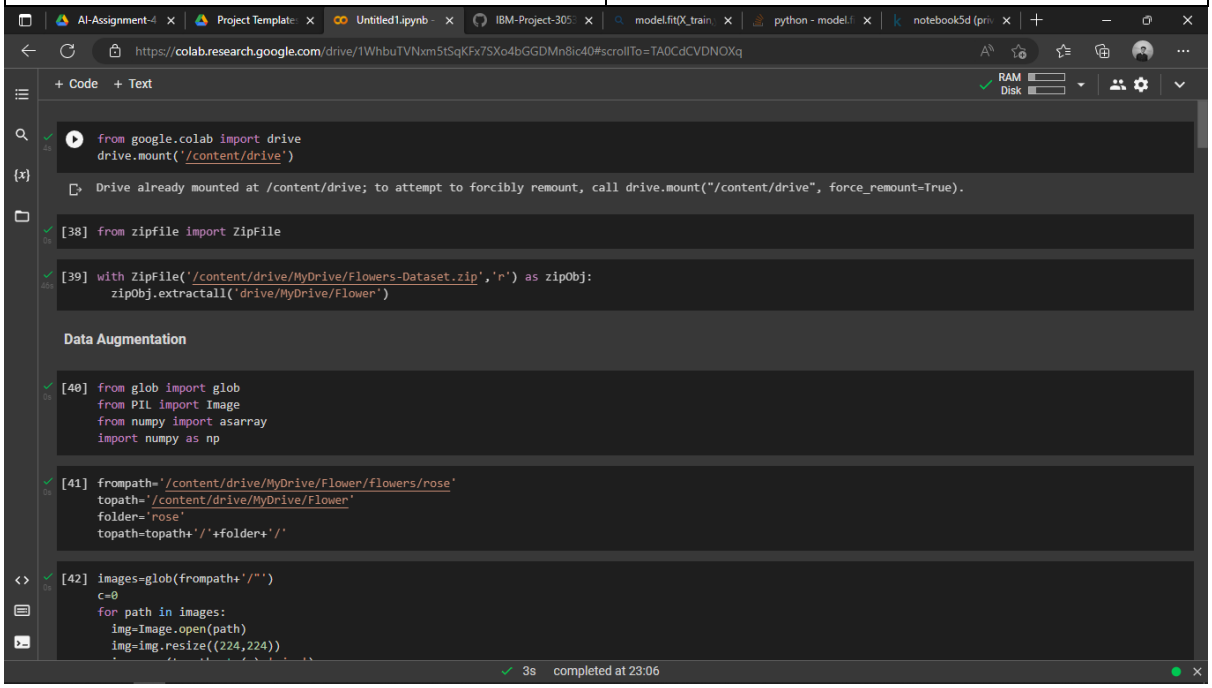
Assignment -3
Build cnn model for classification of flowers

Assignment Date	12 November 2022
Student Name	Mr.S.Dhilip Kumar
Student Roll Number	E1194015
Maximum Marks	2 Marks



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[38] from zipfile import ZipFile
```

```
[39] with ZipFile('/content/drive/MyDrive/Flowers-Dataset.zip','r') as zipObj:
       zipObj.extractall('drive/MyDrive/Flower')
```

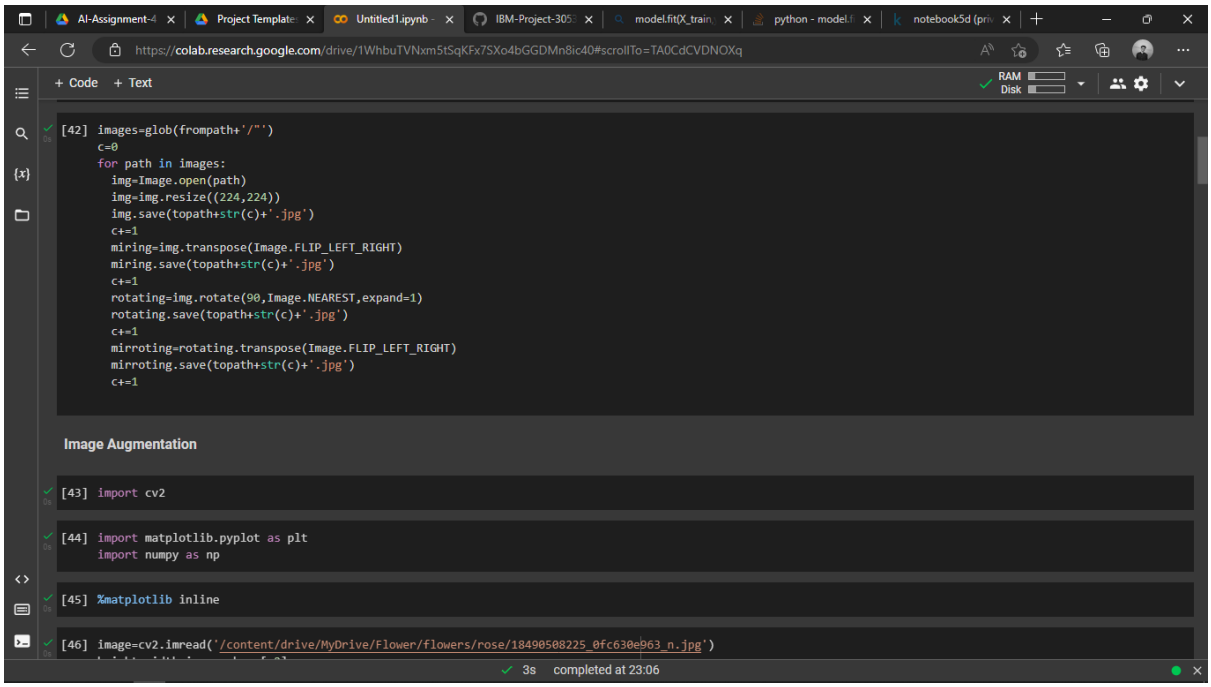
Data Augmentation

```
[40] from glob import glob
     from PIL import Image
     from numpy import asarray
     import numpy as np
```

```
[41] frompath='/content/drive/MyDrive/Flower/flowers/rose'
     topath='/content/drive/MyDrive/Flower'
     folder='rose'
     topath=topath+'/'+folder+'/'
```

```
[42] images=glob(frompath+'/*')
     c=0
     for path in images:
       img=Image.open(path)
       img=img.resize((224,224))
```

3s completed at 23:06



```
[42] images=glob(frompath+'/*')
     c=0
     for path in images:
       img=Image.open(path)
       img=img.resize((224,224))
       img.save(topath+str(c)+'.jpg')
       c+=1
       miring=img.transpose(Image.FLIP_LEFT_RIGHT)
       miring.save(topath+str(c)+'.jpg')
       c+=1
       rotating=img.rotate(90,Image.NEAREST,expand=1)
       rotating.save(topath+str(c)+'.jpg')
       c+=1
       mirroting=rotating.transpose(Image.FLIP_LEFT_RIGHT)
       mirroting.save(topath+str(c)+'.jpg')
       c+=1
```

Image Augmentation

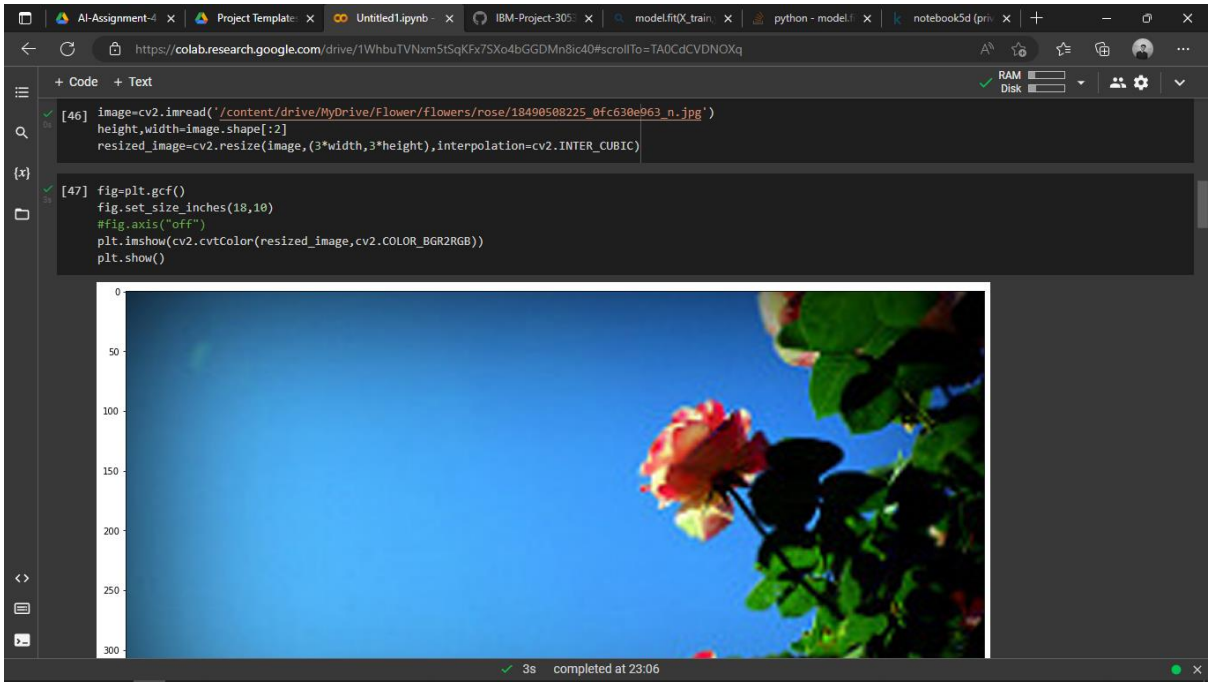
```
[43] import cv2
```

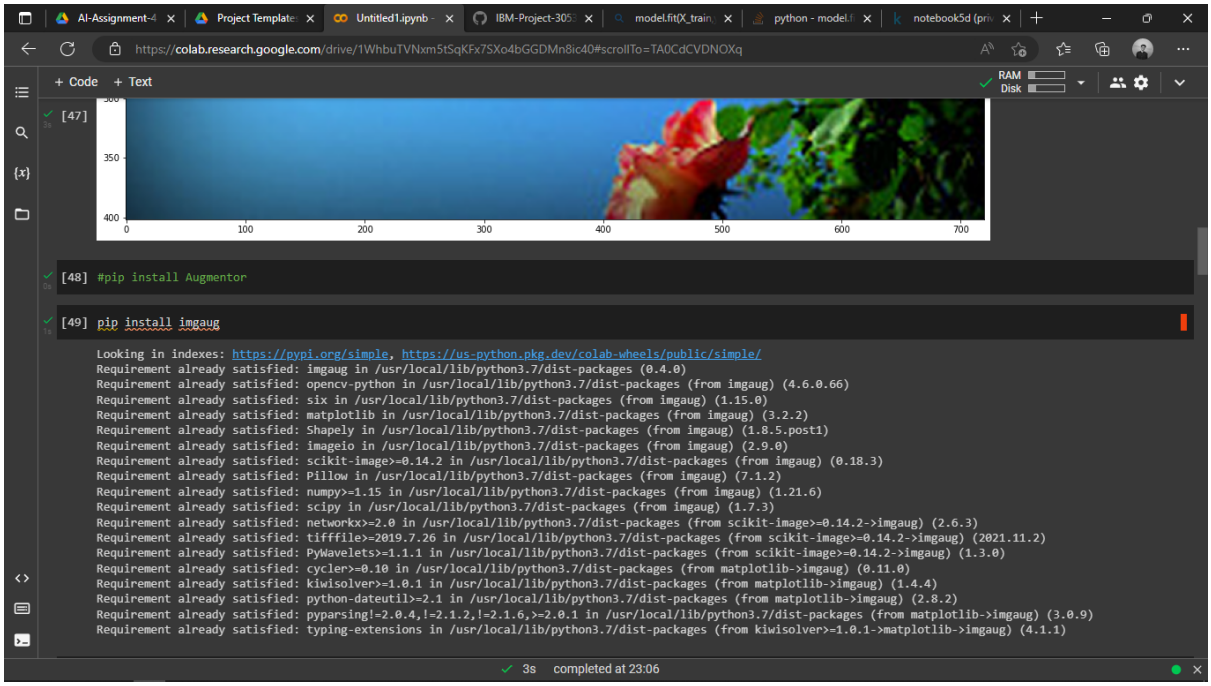
```
[44] import matplotlib.pyplot as plt
     import numpy as np
```

```
[45] %matplotlib inline
```

```
[46] image=cv2.imread('/content/drive/MyDrive/Flower/flowers/rose/18490508225_0fc630e963_n.jpg')
```

3s completed at 23:06


Untitled1.ipynb
https://colab.research.google.com/drive/1WhbuTVNxm5tSqKFx7SXo4bGGDMn8ic40#scrollTo=TA0CdCVDNOXq
+ Code + Text
RAM
Disk
[46] image=cv2.imread('/content/drive/MyDrive/Flower/flowers/rose/18490508225_0fc630e963_n.jpg')
height,width=image.shape[:2]
resized_image=cv2.resize(image,(3*width,3*height),interpolation=cv2.INTER_CUBIC)
[47] fig=plt.gcf()
fig.set_size_inches(18,10)
#fig.axis("off")
plt.imshow(cv2.cvtColor(resized_image,cv2.COLOR_BGR2RGB))
plt.show()
0
50
100
150
200
250
300
3s completed at 23:06


Untitled1.ipynb
https://colab.research.google.com/drive/1WhbuTVNxm5tSqKFx7SXo4bGGDMn8ic40#scrollTo=TA0CdCVDNOXq
+ Code + Text
RAM
Disk
[47]
350
400
0
100
200
300
400
500
600
700
[48] #pip install Augmentor
[49] pip install imgaug
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: imgaug in /usr/local/lib/python3.7/dist-packages (0.4.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (from imgaug) (4.6.0.66)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.15.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from imgaug) (3.2.2)
Requirement already satisfied: Shapely in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.8.5.post1)
Requirement already satisfied: imageio in /usr/local/lib/python3.7/dist-packages (from imgaug) (2.9.0)
Requirement already satisfied: scikit-image>=0.14.2 in /usr/local/lib/python3.7/dist-packages (from imgaug) (0.18.3)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from imgaug) (7.1.2)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.21.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.7.3)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (2.6.3)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (2021.11.2)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (1.4.4)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (3.0.9)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib->imgaug) (4.1.1)
3s completed at 23:06

```
[50] pip install ipyplot
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: ipyplot in /usr/local/lib/python3.7/dist-packages (1.1.1)
Requirement already satisfied: shortuuid in /usr/local/lib/python3.7/dist-packages (from ipyplot) (1.0.10)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from ipyplot) (7.1.2)
Requirement already satisfied: IPython in /usr/local/lib/python3.7/dist-packages (from ipyplot) (7.9.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from ipyplot) (1.21.6)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (57.4.0)
Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (0.18.1)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (4.8.0)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (5.1.1)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (2.0.10)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (2.6.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (0.2.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (0.7.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dist-packages (from jedi>=0.10->IPython->ipyplot) (0.8.3)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.1.0,>=2.0.0->IPython->ipyplot) (0.2.5)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.1.0,>=2.0.0->IPython->ipyplot) (1.15.0)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-packages (from pexpect->IPython->ipyplot) (0.7.0)
```

```
[51] import imageio
     import ipyplot
     import imgaug as ia
     import imgaug.augmenters as iaa
```

```
[52] input_img=imageio.imread('/content/drive/MyDrive/Flower/flowers/rose/20409866779_ac473f55e0_m.jpg')
```

```
[53] hflip=iaa.Fliplr(p=1.0)
     input_hf=hflip.augment_image(input_img)
```

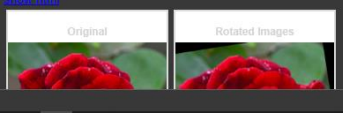
3s completed at 23:06

```
[54] vflip=iaa.Flipud(p=1.0)
     input_vf=vflip.augment_image(input_img)
     image_list=[input_img,input_hf,input_vf]
     labels=['Original','Horizontally flipped','Vertically flipped']
     ipyplot.plot_images(image_list,labels=labels,img_width=180)
```

show input


```
[55] rot1=iaa.Affine(rotate=(-30,30))
     input_rot1=rot1.augment_image(input_img)
     images_list=[input_img,input_rot1]
     labels=['Original','Rotated Images']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

show input



3s completed at 23:06

```
[55] rot1=iaa.Affine(rotate=(-30,30))
     input_rot1=rot1.augment_image(input_img)
     images_list=[input_img,input_rot1]
     labels=['Original','Rotated Images']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

show html

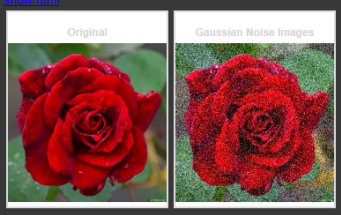

```
[56] noise=iaa.AdditiveGaussianNoise(10,40)
     input_noise=noise.augment_image(input_img)
     images_list=[input_img,input_noise]
     labels=['Original','Gaussian Noise Images']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

show html

3s completed at 23:06

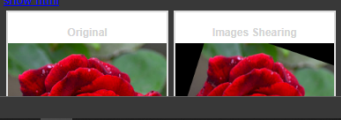

```
[56] noise=iaa.AdditiveGaussianNoise(10,40)
     input_noise=noise.augment_image(input_img)
     images_list=[input_img,input_noise]
     labels=['Original','Gaussian Noise Images']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

show html




```
[57] shear=iaa.Affine(shear=(-40,40))
     input_shear=shear.augment_image(input_img)
     images_list=[input_img,input_shear]
     labels=['Original','Images Shearing']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

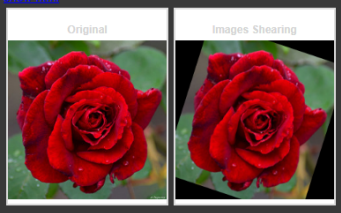
show html



3s completed at 23:06

```
[57] shear=iaa.Affine(shear=(-40,40))
     input_shear=shear.augment_image(input_img)
     images_list=[input_img,input_shear]
     labels=['Original','Images Shearing']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

show input



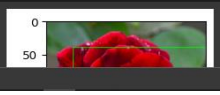

```
[58] contrast=iaa.GammaContrast((0.5, 2.0))
     contrast_sig=iaa.SigmoidContrast(gain=(5,10),cutoff=(0.4,0.6))
     contrast_lin=iaa.LinearContrast((0.6,0.4))
     input_contrast=contrast.augment_image(input_img)
     sigmoid_contrast=contrast_sig.augment_image(input_img)
     linear_contrast=contrast_lin.augment_image(input_img)
     images_list=[input_img,input_contrast,sigmoid_contrast,linear_contrast]
     labels=['Original','Gamma Contrast','SigmoidContrast','LinearContrast']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

3s completed at 23:06


```
[58] contrast=iaa.GammaContrast((0.5, 2.0))
     contrast_sig=iaa.SigmoidContrast(gain=(5,10),cutoff=(0.4,0.6))
     contrast_lin=iaa.LinearContrast((0.6,0.4))
     input_contrast=contrast.augment_image(input_img)
     sigmoid_contrast=contrast_sig.augment_image(input_img)
     linear_contrast=contrast_lin.augment_image(input_img)
     images_list=[input_img,input_contrast,sigmoid_contrast,linear_contrast]
     labels=['Original','Gamma Contrast','SigmoidContrast','LinearContrast']
     ipyplot.plot_images(images_list,labels=labels,img_width=180)
```

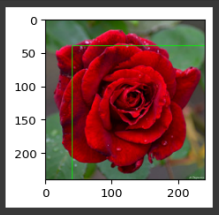

show input


```
[59] from imgaug.augmentables.bbs import BoundingBox,BoundingBoxesOnImage
     bbs=BoundingBoxesOnImage([BoundingBox(x1=40,x2=550,y1=40,y2=780)],shape=input_img.shape)
     ia.imshow(bbs.draw_on_image(input_img))
```



3s completed at 23:06

```
[59] from imgaug.augmentables.bbs import BoundingBox,BoundingBoxesOnImage
     bbs=BoundingBoxesOnImage([BoundingBox(x1=40,x2=550,y1=40,y2=780)],shape=input_img.shape)
     ia.imshow(bbs.draw_on_image(input_img))
```



Add layers(Convolution,Maxpooling,Flatten,Dense-(hidden layer),output)

```
[60] from tensorflow.keras.datasets import mnist
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Conv2D
     from tensorflow.keras.layers import MaxPool2D
     from tensorflow.keras.layers import Flatten
     from tensorflow.keras.layers import Dropout
     from tensorflow.keras.layers import Dense
```

```
[75] (X_train,y_train),(X_test,y_test)=mnist.load_data()
```

3s completed at 23:06

```
[62] X_train=X_train.reshape((X_train.shape[0],X_train.shape[1],X_train.shape[2],1))
     X_test=X_test.reshape((X_test.shape[0],X_test.shape[1],X_test.shape[2],1))
```

```
[63] print(X_train.shape)
     print(X_test.shape)

     (60000, 28, 28, 1)
     (10000, 28, 28, 1)
```

```
[64] X_train=X_train/255
     X_test=X_test/255
```

```
[65] model=Sequential()
```

```
[66] model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
```

```
[67] model.add(MaxPool2D(2,2))
```

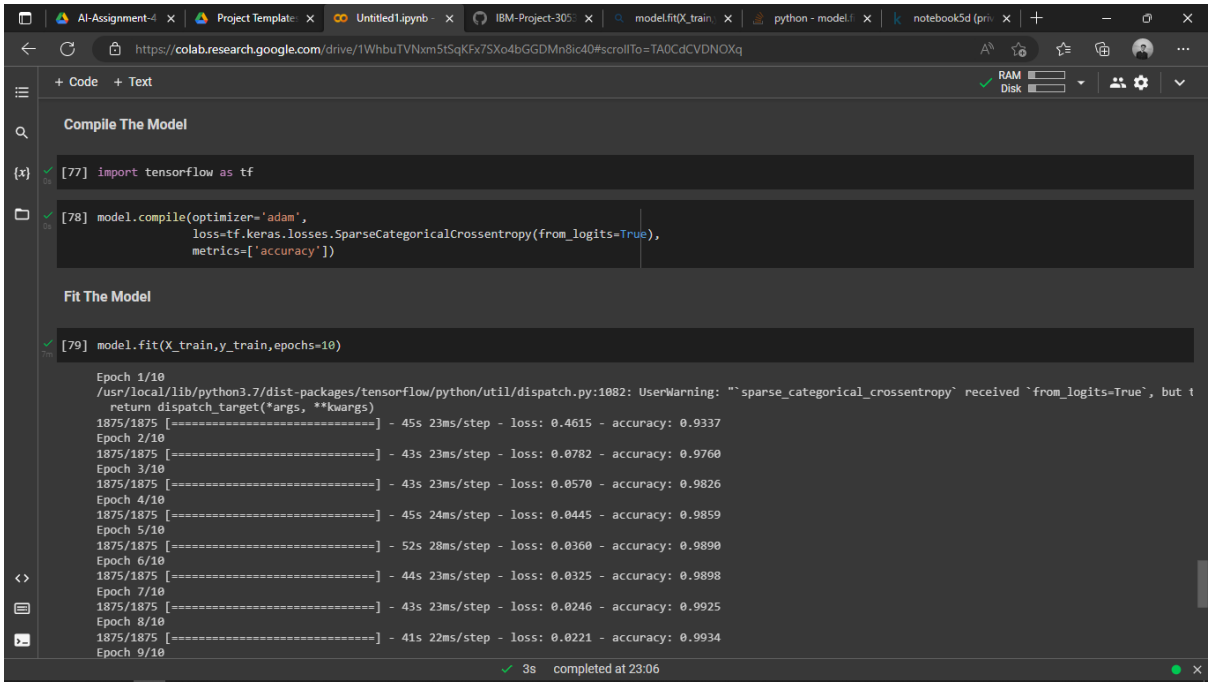
```
[68] model.add(Flatten())
     model.add(Dense(100,activation='relu'))
```

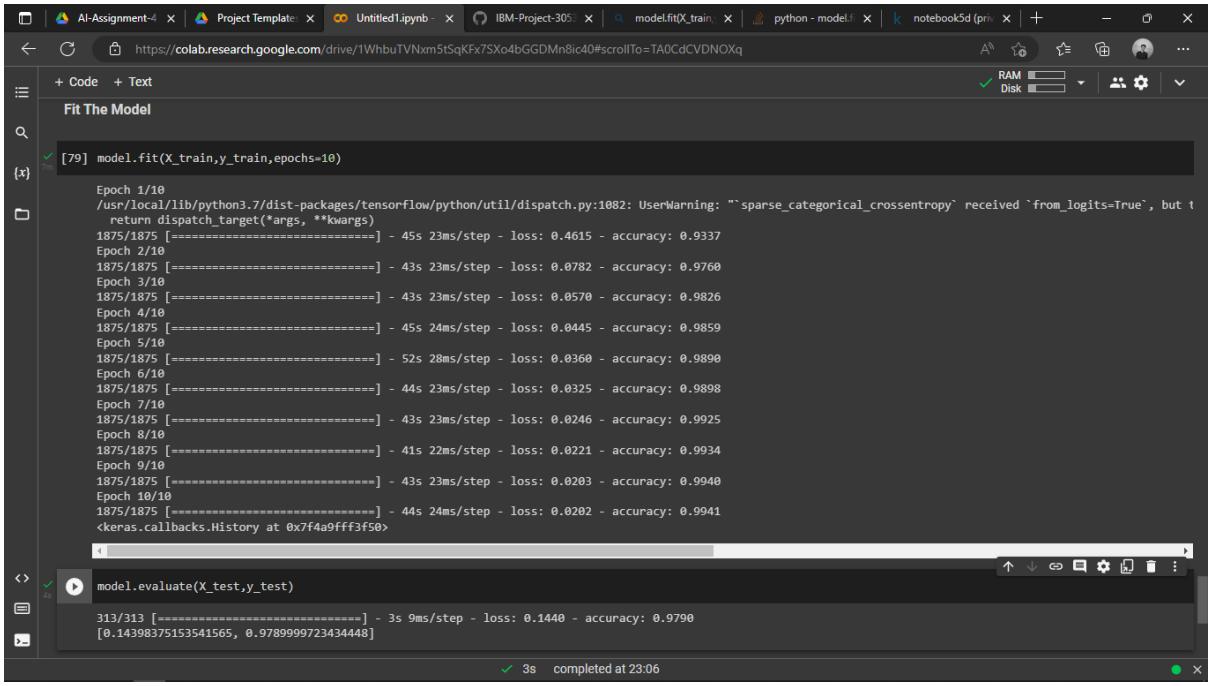
```
[69] model.add(Dense(10,activation='softmax'))
```

Compile The Model

```
[77] import tensorflow as tf
```

3s completed at 23:06


AI-Assignment-4
Project Template
Untitled1.ipynb
IBM-Project-305
model.fit(X_train,
python - model.f
notebook5d (priv
https://colab.research.google.com/drive/1WhbuTVNxm5tSqKFx7SXo4bGGDMn8ic40#scrollTo=TA0CdCVDNOXq
+ Code
+ Text
RAM
Disk
Compile The Model
[77] import tensorflow as tf
[78] model.compile(optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])
Fit The Model
[79] model.fit(X_train,y_train,epochs=10)
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarning: "`sparse_categorical_crossentropy` received `from_logits=True`, but t
return dispatch_target(*args, **kwargs)
1875/1875 [==========] - 45s 23ms/step - loss: 0.4615 - accuracy: 0.9337
Epoch 2/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0782 - accuracy: 0.9760
Epoch 3/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0570 - accuracy: 0.9826
Epoch 4/10
1875/1875 [==========] - 45s 24ms/step - loss: 0.0445 - accuracy: 0.9859
Epoch 5/10
1875/1875 [==========] - 52s 28ms/step - loss: 0.0360 - accuracy: 0.9890
Epoch 6/10
1875/1875 [==========] - 44s 23ms/step - loss: 0.0325 - accuracy: 0.9898
Epoch 7/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0246 - accuracy: 0.9925
Epoch 8/10
1875/1875 [==========] - 41s 22ms/step - loss: 0.0221 - accuracy: 0.9934
Epoch 9/10
3s
completed at 23:06


AI-Assignment-4
Project Template
Untitled1.ipynb
IBM-Project-305
model.fit(X_train,
python - model.f
notebook5d (priv
https://colab.research.google.com/drive/1WhbuTVNxm5tSqKFx7SXo4bGGDMn8ic40#scrollTo=TA0CdCVDNOXq
+ Code
+ Text
RAM
Disk
Fit The Model
[79] model.fit(X_train,y_train,epochs=10)
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarning: "`sparse_categorical_crossentropy` received `from_logits=True`, but t
return dispatch_target(*args, **kwargs)
1875/1875 [==========] - 45s 23ms/step - loss: 0.4615 - accuracy: 0.9337
Epoch 2/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0782 - accuracy: 0.9760
Epoch 3/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0570 - accuracy: 0.9826
Epoch 4/10
1875/1875 [==========] - 45s 24ms/step - loss: 0.0445 - accuracy: 0.9859
Epoch 5/10
1875/1875 [==========] - 52s 28ms/step - loss: 0.0360 - accuracy: 0.9890
Epoch 6/10
1875/1875 [==========] - 44s 23ms/step - loss: 0.0325 - accuracy: 0.9898
Epoch 7/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0246 - accuracy: 0.9925
Epoch 8/10
1875/1875 [==========] - 41s 22ms/step - loss: 0.0221 - accuracy: 0.9934
Epoch 9/10
1875/1875 [==========] - 43s 23ms/step - loss: 0.0203 - accuracy: 0.9940
Epoch 10/10
1875/1875 [==========] - 44s 24ms/step - loss: 0.0202 - accuracy: 0.9941
<keras.callbacks.History at 0x7f4a9fff3f50>
model.evaluate(X_test,y_test)
313/313 [==========] - 3s 9ms/step - loss: 0.1440 - accuracy: 0.9790
[0.14398375153541565, 0.9789999723434448]
3s
completed at 23:06