

```
In [1]: from keras.utils import to_categorical
from keras.preprocessing.image import load_img
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
import os
import pandas as pd
import numpy as np
```

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

```
In [2]: TRAIN_DIR = 'train/'
TEST_DIR = 'test/'
```

```
In [3]: def createdataframe(dir):
    image_paths = []
    labels = []
    for label in os.listdir(dir):
        for imagename in os.listdir(os.path.join(dir,label)):
            image_paths.append(os.path.join(dir,label,imagename))
            labels.append(label)
        print(label, "completed")
    return image_paths,labels
```

```
In [4]: train = pd.DataFrame()
train['image'], train['label'] = createdataframe(TRAIN_DIR)
```

```
angry completed
disgust completed
fear completed
happy completed
neutral completed
sad completed
surprise completed
```

```
In [5]: print(train)
```

	image	label
0	train/angry\Training_10118481.jpg	angry
1	train/angry\Training_10120469.jpg	angry
2	train/angry\Training_10131352.jpg	angry
3	train/angry\Training_10161559.jpg	angry
4	train/angry\Training_1021836.jpg	angry
...
28704	train/surprise\Training_99916297.jpg	surprise
28705	train/surprise\Training_99924420.jpg	surprise
28706	train/surprise\Training_99937001.jpg	surprise
28707	train/surprise\Training_99951755.jpg	surprise
28708	train/surprise\Training_99984132.jpg	surprise

[28709 rows x 2 columns]

```
In [6]: test = pd.DataFrame()  
test['image'], test['label'] = createdataframe(TEST_DIR)
```

angry completed
disgust completed
fear completed
happy completed
neutral completed
sad completed
surprise completed

```
In [7]: print(test)
print(test['image'])
```

	image	label
0	test/angry\PrivateTest_10131363.jpg	angry
1	test/angry\PrivateTest_10304478.jpg	angry
2	test/angry\PrivateTest_1054527.jpg	angry
3	test/angry\PrivateTest_10590091.jpg	angry
4	test/angry\PrivateTest_1109992.jpg	angry
...
7173	test/surprise\PublicTest_98089595.jpg	surprise
7174	test/surprise\PublicTest_98567249.jpg	surprise
7175	test/surprise\PublicTest_98972870.jpg	surprise
7176	test/surprise\PublicTest_99242645.jpg	surprise
7177	test/surprise\PublicTest_99446963.jpg	surprise

```
[7178 rows x 2 columns]
0      test/angry\PrivateTest_10131363.jpg
1      test/angry\PrivateTest_10304478.jpg
2      test/angry\PrivateTest_1054527.jpg
3      test/angry\PrivateTest_10590091.jpg
4      test/angry\PrivateTest_1109992.jpg
...
7173   test/surprise\PublicTest_98089595.jpg
7174   test/surprise\PublicTest_98567249.jpg
7175   test/surprise\PublicTest_98972870.jpg
7176   test/surprise\PublicTest_99242645.jpg
7177   test/surprise\PublicTest_99446963.jpg
Name: image, Length: 7178, dtype: object
```

```
In [8]: from tqdm.notebook import tqdm
```

```
In [9]: def extract_features(images):
        features = []
        for image in tqdm(images):
            img = load_img(image, grayscale = True )
            img = np.array(img)
            features.append(img)
        features = np.array(features)
        features = features.reshape(len(features),48,48,1)
        return features
```

```
In [10]: train_features = extract_features(train['image'])
```

```
0%|          | 0/28709 [00:00<?, ?it/s]
```

C:\Anaconda\Lib\site-packages\keras\src\utils\image_utils.py:409: UserWarning: grayscale is deprecated. Please use color_mode = "grayscale"
warnings.warn(

```
In [11]: test_features = extract_features(test['image'])
```

```
0%|          | 0/7178 [00:00<?, ?it/s]
```

```
In [12]: x_train = train_features/255.0
        x_test = test_features/255.0
```

```
In [13]: from sklearn.preprocessing import LabelEncoder
```

```
In [14]: le = LabelEncoder()
        le.fit(train['label'])
```

```
Out[14]: LabelEncoder()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [15]: y_train = le.transform(train['label'])
        y_test = le.transform(test['label'])
```

```
In [16]: y_train = to_categorical(y_train,num_classes = 7)
y_test = to_categorical(y_test,num_classes = 7)
```

```
In [17]: model = Sequential()
# convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
# fully connected layers
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))
```

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

```
In [18]: model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = 'accuracy' )
```

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\optimizers__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

```
In [19]: model.fit(x= x_train,y = y_train, batch_size = 128, epochs = 5, validation_data = (x_test,y_test))
```

Epoch 1/5

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Anaconda\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

225/225 [=====] - 123s 541ms/step - loss: 1.8202 - accuracy: 0.2452 - val_loss: 1.7881 - val_accuracy: 0.2534

Epoch 2/5

225/225 [=====] - 117s 522ms/step - loss: 1.7463 - accuracy: 0.2793 - val_loss: 1.6509 - val_accuracy: 0.3328

Epoch 3/5

225/225 [=====] - 117s 521ms/step - loss: 1.6081 - accuracy: 0.3615 - val_loss: 1.4860 - val_accuracy: 0.4160

Epoch 4/5

225/225 [=====] - 117s 522ms/step - loss: 1.4904 - accuracy: 0.4208 - val_loss: 1.3627 - val_accuracy: 0.4778

Epoch 5/5

225/225 [=====] - 119s 528ms/step - loss: 1.4261 - accuracy: 0.4491 - val_loss: 1.3205 - val_accuracy: 0.4884

```
Out[19]: <keras.src.callbacks.History at 0x131aed83a10>
```

```
In [20]: model_json = model.to_json()
with open("emotiondetector.json",'w') as json_file:
    json_file.write(model_json)
model.save("emotiondetector.h5")
```

C:\Anaconda\Lib\site-packages\keras\src\engine\training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.

```
    saving_api.save_model(
```

```
In [22]: from keras.models import model_from_json
```

```
In [24]: json_file = open("emotiondetector.json", "r")
model_json = json_file.read()
json_file.close()
model = model_from_json(model_json)
model.load_weights("emotiondetector.h5")
```

```
In [25]: label = ['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise']
```

```
In [26]: def ef(image):
    img = load_img(image, grayscale = True )
    feature = np.array(img)
    feature = feature.reshape(1,48,48,1)
    return feature/255.0
```

```
In [31]: image = 'train/sad/Training_1369050.jpg'
print("original image is of sad")
img = ef(image)
pred = model.predict(img)
pred_label = label[pred.argmax()]
print("model prediction is ",pred_label)
```

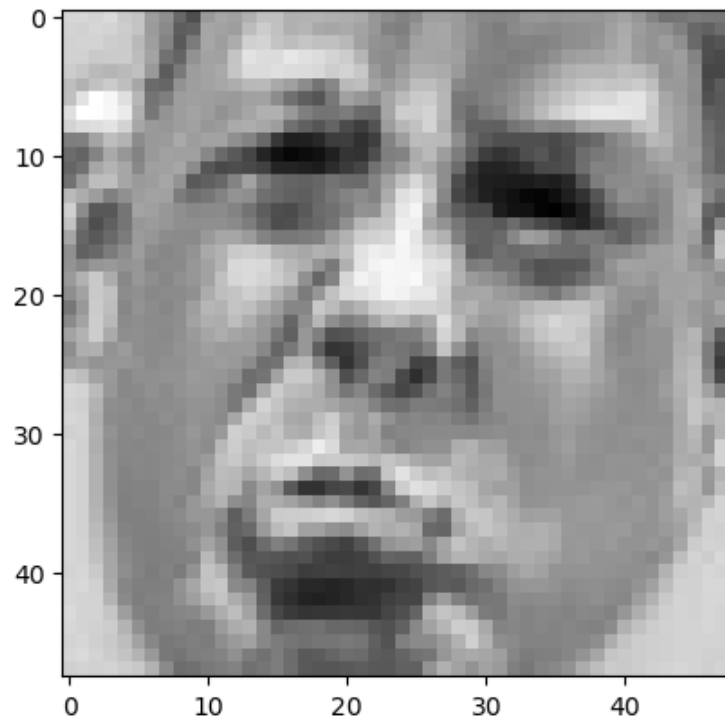
```
original image is of sad
1/1 [=====] - 0s 20ms/step
model prediction is  sad
```

```
In [32]: import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [34]: image = 'train/sad/Training_1369050.jpg'
print("original image is of sad")
img = ef(image)
pred = model.predict(img)
pred_label = label[pred.argmax()]
print("model prediction is ",pred_label)
plt.imshow(img.reshape(48,48),cmap='gray')
```

```
original image is of sad
1/1 [=====] - 0s 23ms/step
model prediction is  sad
```

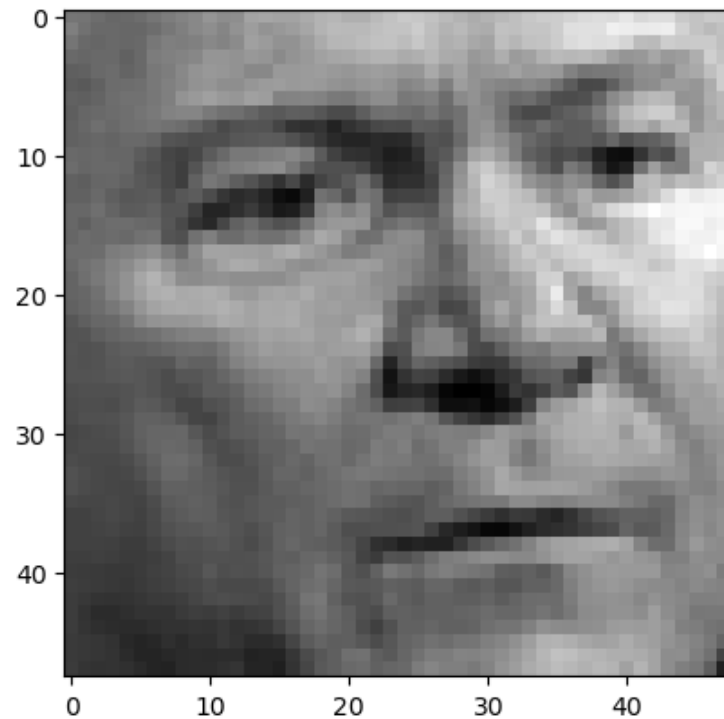
```
Out[34]: <matplotlib.image.AxesImage at 0x131ae8bdd90>
```




```
In [35]: image = 'train/fear/Training_10133194.jpg'
print("original image is of fear")
img = ef(image)
pred = model.predict(img)
pred_label = label[pred.argmax()]
print("model prediction is ",pred_label)
plt.imshow(img.reshape(48,48),cmap='gray')
```

```
original image is of fear
1/1 [=====] - 0s 21ms/step
model prediction is  sad
```

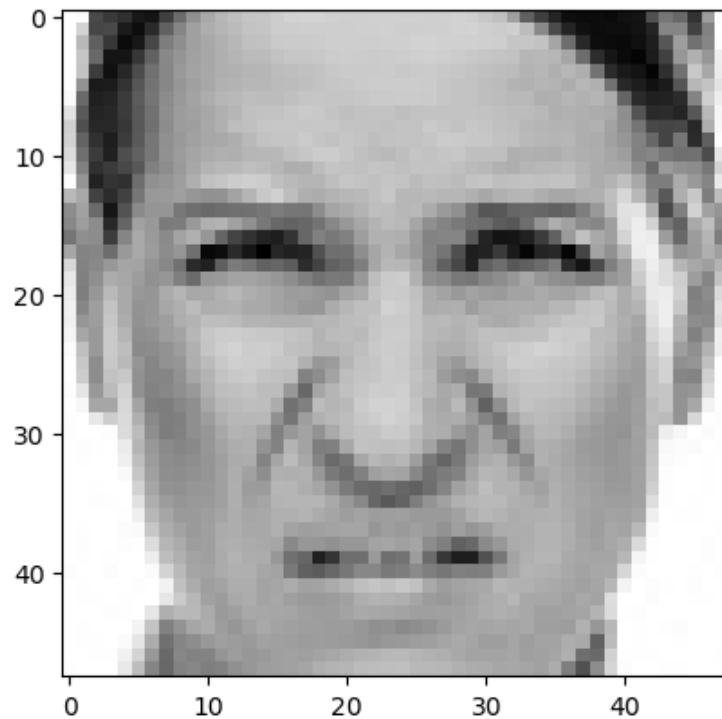
```
Out[35]: <matplotlib.image.AxesImage at 0x131ce4ff3d0>
```



```
In [36]: image = 'train/disgust/Training_680349.jpg'
print("original image is of disgust")
img = ef(image)
pred = model.predict(img)
pred_label = label[pred.argmax()]
print("model prediction is ",pred_label)
plt.imshow(img.reshape(48,48),cmap='gray')

original image is of disgust
1/1 [=====] - 0s 20ms/step
model prediction is  angry
```

Out[36]: <matplotlib.image.AxesImage at 0x131ce494b10>



In []:

