

In [14]:

```
import os
import numpy as np
import pandas as pd
```

In [15]:

```
my_dir=r"C:\Users\Arun Teja\OneDrive\Desktop\Datasets"
```

In [30]:

```
my_dir
```

Out[30]:

```
'C:\\Users\\Arun Teja\\OneDrive\\Desktop\\Datasets'
```

In [4]:

```
train_dir=os.path.join(my_dir, 'train_set')
test_dir=os.path.join(my_dir, 'test_set')
```

In [31]:

```
train_dir
```

Out[31]:

```
'C:\\Users\\Arun Teja\\OneDrive\\Desktop\\Datasets\\train_set'
```

In [32]:

```
test_dir
```

Out[32]:

```
'C:\\Users\\Arun Teja\\OneDrive\\Desktop\\Datasets\\test_set'
```

In [33]:

```
train_dir_males=os.path.join(train_dir, 'males')
train_dir_females=os.path.join(train_dir, 'females')
test_dir_males=os.path.join(train_dir, 'males')
test_dir_females=os.path.join(train_dir, 'females')
```

In [34]:

```
print(train_dir_males)
print(train_dir_females)
print(test_dir_males)
print(test_dir_females)
```

```
C:\Users\Arun Teja\OneDrive\Desktop\Datasets\train_set\males
C:\Users\Arun Teja\OneDrive\Desktop\Datasets\train_set\females
C:\Users\Arun Teja\OneDrive\Desktop\Datasets\train_set\males
C:\Users\Arun Teja\OneDrive\Desktop\Datasets\train_set\females
```

In [5]:

```
import keras
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPool2D
from keras.layers import Flatten
from keras.layers import Dense
```

In [21]:

```
cnn_classifier=Sequential()
#step1:Convolution
```

In [22]:

```
cnn_classifier.add(Conv2D(32,3,3,input_shape = (128,128,3),activation='relu'))
#step2: pooling
cnn_classifier.add(MaxPool2D(pool_size=(2,2)))

cnn_classifier.add(Conv2D(16,3,3,activation='relu'))
cnn_classifier.add(MaxPool2D(pool_size=(2,2)))
#step3:flattering
cnn_classifier.add(Flatten())

#step4: Full Connection
cnn_classifier.add(Dense(units=128,activation='relu'))

cnn_classifier.add(Dense(units=1,activation='sigmoid'))
```

In [23]:

```
cnn_classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
#Compiling the CNN
```

In [24]:

```
from tensorflow.python import training
from keras.preprocessing.image import ImageDataGenerator
train_data = ImageDataGenerator(rescale = 1./255,
                                shear_range=0.2,
                                zoom_range=0.2,
                                horizontal_flip=True)
test_data = ImageDataGenerator(rescale=1./255)

training_set = train_data.flow_from_directory(r'C:\Users\Arun Teja\OneDrive\Desktop\Dataset',
                                              target_size = (128,128),
                                              batch_size = 32,
                                              class_mode='binary')
test_set = test_data.flow_from_directory(r'C:\Users\Arun Teja\OneDrive\Desktop\Datasets\Val',
                                         target_size = (128,128),
                                         batch_size = 32,
                                         class_mode='binary')
```

Found 45539 images belonging to 2 classes.

Found 11649 images belonging to 2 classes.

In [41]:

```
cnn_classifier.fit_generator(training_set, steps_per_epoch=100,
                             epochs=20,
                             validation_data=test_set,
                             validation_steps=100)
```

C:\Users\Arun Teja\AppData\Local\Temp\ipykernel_19276\2030685884.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
cnn_classifier.fit_generator(training_set, steps_per_epoch=100,
```

Epoch 1/20

```
100/100 [=====] - 25s 243ms/step - loss: 0.6846 - accuracy: 0.5472 - val_loss: 0.6482 - val_accuracy: 0.5981
```

Epoch 2/20

```
100/100 [=====] - 24s 237ms/step - loss: 0.5895 - accuracy: 0.6966 - val_loss: 0.5762 - val_accuracy: 0.6506
```

Epoch 3/20

```
100/100 [=====] - 21s 214ms/step - loss: 0.4735 - accuracy: 0.7716 - val_loss: 0.4455 - val_accuracy: 0.7922
```

Epoch 4/20

```
100/100 [=====] - 21s 208ms/step - loss: 0.4109 - accuracy: 0.8272 - val_loss: 0.3575 - val_accuracy: 0.8562
```

Epoch 5/20

```
100/100 [=====] - 20s 204ms/step - loss: 0.3903 - accuracy: 0.8303 - val_loss: 0.3322 - val_accuracy: 0.8587
```

Epoch 6/20

```
100/100 [=====] - 19s 194ms/step - loss: 0.3715 - accuracy: 0.8431 - val_loss: 0.3285 - val_accuracy: 0.8684
```

Epoch 7/20

```
100/100 [=====] - 19s 187ms/step - loss: 0.3328 - accuracy: 0.8666 - val_loss: 0.3553 - val_accuracy: 0.8522
```

Epoch 8/20

```
100/100 [=====] - 18s 179ms/step - loss: 0.3364 - accuracy: 0.8616 - val_loss: 0.3133 - val_accuracy: 0.8672
```

Epoch 9/20

```
100/100 [=====] - 18s 175ms/step - loss: 0.3443 - accuracy: 0.8587 - val_loss: 0.3140 - val_accuracy: 0.8756
```

Epoch 10/20

```
100/100 [=====] - 17s 174ms/step - loss: 0.3344 - accuracy: 0.8637 - val_loss: 0.2885 - val_accuracy: 0.8903
```

Epoch 11/20

```
100/100 [=====] - 17s 174ms/step - loss: 0.3243 - accuracy: 0.8612 - val_loss: 0.2746 - val_accuracy: 0.8875
```

Epoch 12/20

```
100/100 [=====] - 17s 166ms/step - loss: 0.2799 - accuracy: 0.8888 - val_loss: 0.2643 - val_accuracy: 0.8944
```

Epoch 13/20

```
100/100 [=====] - 17s 165ms/step - loss: 0.2861 - accuracy: 0.8847 - val_loss: 0.2639 - val_accuracy: 0.8994
```

Epoch 14/20

```
100/100 [=====] - 16s 161ms/step - loss: 0.3033 - accuracy: 0.8741 - val_loss: 0.2566 - val_accuracy: 0.9016
```

Epoch 15/20

```
100/100 [=====] - 16s 159ms/step - loss: 0.2913 - accuracy: 0.8844 - val_loss: 0.2489 - val_accuracy: 0.8972
```

Epoch 16/20

```
100/100 [=====] - 15s 155ms/step - loss: 0.2839 - accuracy: 0.8856 - val_loss: 0.2521 - val_accuracy: 0.9025
```

Epoch 17/20

```
100/100 [=====] - 15s 154ms/step - loss: 0.2756 - a
ccuracy: 0.8922 - val_loss: 0.2616 - val_accuracy: 0.8928
Epoch 18/20
100/100 [=====] - 15s 152ms/step - loss: 0.2908 - a
ccuracy: 0.8816 - val_loss: 0.2314 - val_accuracy: 0.9081
Epoch 19/20
100/100 [=====] - 15s 152ms/step - loss: 0.2719 - a
ccuracy: 0.8881 - val_loss: 0.2473 - val_accuracy: 0.9016
Epoch 20/20
100/100 [=====] - 15s 152ms/step - loss: 0.2530 - a
ccuracy: 0.8997 - val_loss: 0.3177 - val_accuracy: 0.8741
```

Out[41]:

<keras.callbacks.History at 0x260455569d0>

In [25]:

```
validation_male = []
folder_dir = r"C:\Users\Arun Teja\OneDrive\Desktop\Datasets\Validation\male"
for image in os.listdir(folder_dir) :
    if image is None:
        print('Wrong path:', folder_dir)
    else:
#         img = cv2.resize(cv2.imread(folder_dir + "/" + image), dsize=(128,128))
        test_image=keras.utils.load_img(folder_dir + "/" + image,target_size=(128,128))
        test_image=keras.utils.img_to_array(test_image)
        test_image=np.expand_dims(test_image,axis=0)
        validation_male.append(test_image)
```

In [26]:

```
validation_female = []
folder_dir = r"C:\Users\Arun Teja\OneDrive\Desktop\Datasets\Validation\female"
for image in os.listdir(folder_dir) :
    if image is None:
        print('Wrong path:', folder_dir)
    else:
#         img = cv2.resize(cv2.imread(folder_dir + "/" + image), dsize=(128,128))
        test_image=keras.utils.load_img(folder_dir + "/" + image,target_size=(128,128))
        test_image=keras.utils.img_to_array(test_image)
        test_image=np.expand_dims(test_image,axis=0)
        validation_male.append(test_image)
```

In [27]:

```

test_female = []
folder_dir = r"C:\Users\Arun Teja\OneDrive\Desktop\Datasets\test_dataset\Female Faces"
for image in os.listdir(folder_dir) :
    if image is None:
        print('Wrong path:', folder_dir)
    else:
#         img = cv2.resize(cv2.imread(folder_dir + "/" + image), dsize=(128,128))
        test_image=keras.utils.load_img(folder_dir + "/" + image,target_size=(128,128))
        test_image=keras.utils.img_to_array(test_image)
        test_image=np.expand_dims(test_image,axis=0)
        test_female.append(test_image)

```

C:\Users\Arun Teja\anaconda3\lib\site-packages\PIL\Image.py:945: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
 warnings.warn(

In [28]:

```

test_male = []
folder_dir = r"C:\Users\Arun Teja\OneDrive\Desktop\Datasets\test_dataset\Male Faces"
for image in os.listdir(folder_dir) :
    if image is None:
        print('Wrong path:', folder_dir)
    else:
#         img = cv2.resize(cv2.imread(folder_dir + "/" + image), dsize=(128,128))
        test_image=keras.utils.load_img(folder_dir + "/" + image,target_size=(128,128))
        test_image=keras.utils.img_to_array(test_image)
        test_image=np.expand_dims(test_image,axis=0)
        test_male.append(test_image)

```

In [35]:

```

validation_male = np.array(validation_male)
validation_male.shape

```

Out[35]:

```
(11649, 1, 128, 128, 3)
```

In [36]:

```

validation_female = np.array(validation_female)
validation_female.shape

```

Out[36]:

```
(0,)
```

In [30]:

```

test_male = np.array(test_male)
test_female = np.array(test_female)

```

In [43]:

```

#prediction
import numpy as np
import keras

result_1 = []
for i in range(0,5808):
    single_result=cnn_classifier.predict(validation_male[i])
    result.append(single_result)

training_set.class_indices
if result [0][0]==1:
    prediction='male'
else:
    prediction='female'
print(prediction)

print(test_image.shape)

```

```

1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 25ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 25ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 19ms/step

```

In [47]:

```

#Prediction
import numpy as np
import keras

result_2 = []
for i in range(0,0):
    single_result=cnn_classifier.predict(validation_female[i]) #prediction of images
    result_2.append(single_result)

```

In [50]:

```
count_male = 0
for i in range(0,0):
    if(np.argmax(result_1[i])==1):
        count_male += 1
print("Accuracy on validation data of male : ",(count_male/5808)*100)
```

Accuracy on validation data of male : 0.0

In [52]:

```
count_female = 0
for i in range(0,0):
    if(np.argmax(result_2[i])==0):
        count_female += 1
print("Accuracy on validation data of female : ",(count_female/5808)*100)
```

Accuracy on validation data of female : 0.0

In [53]:

```
#Prediction

result_3 = []
for i in range(0,2720):
    single_result=cnn_classifier.predict(test_male[i]) #prediction of images
    result_3.append(single_result)

result_4 = []
for i in range(0,2698):
    single_result=cnn_classifier.predict(test_female[i]) #prediction of images
    result_4.append(single_result)
```

```
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 32ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 34ms/step
```


In [54]:

```
count_male = 0
for i in range(0,2720):
    if(np.argmax(result_3[i])==1):
        count_male += 1
print("Accuracy on test data of male : ",(count_male/2720)*100)
```

Accuracy on test data of male : 0.0

In [55]:

```
count_female = 0
for i in range(0,2698):
    if(np.argmax(result_4[i])==0):
        count_female += 1
print("Accuracy on test data of female : ",(count_female/2720)*100)
```

Accuracy on test data of female : 99.19117647058823

In [56]:

```
def show_image(image, title=None):
    plt.imshow(image)
    plt.title(title)
    plt.axis('off')
```

In [58]:

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

test_data = ImageDataGenerator(rescale = 1./255)

test_set = test_data.flow_from_directory(r'C:\Users\Arun Teja\OneDrive\Desktop\Datasets\tes
                                         target_size = (128, 128),
                                         batch_size = 32,
                                         class_mode = 'binary')
```

Found 5418 images belonging to 2 classes.

In [59]:

```

import matplotlib.pyplot as plt

class_names = ['female', 'male']
i=1
plt.figure(figsize=(20,8))
for images, labels in test_set:
    index = np.random.randint(len(images))
    image, label = images[index], class_names[int(labels[index])]
    pred_label = class_names[np.argmax(cnn_classifier.predict(image[np.newaxis, ...]))]

    plt.subplot(10,5, i)
    show_image(image, title=f"Org : {label}, Pred : {pred_label}")
    i+=1
    if i>=51: break

plt.tight_layout()
plt.show()



















































```

```

1/1 [=====] - 0s 51ms/step
1/1 [=====] - 0s 46ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 55ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 36ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 36ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 36ms/step
1/1 [=====] - 0s 49ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 44ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 44ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 25ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 21ms/step

```

```
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 45ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 31ms/step
```

Org : male, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 
Org : male, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 
Org : male, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 
Org : female, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 
Org : male, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 
Org : female, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 
Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 
Org : female, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 
Org : female, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 	Org : male, Pred : female 	Org : female, Pred : female 
Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 	Org : female, Pred : female 

In []: