Importing the packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
import cv2 as cv
```

```
from google.colab import drive
drive.mount("/content/drive", force_remount=True)
```

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

```
train_dir="/content/drive/MyDrive/DISEASEDATASET/train/flip"
test_dir="/content/drive/MyDrive/DISEASEDATASET/test/flip"
```

Importing the dataset directory

```
class_names=['BUMPS','HAIR LOSS','HOT SPOTS','RASHES','SORES']
```

Generating the Dataset to input it as the model

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen= ImageDataGenerator()
test_datagen= ImageDataGenerator()
train_gen=train_datagen.flow_from_directory(train_dir,target_size=(128,128),batch_size=3)
val_gen=test_datagen.flow_from_directory(test_dir,target_size=(128,128),batch_size=1)
```

```
    Found 469 images belonging to 5 classes.
    Found 101 images belonging to 5 classes.
```

Importing the packages for creating the cnn model

```
from tensorflow.keras import layers
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Input
```

```
inputs = Input(shape=[128,128,3])

convo1=Conv2D(128,kernel_size=(3,3), padding="same",activation='relu',use_bias=True)(inputs)
convo1_pooling=MaxPooling2D((2, 2))(convo1)

convo2=(Conv2D(64, (3, 3), activation='relu',use_bias=True))(convo1_pooling)
convo2_pooling=(MaxPooling2D((2, 2)))(convo2)

convo3=(Conv2D(32, (3, 3), activation='relu',use_bias=True))(convo2_pooling)
convo3_pooling=(MaxPooling2D((2, 2)))(convo3)

convo4=(Conv2D(128, (3, 3), activation='relu',use_bias=True, kernel_regularizer =tf.keras.regularizers.l2( l=0.001)))(convo3_poo
convo4_pooling=(MaxPooling2D((2, 2)))(convo4)

flatten=layers.Flatten()(convo4_pooling)

hidden1=layers.Dense(256,activation='relu')(flatten)
hidden2=layers.Dense(128,activation='relu')(hidden1)
hidden3=layers.Dense(64,activation='relu')(hidden2)
concat_1=keras.layers.Concatenate()([hidden1,hidden3])
hidden4=layers.Dense(32,activation='relu')(concat_1)
hidden5=layers.Dense(64,activation='relu')(hidden4)
concat_2=keras.layers.Concatenate()([hidden4,hidden5])
hidden6=layers.Dense(128,activation='relu')(concat_2)
concat_3=keras.layers.Concatenate()([hidden2,hidden6])
hidden7=layers.Dense(64,activation='relu')(concat_3)
hidden8=layers.Dense(32,activation='relu')(hidden7)
output=layers.Dense(5,activation='softmax')(hidden8)
model=keras.models.Model(inputs=[inputs],outputs=[output])
```

```
model.summary()
```

```
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_2 (InputLayer)        [(None, 128, 128, 3)]        0         []

 conv2d_4 (Conv2D)           (None, 128, 128, 128)        3584      ['input_2[0][0]']

 max_pooling2d_4 (MaxPoolin  (None, 64, 64, 128)          0         ['conv2d_4[0][0]']
 g2D)

 conv2d_5 (Conv2D)           (None, 62, 62, 64)           73792     ['max_pooling2d_4[0][0]']

 max_pooling2d_5 (MaxPoolin  (None, 31, 31, 64)           0         ['conv2d_5[0][0]']
 g2D)

 conv2d_6 (Conv2D)           (None, 29, 29, 32)           18464     ['max_pooling2d_5[0][0]']

 max_pooling2d_6 (MaxPoolin  (None, 14, 14, 32)           0         ['conv2d_6[0][0]']
 g2D)

 conv2d_7 (Conv2D)           (None, 12, 12, 128)          36992     ['max_pooling2d_6[0][0]']

 max_pooling2d_7 (MaxPoolin  (None, 6, 6, 128)            0         ['conv2d_7[0][0]']
 g2D)

 flatten_1 (Flatten)         (None, 4608)                 0         ['max_pooling2d_7[0][0]']

 dense_9 (Dense)             (None, 256)                  1179904   ['flatten_1[0][0]']

 dense_10 (Dense)            (None, 128)                  32896     ['dense_9[0][0]']

 dense_11 (Dense)            (None, 64)                   8256      ['dense_10[0][0]']

 concatenate_3 (Concatenate  (None, 320)                  0         ['dense_9[0][0]',
 )                                                                   'dense_11[0][0]']

 dense_12 (Dense)            (None, 32)                   10272     ['concatenate_3[0][0]']

 dense_13 (Dense)            (None, 64)                   2112      ['dense_12[0][0]']

 concatenate_4 (Concatenate  (None, 96)                   0         ['dense_12[0][0]',
 )                                                                   'dense_13[0][0]']

 dense_14 (Dense)            (None, 128)                  12416     ['concatenate_4[0][0]']

 concatenate_5 (Concatenate  (None, 256)                  0         ['dense_10[0][0]',
 )                                                                   'dense_14[0][0]']

 dense_15 (Dense)            (None, 64)                   16448     ['concatenate_5[0][0]']

 dense_16 (Dense)            (None, 32)                   2080      ['dense_15[0][0]']

 dense_17 (Dense)            (None, 5)                    165       ['dense_16[0][0]']

==================================================================================================
Total params: 1397381 (5.33 MB)
Trainable params: 1397381 (5.33 MB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```

Compiling the model

```
from tensorflow.keras import optimizers
#from keras.optimizers import Adam
model.compile(optimizer=keras.optimizers.RMSprop(learning_rate=0.0001),
             loss='categorical_crossentropy',
             metrics=['acc'])
```

```
checkpoint_cb= keras.callbacks.ModelCheckpoint("Bestmodel.h5",save_best_only=True)
```

```
early_stop=keras.callbacks.EarlyStopping(patience=10,restore_best_weights=True)
```

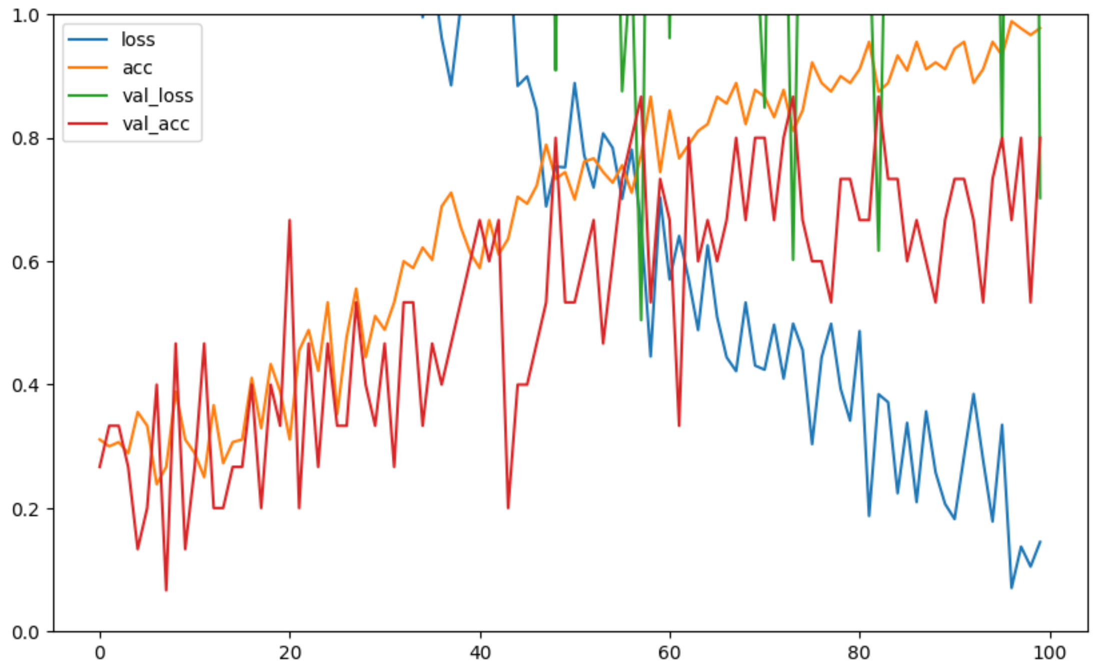
```
model_history=model.fit(train_gen,steps_per_epoch=30,epochs=100,validation_data=val_gen,validation_steps=15,callbacks=[checkpoir
```

```
Epoch 1/100
30/30 [==============================] - 11s 279ms/step - loss: 2.2252 - acc: 0.3111 - val_loss: 2.0243 - val_acc: 0.2667
Epoch 2/100
30/30 [==============================] - 9s 307ms/step - loss: 1.6460 - acc: 0.3000 - val_loss: 1.9330 - val_acc: 0.3333
```

```
Epoch 3/100
30/30 [==============================] - 11s 377ms/step - loss: 1.7019 - acc: 0.3068 - val_loss: 1.6378 - val_acc: 0.3333
Epoch 4/100
30/30 [==============================] - 8s 266ms/step - loss: 1.7080 - acc: 0.2889 - val_loss: 2.0290 - val_acc: 0.2667
Epoch 5/100
30/30 [==============================] - 9s 310ms/step - loss: 1.6611 - acc: 0.3556 - val_loss: 1.7011 - val_acc: 0.1333
Epoch 6/100
30/30 [==============================] - 8s 281ms/step - loss: 1.5821 - acc: 0.3333 - val_loss: 1.9043 - val_acc: 0.2000
Epoch 7/100
30/30 [==============================] - 8s 260ms/step - loss: 1.6468 - acc: 0.2386 - val_loss: 1.5905 - val_acc: 0.4000
Epoch 8/100
30/30 [==============================] - 8s 268ms/step - loss: 1.6677 - acc: 0.2667 - val_loss: 2.1418 - val_acc: 0.0667
Epoch 9/100
30/30 [==============================] - 10s 321ms/step - loss: 1.6501 - acc: 0.3889 - val_loss: 1.4328 - val_acc: 0.4667
Epoch 10/100
30/30 [==============================] - 8s 261ms/step - loss: 1.5968 - acc: 0.3111 - val_loss: 1.8469 - val_acc: 0.1333
Epoch 11/100
30/30 [==============================] - 8s 252ms/step - loss: 1.6034 - acc: 0.2889 - val_loss: 1.8406 - val_acc: 0.2667
Epoch 12/100
30/30 [==============================] - 9s 312ms/step - loss: 1.6317 - acc: 0.2500 - val_loss: 1.7040 - val_acc: 0.4667
Epoch 13/100
30/30 [==============================] - 8s 253ms/step - loss: 1.6182 - acc: 0.3667 - val_loss: 1.8708 - val_acc: 0.2000
Epoch 14/100
30/30 [==============================] - 8s 260ms/step - loss: 1.6094 - acc: 0.2727 - val_loss: 1.5497 - val_acc: 0.2000
Epoch 15/100
30/30 [==============================] - 9s 289ms/step - loss: 1.5227 - acc: 0.3068 - val_loss: 1.8908 - val_acc: 0.2667
Epoch 16/100
30/30 [==============================] - 11s 362ms/step - loss: 1.5711 - acc: 0.3111 - val_loss: 2.0427 - val_acc: 0.2667
Epoch 17/100
30/30 [==============================] - 8s 250ms/step - loss: 1.5235 - acc: 0.4111 - val_loss: 1.4892 - val_acc: 0.4000
Epoch 18/100
30/30 [==============================] - 7s 249ms/step - loss: 1.5471 - acc: 0.3295 - val_loss: 1.7261 - val_acc: 0.2000
Epoch 19/100
30/30 [==============================] - 9s 309ms/step - loss: 1.5197 - acc: 0.4333 - val_loss: 1.4547 - val_acc: 0.4000
Epoch 20/100
30/30 [==============================] - 8s 269ms/step - loss: 1.4743 - acc: 0.3889 - val_loss: 1.7203 - val_acc: 0.3333
Epoch 21/100
30/30 [==============================] - 8s 266ms/step - loss: 1.6138 - acc: 0.3111 - val_loss: 1.2374 - val_acc: 0.6667
Epoch 22/100
30/30 [==============================] - 9s 316ms/step - loss: 1.4097 - acc: 0.4556 - val_loss: 1.9078 - val_acc: 0.2000
Epoch 23/100
30/30 [==============================] - 9s 304ms/step - loss: 1.4155 - acc: 0.4886 - val_loss: 1.4311 - val_acc: 0.4667
Epoch 24/100
30/30 [==============================] - 8s 265ms/step - loss: 1.5081 - acc: 0.4222 - val_loss: 1.8174 - val_acc: 0.2667
Epoch 25/100
30/30 [==============================] - 9s 305ms/step - loss: 1.3997 - acc: 0.5333 - val_loss: 1.8016 - val_acc: 0.4667
Epoch 26/100
30/30 [==============================] - 9s 300ms/step - loss: 1.4961 - acc: 0.3523 - val_loss: 1.7191 - val_acc: 0.3333
Epoch 27/100
30/30 [==============================] - 8s 265ms/step - loss: 1.3585 - acc: 0.4778 - val_loss: 2.4643 - val_acc: 0.3333
Epoch 28/100
30/30 [==============================] - 8s 270ms/step - loss: 1.2362 - acc: 0.5556 - val_loss: 1.5166 - val_acc: 0.5333
Epoch 29/100
30/30 [==============================] - 11s 376ms/step - loss: 1.3031 - acc: 0.4444 - val_loss: 1.3870 - val_acc: 0.4000
```

```
pd.DataFrame(model_history.history).plot(figsize=[10,6])
plt.gca().set_ylim(0,1)
```

```
(0.0, 1.0)
```



Saving the model

```
#download_path = "/Home/Yuvanika/Downloads/AI"
```

```
model.save('/content/drive/MyDrive/widen.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5
  saving_api.save_model(
```

Testing for an output

```
image = cv.imread('/content/drive/MyDrive/DISEASEDATASET/train/flip/HAIR LOSS/img9.jpg')
```

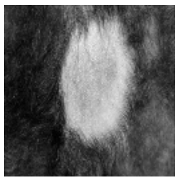
```
image.shape
```

```
(128, 128, 3)
```

```
from google.colab.patches import cv2_imshow
```

Resizing the user given input

```
cv2_imshow(image)
```



```
def crop_resize(img):
  image_1_resize = cv.resize(img,(128,128))
  return image_1_resize
```

```
image=crop_resize(image)
image.shape
```

```
(128, 128, 3)
```

```
image = tf.expand_dims(image, axis=0)
image.shape.as_list()
```

```
[1, 128, 128, 3]
```

```
yproba = model.predict(image)
yproba
```

```
1/1 [==============================] - 0s 173ms/step
array([[7.3198685e-06, 9.9983275e-01, 1.5986059e-04, 1.6718043e-07,
        2.8719316e-09]], dtype=float32)
```

```
class_name=class_names[np.argmax(yproba,axis=1)[0]]
```