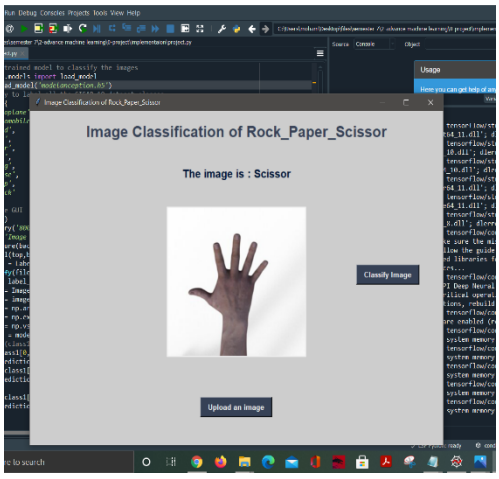
```
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()


plt.show()
```


```
<Figure size 432x288 with 0 Axes>
```

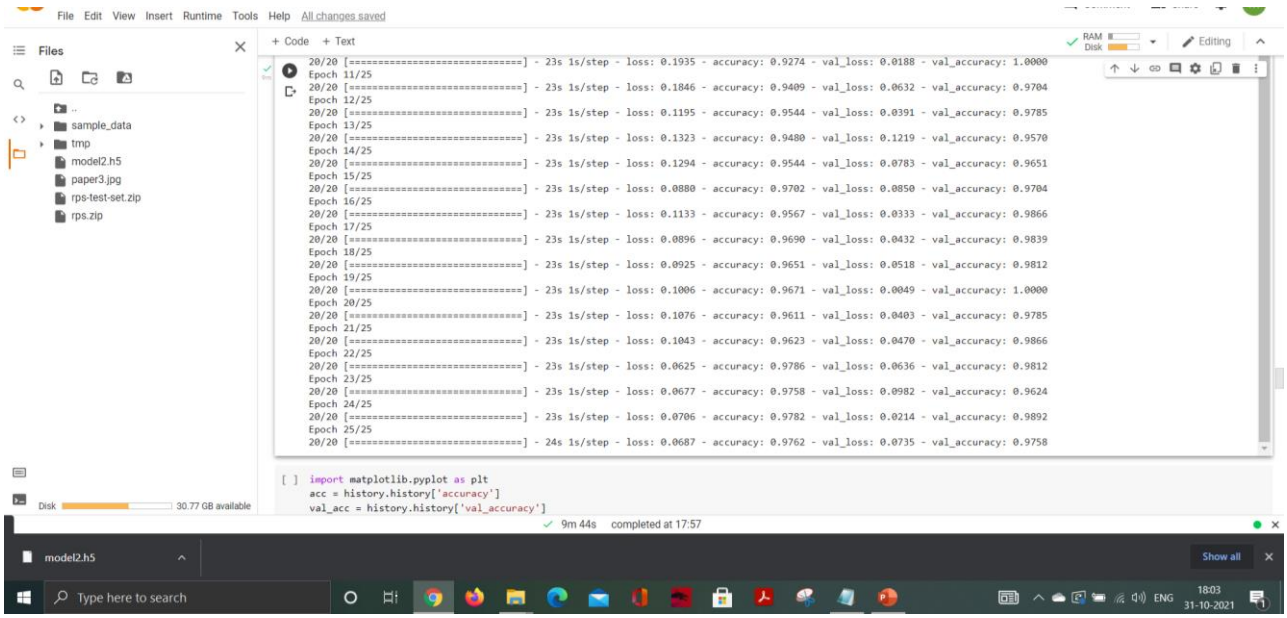
Type here to search ENG 17:02 31-10-2021

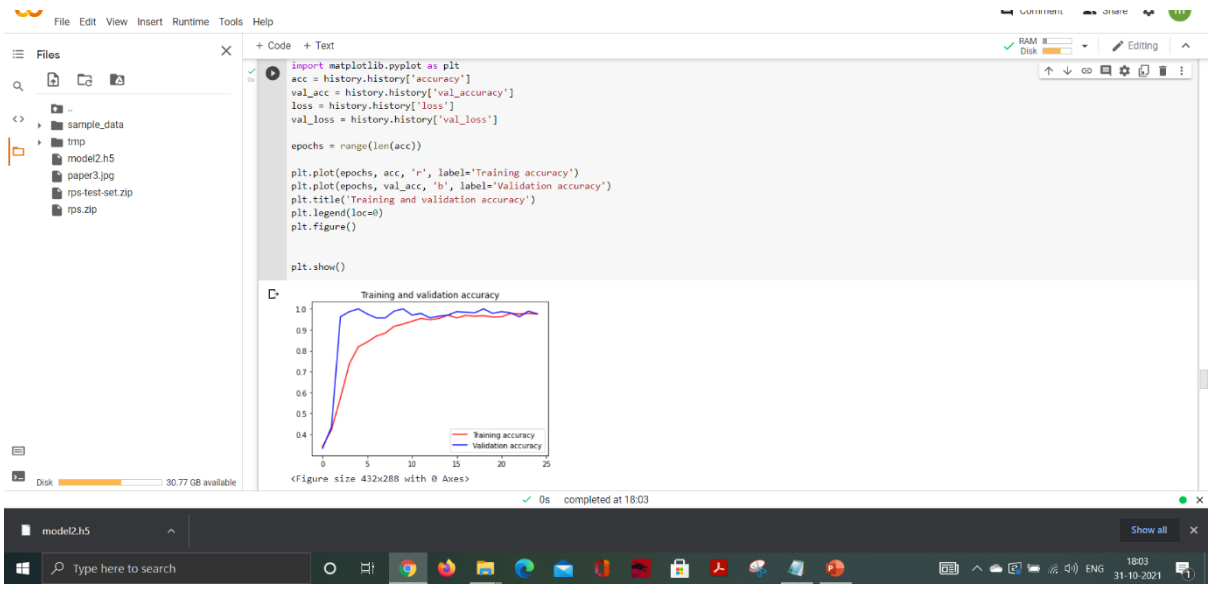


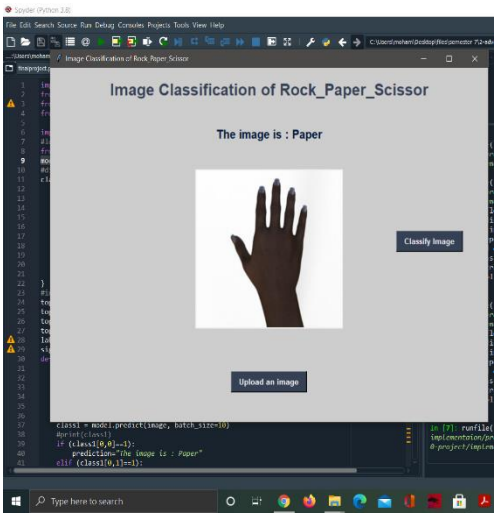


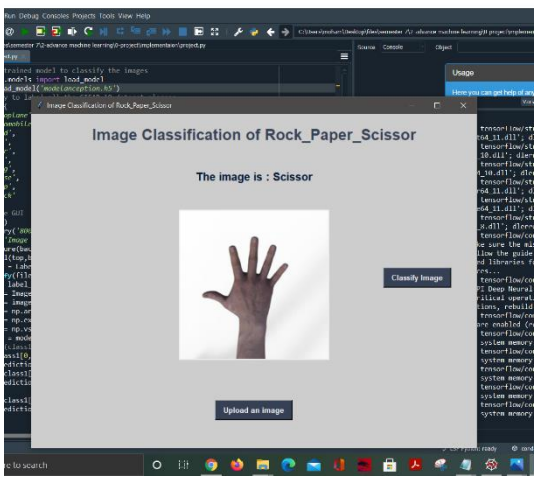


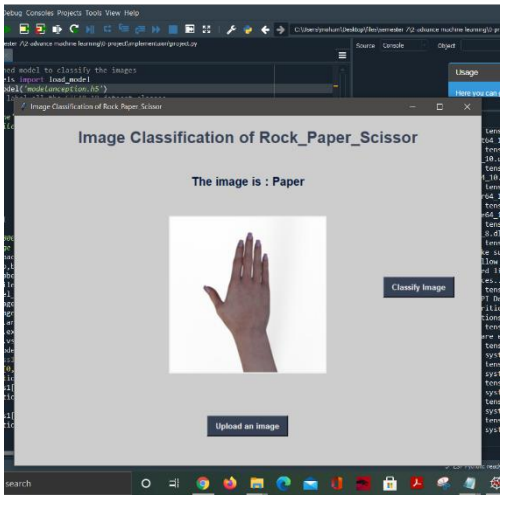
4.1-Using Adam Optimizer.



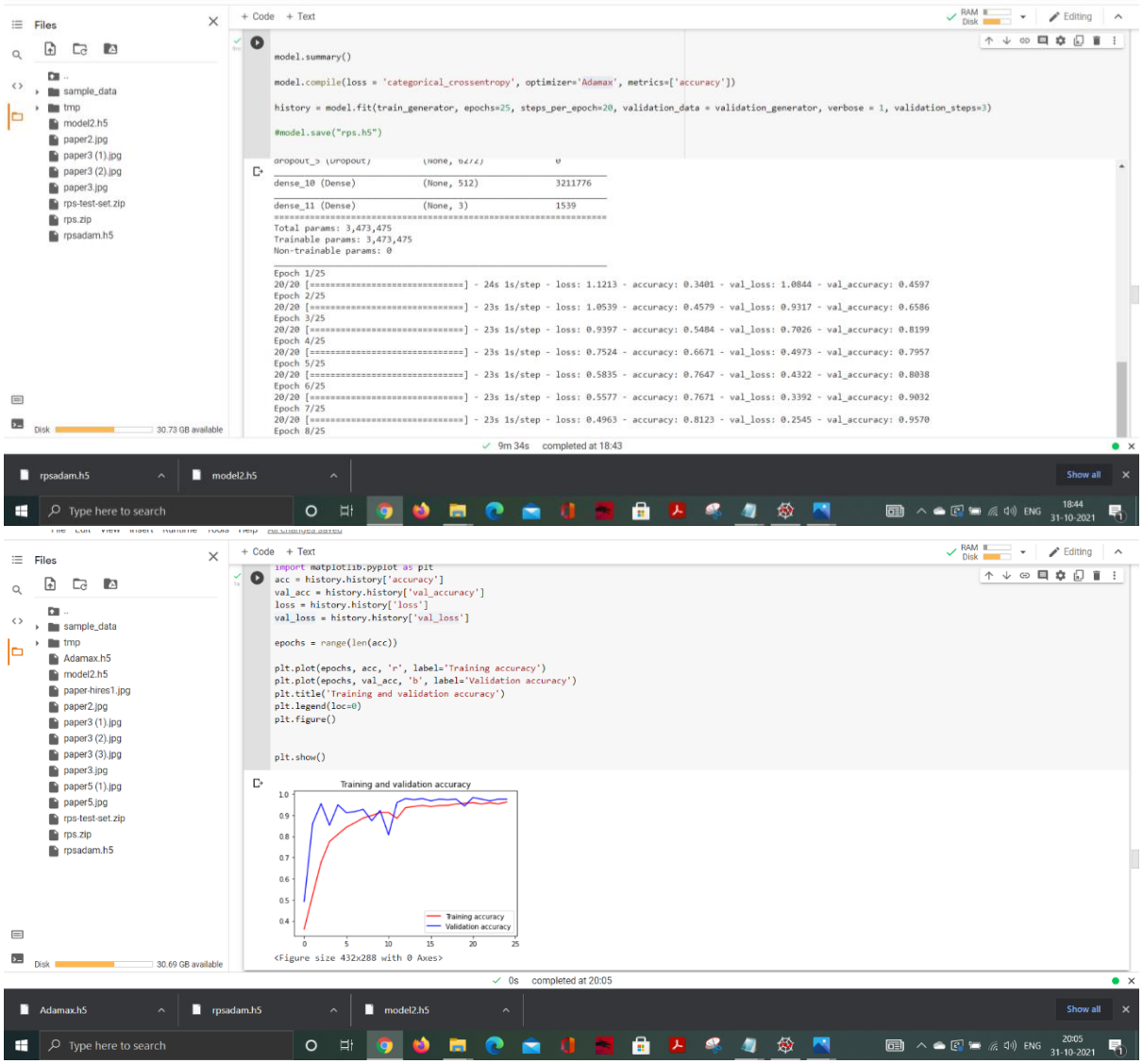


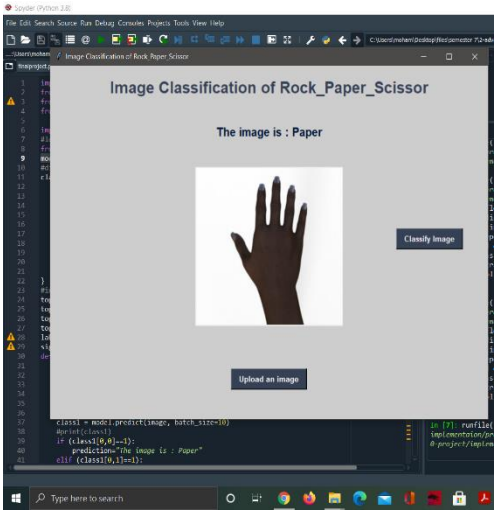


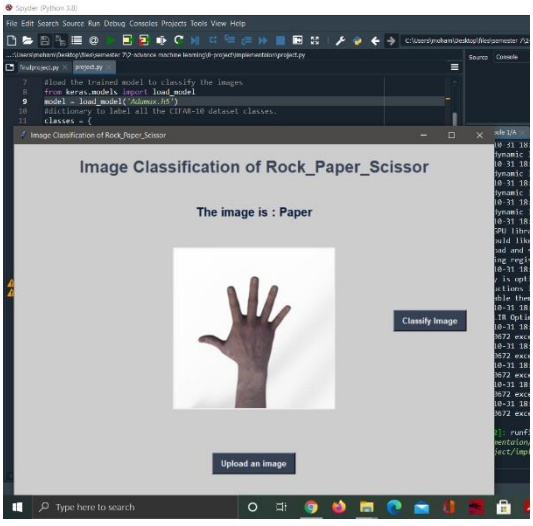


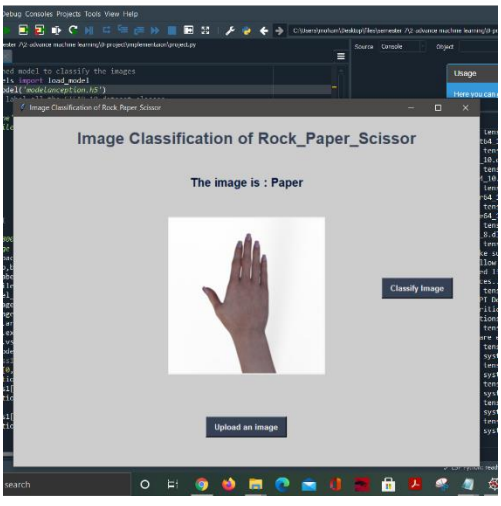


4.1-Using Adammax Optimizer.

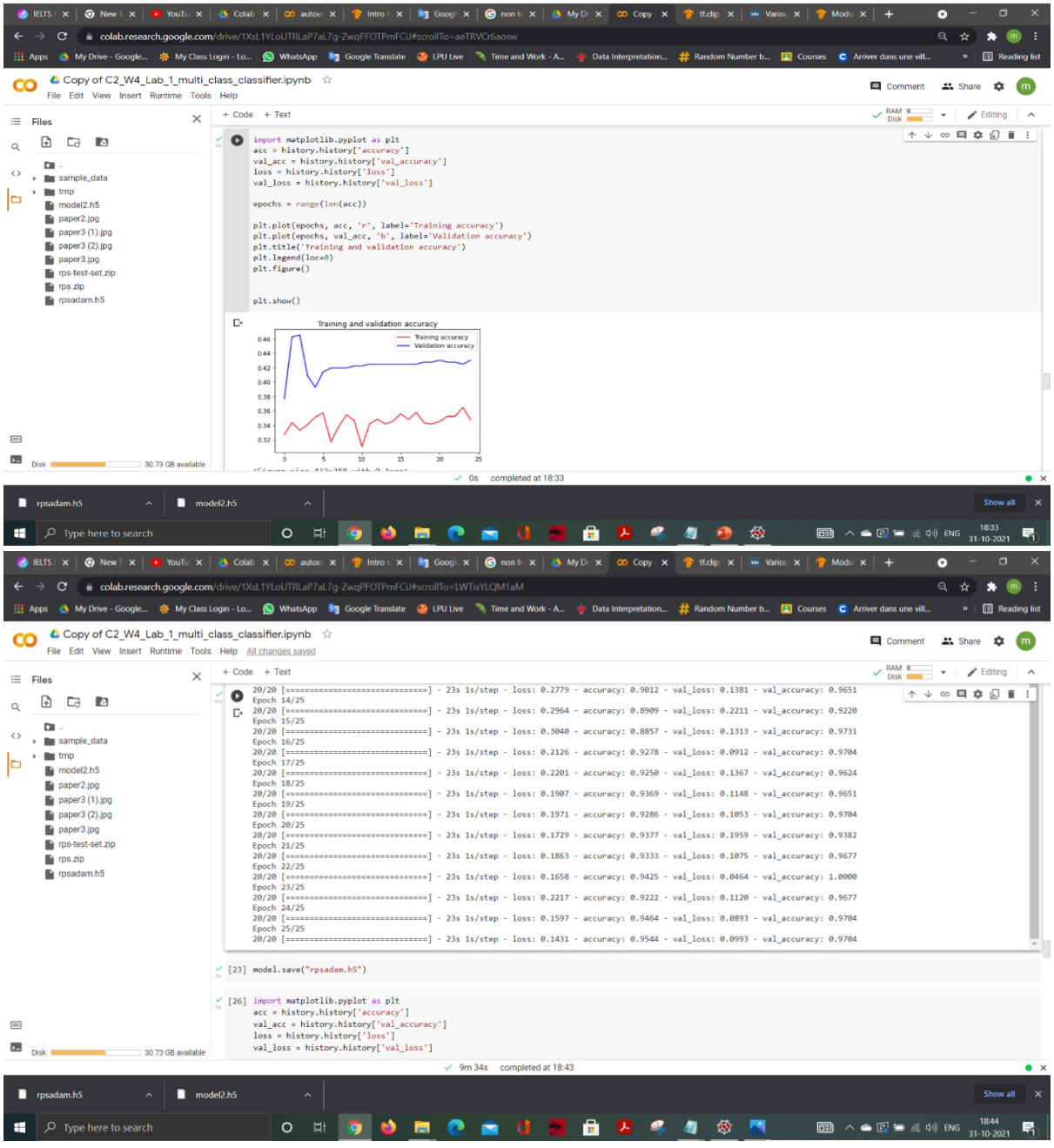


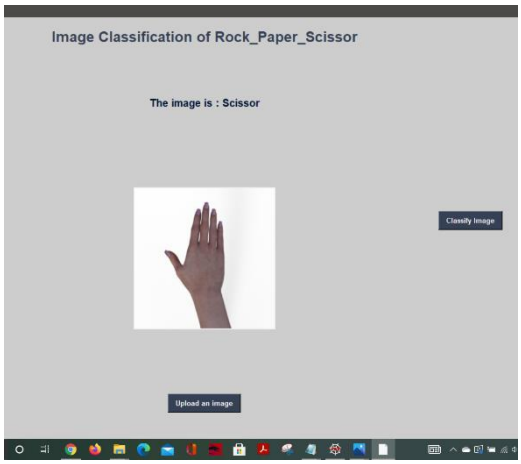


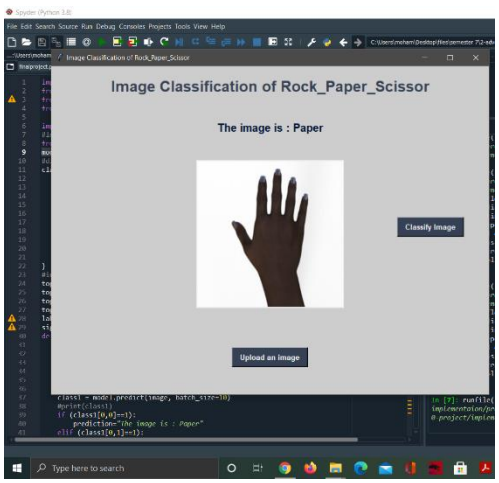


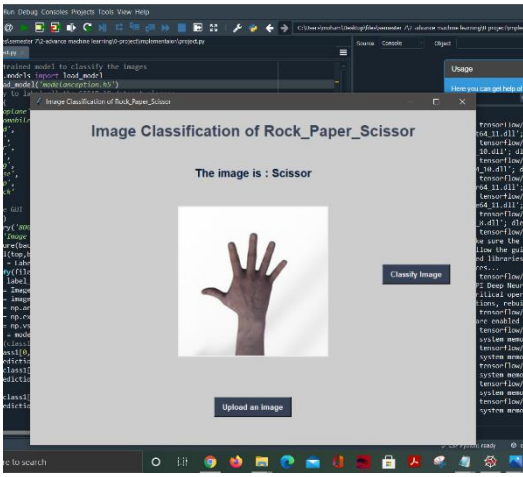


4.1-Using Gradient Decent Optimizer.









7-Conclusion and future scope:

Convolution neural network has played a critical role in today improvement of classification by mimicking the human brain. It is structured with layers that are stacked above each other.

These layers are consisting of convolution layers that extract the most important features. Hence, reduce the complexity of the model and boost it in terms of accuracy .Next ,is the pooling and last the flatten layer that convert the input into one dimension which could be easily fed and processed through the built model.

Everyday new algorithms are created and further improved to enhance the capability of the overall of the model .Concretely, it is a good practice to choose the optimization function the best suits the problem given.

8-References:

1-Keiron O'Shea1 and Ryan Nash2

2- https://www.tensorflow.org/datasets/catalog/rock_paper_scissors

3- Markus Utke∗ , Saman Zadtootaghaj∗ , Steven Schmidt∗ , Sebastian M ¨oller∗† ∗Quality and Usability Lab, Technische Universitt Berlin, Germany, markus.utke@campus.tu-berlin.de

4-https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

5-https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

6-https://ruder.io/transfer-learning/

7-https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202