Visualising Dataset

```
from tensorflow.keras.utils import load_img, img_to_array
import os
import matplotlib.pyplot as plt
plt.style.use('dark_background')
```

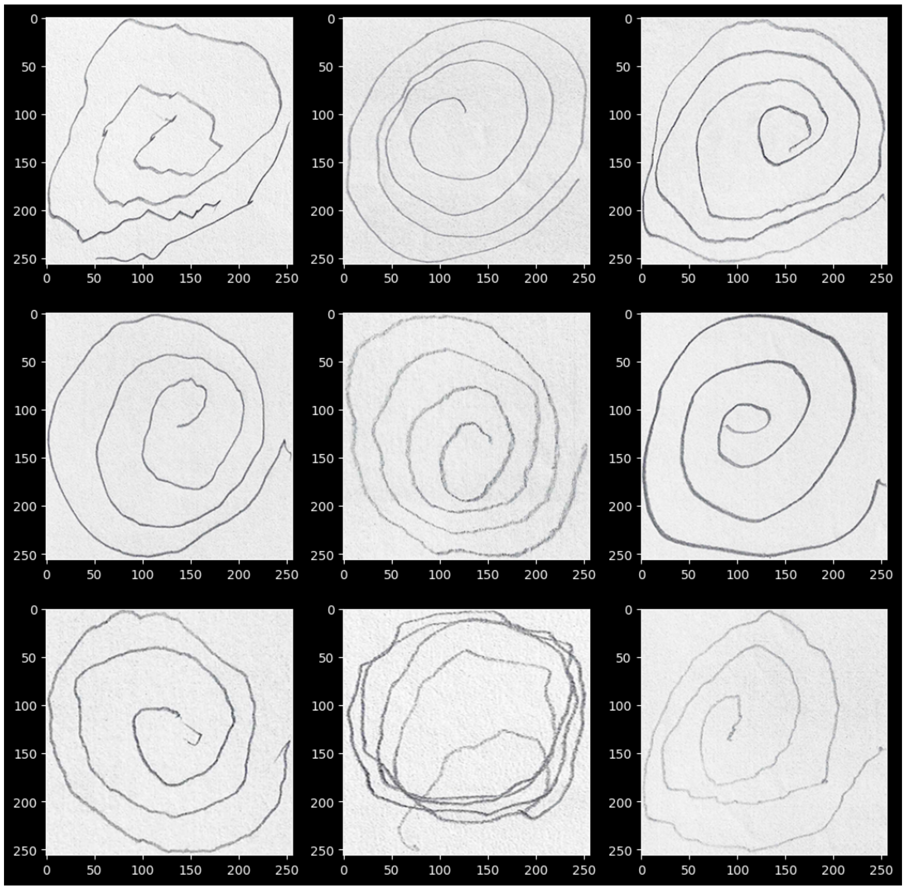
```
import zipfile

# Upload the zip file to Google Drive
zip_file_name = 'archive (2).zip'

# Unzip the file
with zipfile.ZipFile(zip_file_name, 'r') as zip_file:
    zip_file.extractall('unzipped_files')
```

Spiral (Healthy)

```
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img("/content/unzipped_files/drawings/spiral/training/healthy/"+
                  os.listdir("/content/unzipped_files/drawings/spiral/training/healthy")[i])
    plt.imshow(img)
plt.show()
```

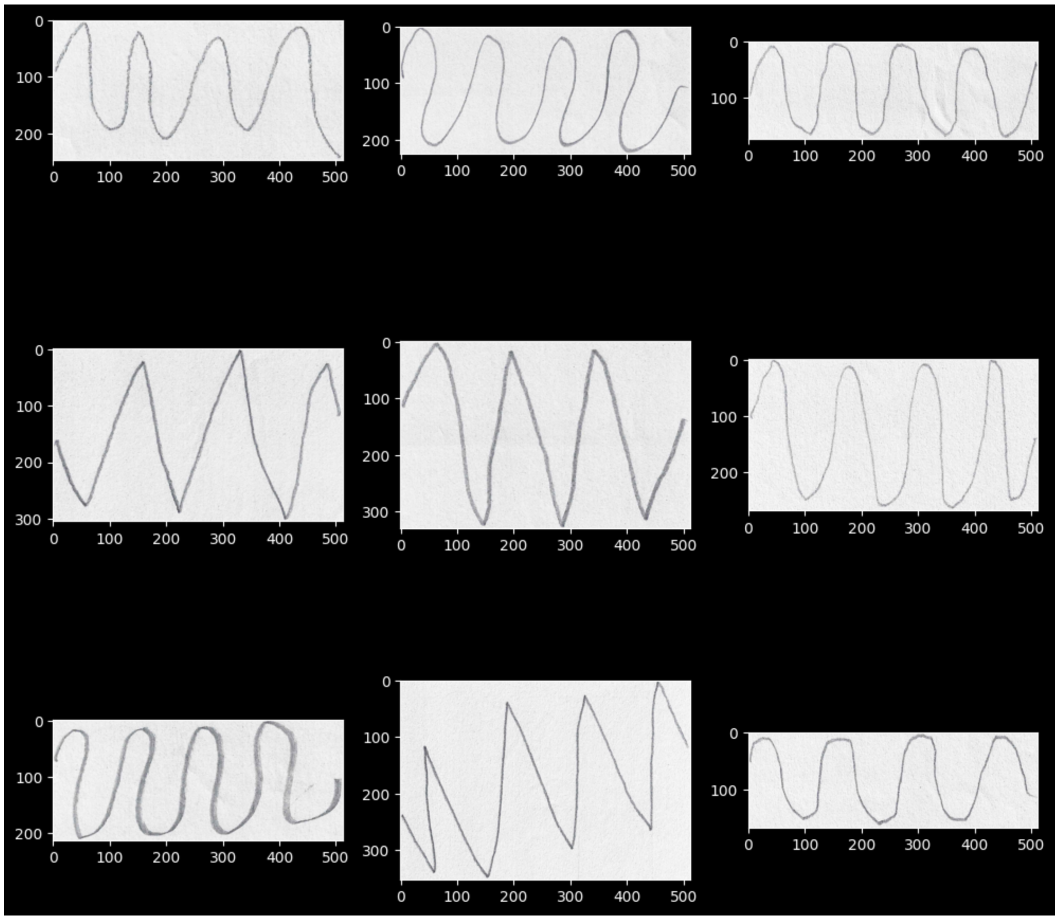
Spiral (Parkinson)

```
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img("/content/unzipped_files/drawings/spiral/training/parkinson/"+
                  os.listdir("/content/unzipped_files/drawings/spiral/training/parkinson")[i])
    plt.imshow(img)
plt.show()
```



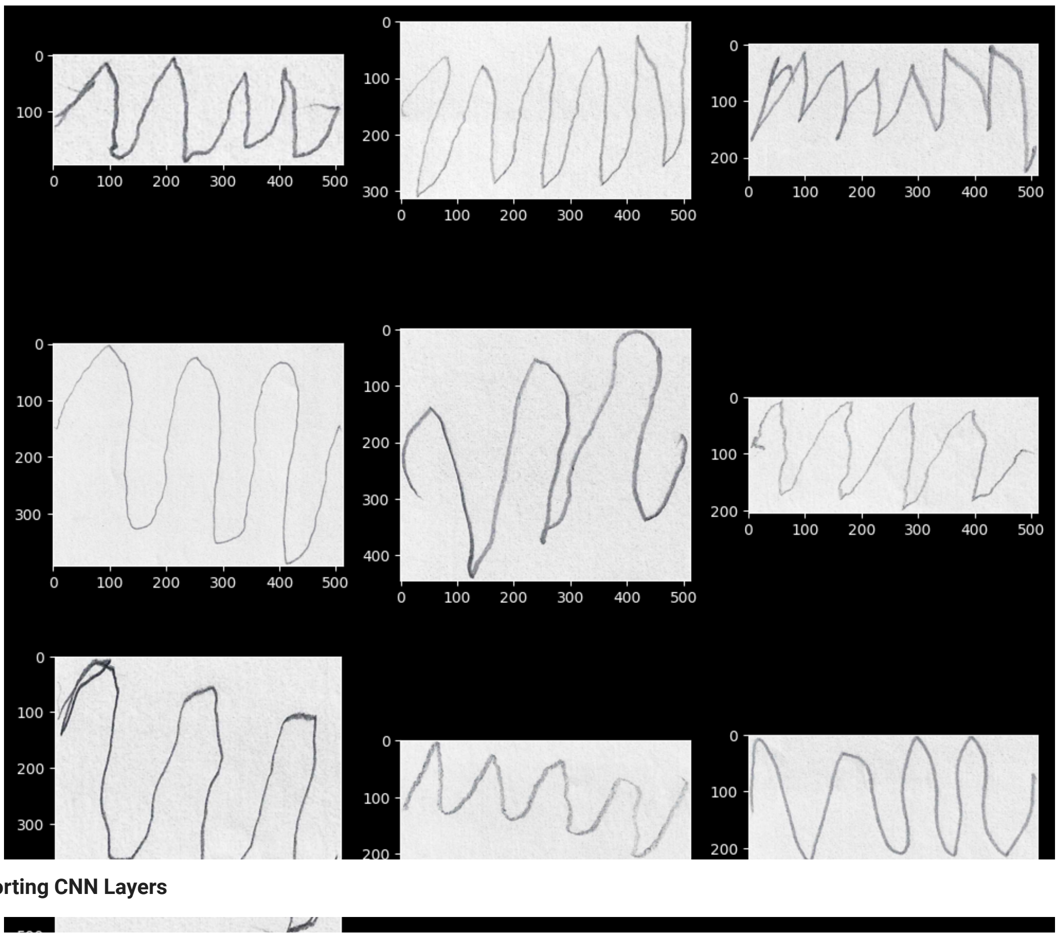
Wave (Healthy)

```
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img("/content/unzipped_files/drawings/wave/training/healthy/"+
                  os.listdir("/content/unzipped_files/drawings/wave/training/healthy")[i])
    plt.imshow(img)
plt.show()
```



Wave (Parkinson)

```
plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img("/content/unzipped_files/drawings/wave/training/parkinson/"+
                  os.listdir("/content/unzipped_files/drawings/wave/training/parkinson")[i])
    plt.imshow(img)
plt.show()
```



Importing CNN Layers

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

Building Classifier

```
classifier=Sequential()
classifier.add(Conv2D(32,(3,3),input_shape=(128, 128, 3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Conv2D(32,(3,3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Flatten())
classifier.add(Dense(activation='relu',units=128))
classifier.add(Dense(activation='sigmoid',units=1))
```

Image Data Generation

```
from keras.preprocessing.image import ImageDataGenerator


train_datagen = ImageDataGenerator(rescale = 1./255,
                                  shear_range = 0.2,
                                  zoom_range = 0.2,
                                  horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)


spiral_train_generator = train_datagen.flow_from_directory('/content/unzipped_files/drawings/spiral/training',
                                                          target_size = (128,128),
                                                          batch_size = 32,
                                                          class_mode = 'binary')

spiral_test_generator = test_datagen.flow_from_directory('/content/unzipped_files/drawings/spiral/testing',
                                                          target_size = (128,128),
                                                          batch_size = 32,
                                                          class_mode = 'binary')
```

```
Found 72 images belonging to 2 classes.
Found 30 images belonging to 2 classes.
```

```
wave_train_generator = train_datagen.flow_from_directory('/content/unzipped_files/drawings/wave/training',
                                                    target_size = (128,128),
                                                    batch_size = 32,
                                                    class_mode = 'binary')

wave_test_generator = test_datagen.flow_from_directory('/content/unzipped_files/drawings/wave/testing',
                                                    target_size = (128,128),
                                                    batch_size = 32,
                                                    class_mode = 'binary')
```

```
Found 72 images belonging to 2 classes.
Found 30 images belonging to 2 classes.
```

Fitting The Model with Data

```
from keras.optimizers import Adam

from keras.callbacks import EarlyStopping, ReduceLROnPlateau

early_stopping = EarlyStopping(monitor='val_loss',
                          min_delta=0,
                          patience=3,
                          verbose=1,
                          restore_best_weights=True
                          )

reduce_learningrate = ReduceLROnPlateau(monitor='val_loss',
                              factor=0.2,
                              patience=3,
                              verbose=1,
                              min_delta=0.0001)

callbacks_list = [early_stopping,reduce_learningrate]

epochs = 48

classifier.compile(loss='binary_crossentropy',
              optimizer = Adam(lr=0.001),
              metrics=['accuracy'])
```

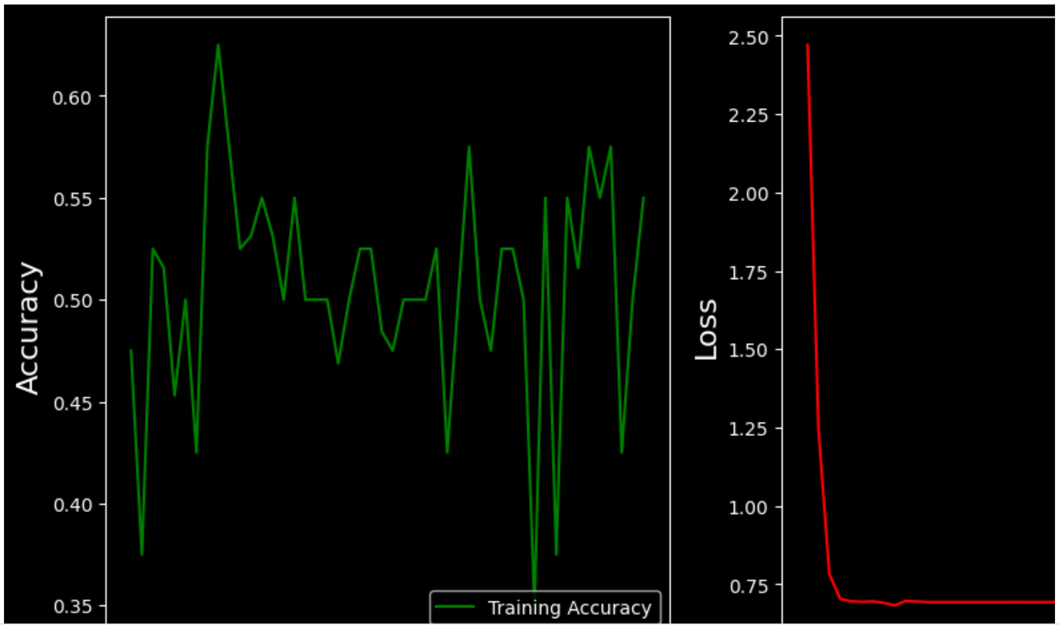
```
/usr/local/lib/python3.10/dist-packages/keras/optimizers/legacy/adam.py:117: UserWarning: The `lr` argument is deprecated, use `learning
  super().__init__(name, **kwargs)
```

```
history = classifier.fit_generator(
        spiral_train_generator,
        steps_per_epoch=spiral_train_generator.n//spiral_train_generator.batch_size,
        epochs=48,
        validation_data=spiral_test_generator,
        validation_steps=spiral_test_generator.n//spiral_test_generator.batch_size,
        callbacks=callbacks_list)
```

```
2/2 [==============================] - ETA: 0s - loss: 0.6932 - accuracy: 0.3750WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 888ms/step - loss: 0.6932 - accuracy: 0.3750 - lr: 0.0010
Epoch 41/48
2/2 [==============================] - ETA: 0s - loss: 0.6931 - accuracy: 0.5500WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 342ms/step - loss: 0.6931 - accuracy: 0.5500 - lr: 0.0010
Epoch 42/48
2/2 [==============================] - ETA: 0s - loss: 0.6931 - accuracy: 0.5156WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 3s 1s/step - loss: 0.6931 - accuracy: 0.5156 - lr: 0.0010
Epoch 43/48
2/2 [==============================] - ETA: 0s - loss: 0.6930 - accuracy: 0.5750WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 2s 1s/step - loss: 0.6930 - accuracy: 0.5750 - lr: 0.0010
Epoch 44/48
2/2 [==============================] - ETA: 0s - loss: 0.6930 - accuracy: 0.5500WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 275ms/step - loss: 0.6930 - accuracy: 0.5500 - lr: 0.0010
Epoch 45/48
2/2 [==============================] - ETA: 0s - loss: 0.6929 - accuracy: 0.5750WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 989ms/step - loss: 0.6929 - accuracy: 0.5750 - lr: 0.0010
Epoch 46/48
2/2 [==============================] - ETA: 0s - loss: 0.6935 - accuracy: 0.4250WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 312ms/step - loss: 0.6935 - accuracy: 0.4250 - lr: 0.0010
Epoch 47/48
2/2 [==============================] - ETA: 0s - loss: 0.6932 - accuracy: 0.5000WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 977ms/step - loss: 0.6932 - accuracy: 0.5000 - lr: 0.0010
Epoch 48/48
2/2 [==============================] - ETA: 0s - loss: 0.6928 - accuracy: 0.5500WARNING:tensorflow:Early stopping conditioned on metr
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,ac
2/2 [==============================] - 1s 919ms/step - loss: 0.6928 - accuracy: 0.5500 - lr: 0.0010
```

Plotting Accuracy and Loss

```
plt.style.use('dark_background')
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.ylabel('Accuracy', fontsize=16)
plt.plot(history.history['accuracy'], label='Training Accuracy', color = 'green')
plt.legend(loc='lower right')



plt.subplot(1,2,2)
plt.ylabel('Loss', fontsize=16)
plt.plot(history.history['loss'], label='Training Loss', color = 'red')
plt.legend(loc='lower right')
plt.show()
```



✓ 0s completed at 11:43