

## IMPORTING LIBRARIES

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
```

## Importing Deep Learning Libraries

```
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import
Dense, Input, Dropout, GlobalAveragePooling2D, Flatten, Conv2D, BatchNormali
zation, Activation, MaxPooling2D
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam, SGD, RMSprop
```

```
2024-03-30 07:58:21.154785: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable
to register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
2024-03-30 07:58:21.154925: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
2024-03-30 07:58:21.329835: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable
to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
```

## DISPLAYING IMAGES

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
expression = 'happy'
```

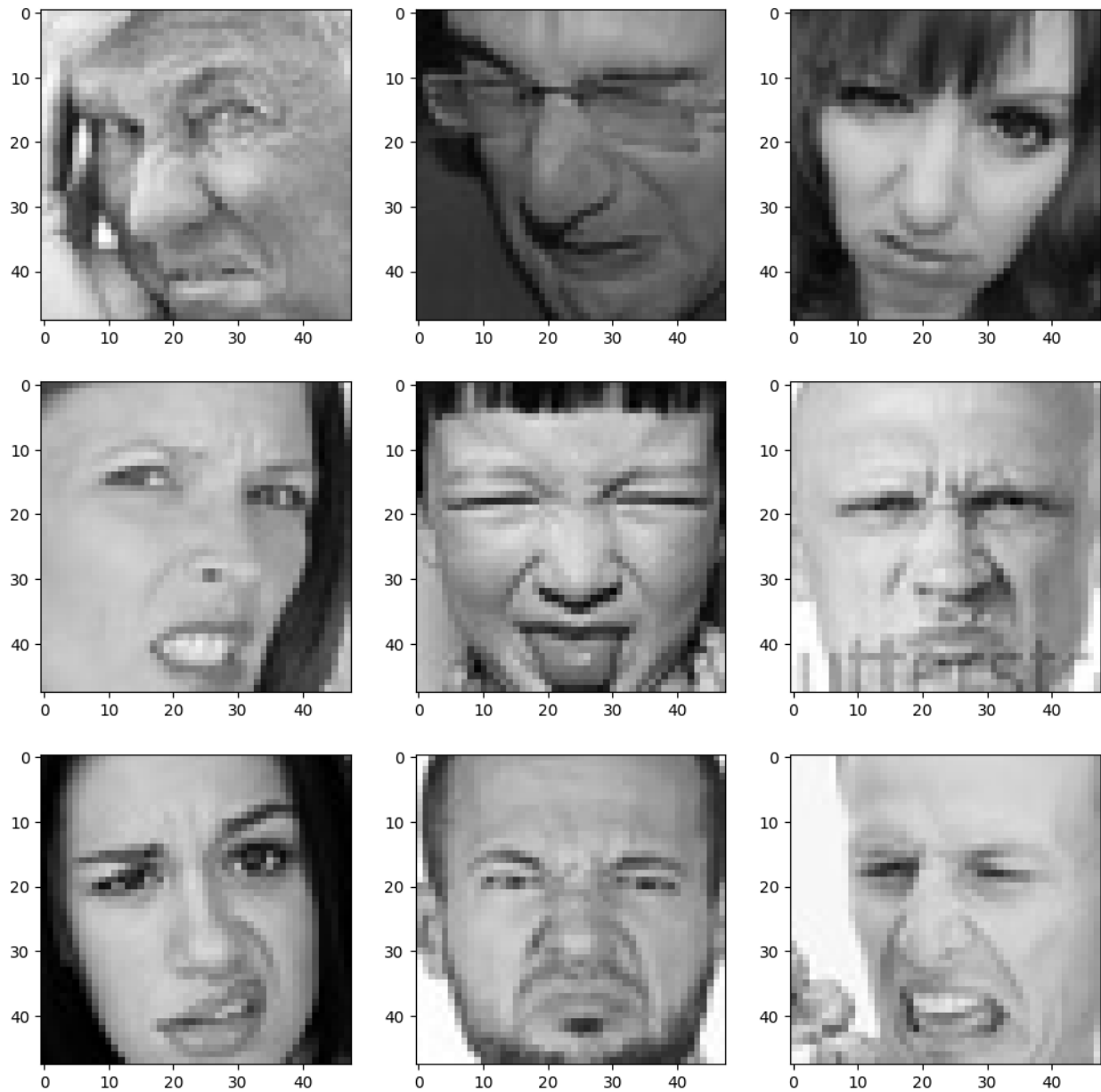
```
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img(folder_path+"train/"+expression+"/"+
                    os.listdir(folder_path + "train/" + expression)[i],
                    target_size=(picture_size, picture_size))
    plt.imshow(img)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
import seaborn as sns
import os
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
expression = 'disgust'

plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img(folder_path+"train/"+expression+"/"+
                    os.listdir(folder_path + "train/" + expression)[i],
                    target_size=(picture_size, picture_size))
    plt.imshow(img)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
expression = 'fear'

plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
```

```
img = load_img(folder_path+"train/"+expression+"/"+
               os.listdir(folder_path + "train/" + expression)[i],
               target_size=(picture_size, picture_size))
plt.imshow(img)
plt.show()
```



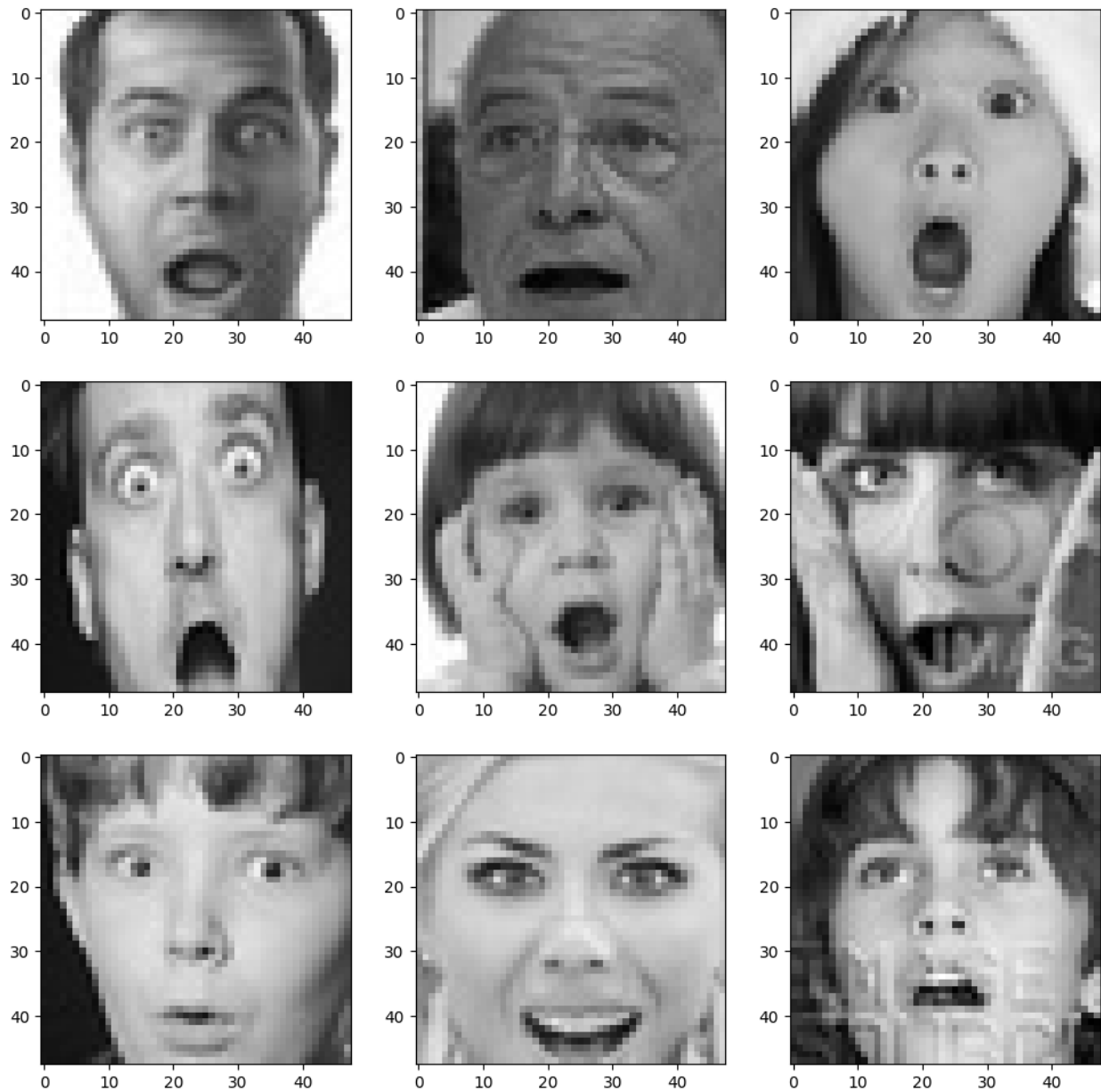
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
picture_size = 48
```

```
folder_path = "../input/face-expression-recognition-dataset/images/"  
expression = 'sad'
```

```
plt.figure(figsize= (12,12))  
for i in range(1, 10, 1):  
    plt.subplot(3,3,i)  
    img = load_img(folder_path+"train/"+expression+"/"+  
                    os.listdir(folder_path + "train/" + expression)[i],  
                    target_size=(picture_size, picture_size))  
    plt.imshow(img)  
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
expression = 'surprise'
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img(folder_path+"train/"+expression+"/"+
                    os.listdir(folder_path + "train/" + expression)[i],
                    target_size=(picture_size, picture_size))
    plt.imshow(img)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
expression = 'neutral'
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
```



```
img = load_img(folder_path+"train/"+expression+"/"+  
               os.listdir(folder_path + "train/" + expression)[i],  
               target_size=(picture_size, picture_size))  
plt.imshow(img)  
plt.show()
```

