# Assignment –3

# Build CNN for Classification of Flowers

## Import required packages

```
[ ]  import keras
```

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

```
Mounted at /content/drive
```

```
[ ]  from keras.preprocessing.image import ImageDataGenerator
```

```
[ ]  train_datagen=ImageDataGenerator(rescale=1./255,
                                       shear_range=0.2,
                                       rotation_range=180,
                                       zoom_range=0.2,
                                       horizontal_flip=True)

     test_dataGen=ImageDataGenerator(rescale=1./255)
```

```
[ ]  #install Kaggle
     !pip install -q kaggle
```

```
[ ]  #create a kaggle folder
     !mkdir ~/.kaggle
```

```
[ ]  #copy the kaggle.json to folder created
     !cp kaggle.json ~/.kaggle/
```

```
cp: cannot stat 'kaggle.json': No such file or directory
```

```
[ ]  #permission for the json to act
     ! chmod 600 ~/.kaggle/kaggle.json
```

```
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
```

## 1.Download the dataset

```
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/train_set',
target_size = (128,128),
batch_size = 32,
class_mode = 'binary')
```

```
Found 2313 images belonging to 5 classes.
```

```
x_test = test_dataGen.flow_from_directory(r'/content/drive/MyDrive/IBM/test_set',
target_size = (128,128),
batch_size = 32,
class_mode = 'binary')
```

```
Found 2068 images belonging to 5 classes.
```

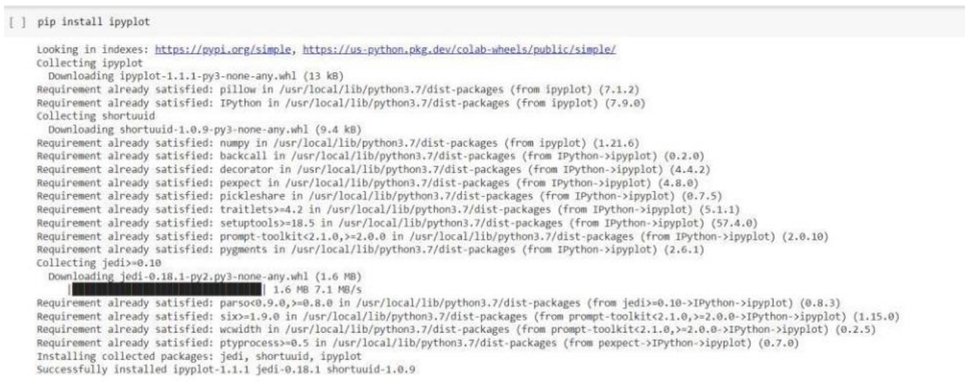
## 2. Image Augmentation

```
#give any random image path

img = image.load_img(r'/content/drive/MyDrive/IBM/test_set/daisy/10300722094_28fa978807_n.jpg')

x = image.img_to_array(img)

#expand the image shape

x = np.expand_dims(x,axis= 0)
```

```
img
```

```
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: scipy in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.7.3)
Requirement already satisfied: scikit-image>=0.14.2 in /usr/local/lib/python3.7/dist-packages (from imgaug) (0.18.3)
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) (1.8.4)
Requirement already satisfied: imageio in /usr/local/lib/python3.7/dist-packages (from imgaug) (2.9.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.15.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (from imgaug) (4.6.0.66)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.21.6)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from imgaug) (7.1.2)
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: 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: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (2.6.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (1.4.4)
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: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (0.11.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib->imgaug) (4.1.1)
```



```
[ ] pip install ipyplot

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Collecting ipyplot
      Downloading ipyplot-1.1.1-py3-none-any.whl (13 kB)
    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)
    Collecting shortuuid
      Downloading shortuuid-1.0.9-py3-none-any.whl (9.4 kB)
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from ipyplot) (1.21.6)
    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: pexpect in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (4.8.0)
    Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (0.7.5)
    Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (5.1.1)
    Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from IPython->ipyplot) (57.4.0)
    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)
    Collecting jedi>=0.10
      Downloading jedi-0.18.1-py2.py3-none-any.whl (1.6 MB)
        |████████████████████████████████| 1.6 MB 7.1 MB/s
    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: 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: 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: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-packages (from pexpect->IPython->ipyplot) (0.7.0)
    Installing collected packages: jedi, shortuuid, ipyplot
    Successfully installed ipyplot-1.1.1 jedi-0.18.1 shortuuid-1.0.9
```

## 3.Create Model

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

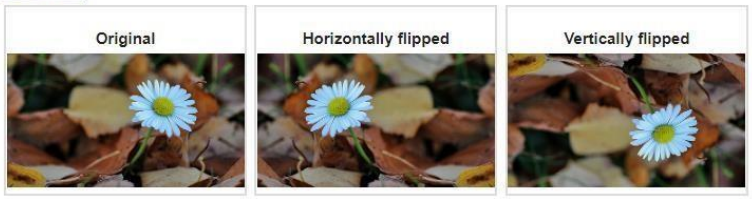
```
        WARNING! Google Colab Environment detected!
        You might encounter issues while running in Google Colab environment.
        If images are not displaying properly please try setting `force_b64` param to `True`.
```

```
[ ] input=imageio.imread("/content/drive/MyDrive/IBM/test_set/daisy/10559679065_50d2b16f6d.jpg")
```

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

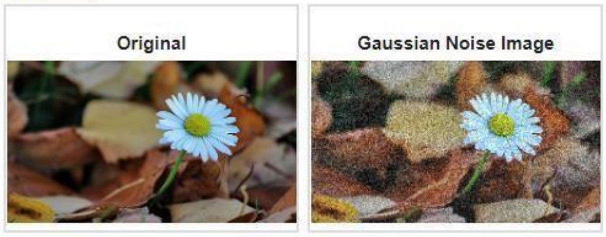
```
[ ] vflip=iaa.Flipud (p=1.0)
    input_vf=vflip.augment_image(input)
    images_list=[input, input_hf, input_vf]
    labels =['Original', 'Horizontally flipped', 'Vertically flipped']
    ipyplot.plot_images (images_list,labels=labels, img_width=180)
```

show html



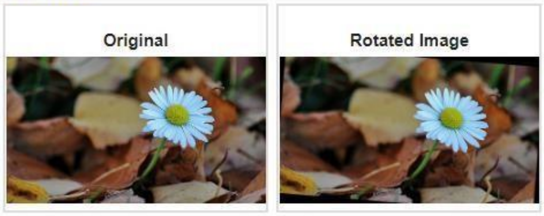
```
[ ] noise=iaa. AdditiveGaussianNoise (18,48)
    input_noise=noise.augment_image(input)
    images_list=[input, input_noise]
    labels= ["Original", "Gaussian Noise Image"]
    ipyplot.plot_images(images_list, labels=labels, img_width=188)
```

show html

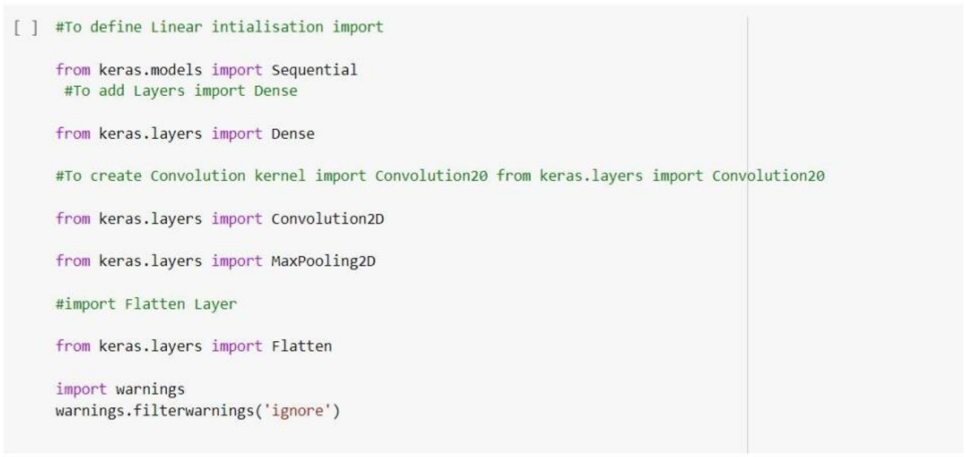


```
[ ] rot1 = iaa.Affine(rotate=(-30,30))

    input_rot1 = rot1.augment_image(input)
    images_list=[input, input_rot1]
    labels= ['Original', 'Rotated Image']
    ipyplot.plot_images(images_list,labels=labels, img_width=180)
```

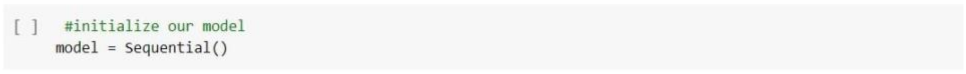
show html



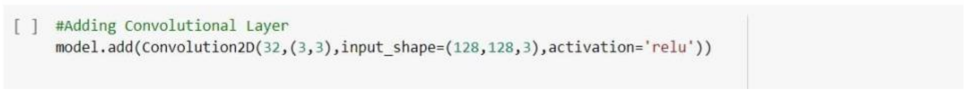
## 4.Add Layers(Convolution, MaxPooling, Flatten,Dense-(Hidden Layers),Output)



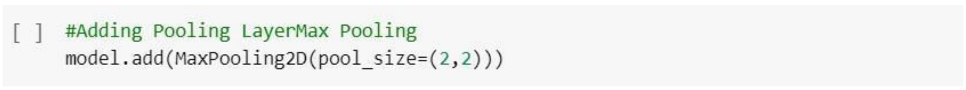
```
[ ]  #To define Linear intialisation import

     from keras.models import Sequential
      #To add Layers import Dense

     from keras.layers import Dense

     #To create Convolution kernel import Convolution20 from keras.layers import Convolution20

     from keras.layers import Convolution2D

     from keras.layers import MaxPooling2D

     #import Flatten Layer

     from keras.layers import Flatten

     import warnings
     warnings.filterwarnings('ignore')
```



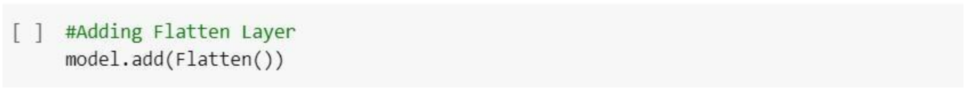
```
[ ]   #initialize our model
     model = Sequential()
```



```
[ ]  #Adding Convolutional Layer
     model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

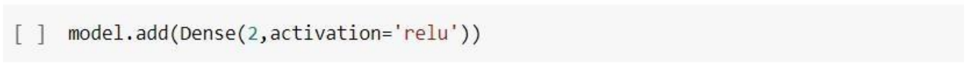


```
[ ]  #Adding Pooling LayerMax Pooling
     model.add(MaxPooling2D(pool_size=(2,2)))
```

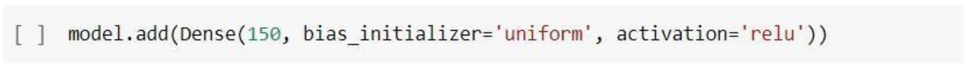


```
[ ]  #Adding Flatten Layer
     model.add(Flatten())
```

### Adding Hidden Layers

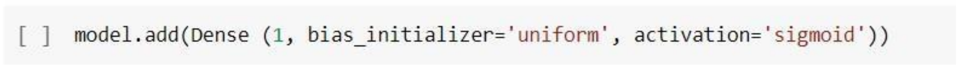


```
[ ]  model.add(Dense(2,activation='relu'))
```



```
[ ]  model.add(Dense(150, bias_initializer='uniform', activation='relu'))
```

### Adding Output Layer

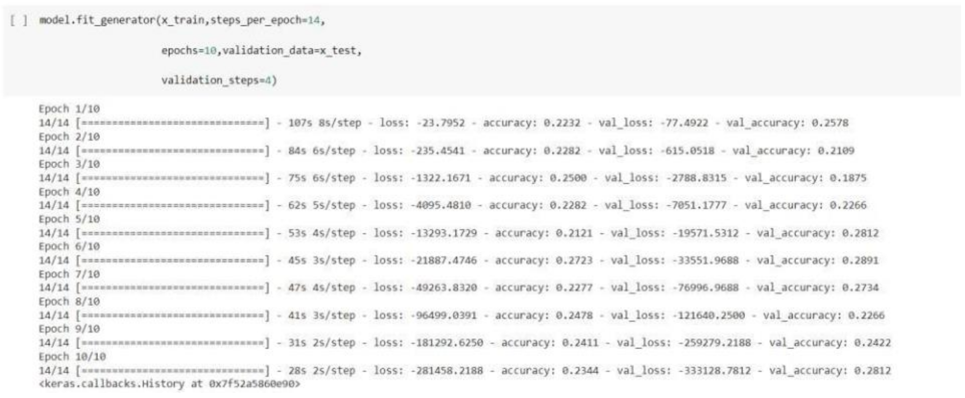


```
[ ]  model.add(Dense (1, bias_initializer='uniform', activation='sigmoid'))
```

## 5.Compile the Model

```
[ ]
    model.compile(loss = 'binary_crossentropy',

                  optimizer = "adam",

                  metrics = ["accuracy"])
```

**6.Fit the Model**



```
[ ] model.fit_generator(x_train,steps_per_epoch=14,

                   epochs=10,validation_data=x_test,

                   validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 107s 8s/step - loss: -23.7952 - accuracy: 0.2232 - val_loss: -77.4922 - val_accuracy: 0.2578
Epoch 2/10
14/14 [==============================] - 84s 6s/step - loss: -235.4541 - accuracy: 0.2282 - val_loss: -615.0518 - val_accuracy: 0.2109
Epoch 3/10
14/14 [==============================] - 75s 6s/step - loss: -1322.1671 - accuracy: 0.2500 - val_loss: -2788.8315 - val_accuracy: 0.1875
Epoch 4/10
14/14 [==============================] - 62s 5s/step - loss: -4095.4810 - accuracy: 0.2282 - val_loss: -7051.1777 - val_accuracy: 0.2266
Epoch 5/10
14/14 [==============================] - 53s 4s/step - loss: -13293.1729 - accuracy: 0.2121 - val_loss: -19571.5312 - val_accuracy: 0.2812
Epoch 6/10
14/14 [==============================] - 45s 3s/step - loss: -21887.4746 - accuracy: 0.2723 - val_loss: -33551.9688 - val_accuracy: 0.2891
Epoch 7/10
14/14 [==============================] - 47s 4s/step - loss: -49263.8320 - accuracy: 0.2277 - val_loss: -76996.9688 - val_accuracy: 0.2734
Epoch 8/10
14/14 [==============================] - 41s 3s/step - loss: -96499.0391 - accuracy: 0.2478 - val_loss: -121640.2500 - val_accuracy: 0.2266
Epoch 9/10
14/14 [==============================] - 31s 2s/step - loss: -181292.6250 - accuracy: 0.2411 - val_loss: -259279.2188 - val_accuracy: 0.2422
Epoch 10/10
14/14 [==============================] - 28s 2s/step - loss: -281458.2188 - accuracy: 0.2344 - val_loss: -333128.7812 - val_accuracy: 0.2812
<keras.callbacks.History at 0x7f52a5860e90>
```

**7.Save the Model & Test the Model**

```
[ ] model.save("flowers.h5")
```

```
[ ] ls

    drive/  flowers.h5  sample_data/
```

```
[ ] from keras.models import load_model

    #import image class from keros

    from keras.preprocessing import image

    #import numpy

    import numpy as np

    #import cv2

    import cv2
```

```
[ ] #Load the saved model

    model = load_model("flowers.h5")
```