```
import numpy as np
import pandas as pd
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import tensorflow as tf
import matplotlib.image as mpimg
from tensorflow import keras
from tensorflow.keras import layers
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense, Bat
import keras
from keras.preprocessing.image import ImageDataGenerator

from zipfile import ZipFile

with ZipFile("/content/drive/MyDrive/archive.zip" , 'r') as ob:
  ob.extractall("/content/dataset")

train_dir = r"/content/dataset/training_set/training_set"
test_dir  = r"/content/dataset/test_set/test_set"

img_width=128
img_height=128
img_size=(128,128)
img_channels=3

train_data= ImageDataGenerator(rotation_range=15,rescale=1/255,sh
                               zoom_range=0.2,horizontal_flip=Tru
                               width_shift_range=0.1,
                                height_shift_range=0.1)
test_data=  ImageDataGenerator(rotation_range=15,rescale=1/255,sh
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
al_flip=True
                  width_shift_range=0.1,height_shift_
```

```
train_set=      train_data.flow_from_directory(directory=train_di
validation_set= test_data.flow_from_directory(directory=test_dir,
```

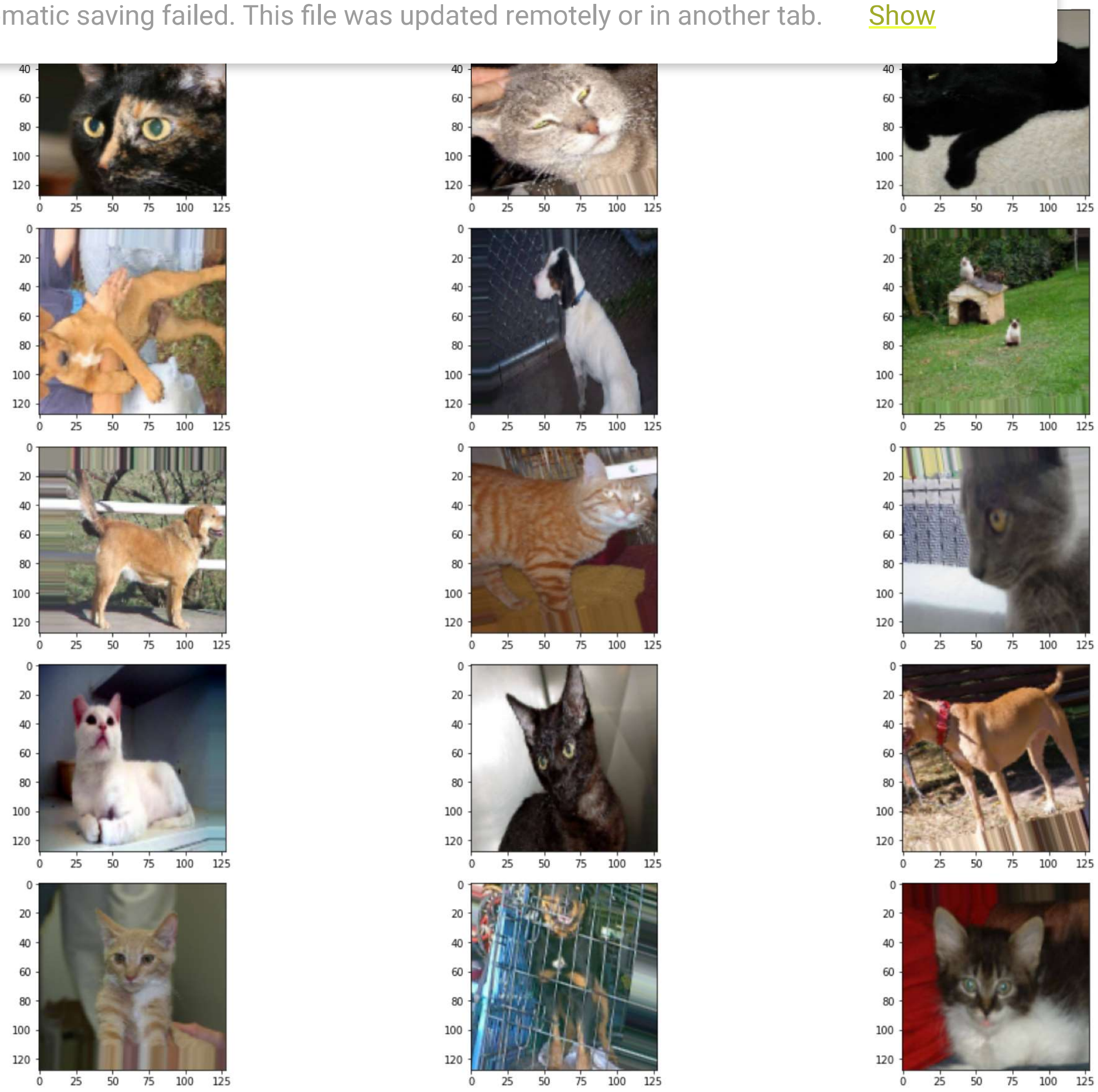
```
Found 8005 images belonging to 2 classes.
Found 2023 images belonging to 2 classes.
```

```
validation_set.class_indices
```

```
{'cats': 0, 'dogs': 1}
```

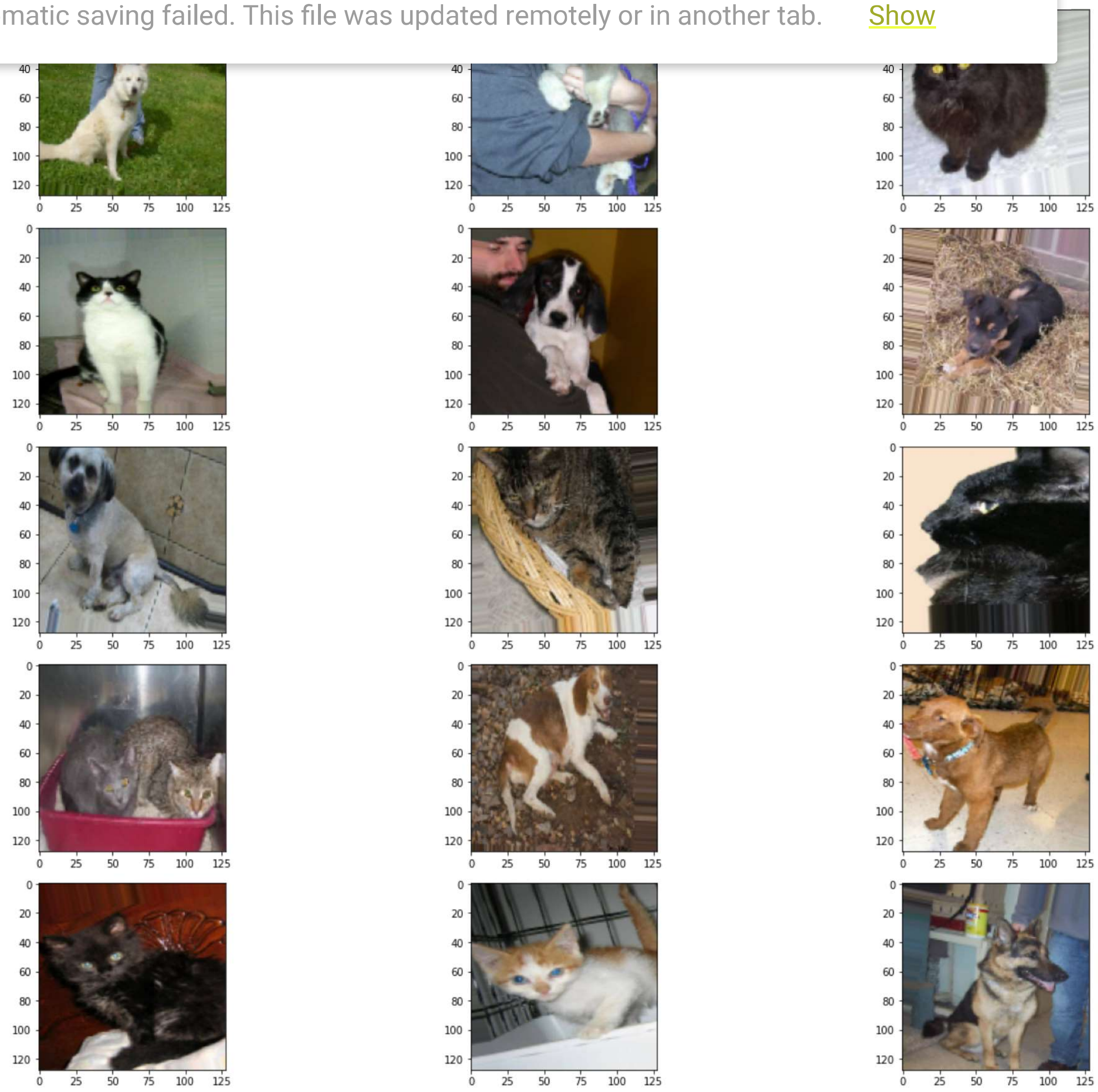
```
plt.figure(figsize=(20,15))
for i in range(0,15):
  plt.subplot(5, 3, i+1 )
  for img in next(train_set):
    image= img[0]
    plt.imshow(image)
    break
plt.tight_layout()
plt.show()
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff



```
plt.figure(figsize=(20,15))
for i in range(0,15):
  plt.subplot(5, 3, i+1 )
  for img in next(train_set):
    image= img[1]
    plt.imshow(image)
    break
plt.tight_layout()
plt.show()
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff



Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
|model=Sequential()

model.add(Conv2D(64,(3,3),activation='relu',input_shape=(128,128,
model.add(BatchNormalization())
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.25))

model.add(Conv2D(64,(3,3),activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.25))

model.add(Conv2D(128,(3,3),activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(2,activation='softmax'))
```

```
  File "<ipython-input-78-340f05c6caef>", line 1
    |model=Sequential()
    ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
model.summary()
```

```
model.compile(optimizer="rmsprop",loss="categorical_crossentropy"
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
ResultMap ={}
for faceValue,faceName in zip(TrainClasses.values(),TrainClasses.
  ResultMap[faceValue]=faceName


#Saving the face map for future reference
import pickle
with open("ResultMap.pkl", 'wb') as filesWriteStream:
    pickle.dump(ResultMap , filesWriteStream)

#The model will give as a numeric tag
#This mapping will help to get the corresponding face name for it
print("Mapping of Face and its ID",ResultMap)

#The number of neurons for the output layer is equal to the numbe
OutputNeurons=len(ResultMap)
print('\n The Number of OUtput Neurons:',OutputNeurons)


final_train_set = 8005
final_validation_set = 2023
batch_size = 15
no_epochs  = 20


model.fit_generator(train_set, epochs=no_epochs,validation_data =
                    steps_per_epoch= final_train_set//batch_size,
```

```
Epoch 1/20
533/533 [==============================] - 60s 112ms/step - loss: 0.8387 - accuracy: 0.
Epoch 2/20
533/533 [==============================] - 59s 111ms/step - loss: 0.6632 - accuracy: 0.
Epoch 3/20
533/533 [==============================] - 59s 111ms/step - loss: 0.6341 - accuracy: 0.
Epoch 4/20
533/533 [==============================] - 60s 113ms/step - loss: 0.6273 - accuracy: 0.
Epoch 5/20
533/533 [==============================] - 59s 111ms/step - loss: 0.6093 - accuracy: 0.
Epoch 6/20
533/533 [==============================] - 59s 111ms/step - loss: 0.5938 - accuracy: 0.
Epoch 7/20
533/533 [==============================] - 59s 110ms/step - loss: 0.5651 - accuracy: 0.
Epoch 8/20
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
l1 - accuracy: 0.
533/533 [==============================] - 59s 111ms/step - loss: 0.5283 - accuracy: 0.
Epoch 10/20
533/533 [==============================] - 59s 110ms/step - loss: 0.5191 - accuracy: 0.
Epoch 11/20
533/533 [==============================] - 59s 110ms/step - loss: 0.5127 - accuracy: 0.
Epoch 12/20
533/533 [==============================] - 60s 113ms/step - loss: 0.5080 - accuracy: 0.
Epoch 13/20
533/533 [==============================] - 69s 129ms/step - loss: 0.4807 - accuracy: 0.
Epoch 14/20
533/533 [==============================] - 59s 110ms/step - loss: 0.4921 - accuracy: 0.
Epoch 15/20
533/533 [==============================] - 68s 128ms/step - loss: 0.4701 - accuracy: 0.
Epoch 16/20
533/533 [==============================] - 69s 129ms/step - loss: 0.4924 - accuracy: 0.
Epoch 17/20
533/533 [==============================] - 69s 129ms/step - loss: 0.4617 - accuracy: 0.
Epoch 18/20
533/533 [==============================] - 59s 110ms/step - loss: 0.4524 - accuracy: 0.
Epoch 19/20
533/533 [==============================] - 69s 129ms/step - loss: 0.4490 - accuracy: 0.
Epoch 20/20
533/533 [==============================] - 60s 113ms/step - loss: 0.4455 - accuracy: 0.
<keras.callbacks.History at 0x7fee419d4850>
```

```
from keras.preprocessing import image
from tensorflow.keras.utils import load_img, img_to_array


ImagePath='/content/animal (8).jpg'
test_image=keras.utils.load_img(ImagePath, target_size=(128,128))
test_image=keras.utils.img_to_array(test_image)

# import the cv2 library
import cv2
from google.colab.patches import cv2_imshow

# The function cv2.imread() is used to read an image.
img= cv2.imread('/content/animal (8).jpg',1)

# The function cv2.imshow() is used to display an image in a wind
print('Testing Image of CAT')
cv2_imshow(img)

# waitKey() waits for a key press to close the window and 0 speci
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
# cv2.destroyAllWindows() simply destroys all the windows we crea
#cv2.destroyAllWindows()
```

```
Testing Image of CAT
```

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

result = model.predict_generator(test_image,verbose=0)

print('Prediction is',ResultMap[np.argmax(result)])
```

```
Prediction is cats
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Colab paid products - Cancel contracts here

✓ 0s completed at 8:52 PM ● ✕

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.