

```
In [1]: import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
In [2]: from keras.preprocessing.image import ImageDataGenerator

Using TensorFlow backend.
```

```
In [3]: train_data_generator=ImageDataGenerator(rescale = 1./255,
                                              shear_range = 0.2,
                                              zoom_range = 0.2,
                                              horizontal_flip = True)
```

```
In [4]: test_data_generator = ImageDataGenerator(rescale = 1./255)
```

```
In [5]: training_set = train_data_generator.flow_from_directory('dataset/train',
                                                              target_size = (64, 64),
                                                              batch_size = 32,
                                                              class_mode = 'binary')

Found 2001 images belonging to 2 classes.
```

```
In [6]: test_set = test_data_generator.flow_from_directory('dataset/test',
                                                         target_size = (64, 64),
                                                         batch_size = 32,
                                                         class_mode = 'binary')

Found 270 images belonging to 2 classes.
```

```
In [7]: cnn = tf.keras.models.Sequential()
```

```
In [8]: cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64, 64,3]))
```

```
In [9]: cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
```

```
In [10]: cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
```

```
In [11]: cnn.add(tf.keras.layers.Flatten())
```

```
In [12]: cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
```

```
In [13]: cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

```
In [14]: cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

```
In [15]: cnn.fit(x = training_set, validation_data = test_set, epochs = 25)

WARNING:tensorflow:sample_weight modes were coerced from
...
to
['...']
WARNING:tensorflow:sample_weight modes were coerced from
...
to
['...']
Train for 63 steps, validate for 9 steps
Epoch 1/25
63/63 [=====] - 133s 2s/step - loss: 0.3083 - accuracy: 0.8476 - val_loss: 0.1520
- val_accuracy: 0.9519
Epoch 2/25
63/63 [=====] - 109s 2s/step - loss: 0.1302 - accuracy: 0.9535 - val_loss: 0.2501
- val_accuracy: 0.8963
Epoch 3/25
63/63 [=====] - 109s 2s/step - loss: 0.1174 - accuracy: 0.9565 - val_loss: 0.1765
- val_accuracy: 0.9333
Epoch 4/25
63/63 [=====] - 109s 2s/step - loss: 0.1018 - accuracy: 0.9655 - val_loss: 0.0325
- val_accuracy: 0.9889
Epoch 5/25
63/63 [=====] - 109s 2s/step - loss: 0.0795 - accuracy: 0.9760 - val_loss: 0.2168
- val_accuracy: 0.9222
Epoch 6/25
63/63 [=====] - 108s 2s/step - loss: 0.0926 - accuracy: 0.9655 - val_loss: 0.0973
- val_accuracy: 0.9704
Epoch 7/25
63/63 [=====] - 108s 2s/step - loss: 0.0784 - accuracy: 0.9730 - val_loss: 0.1567
- val_accuracy: 0.9556
Epoch 8/25
63/63 [=====] - 108s 2s/step - loss: 0.0643 - accuracy: 0.9790 - val_loss: 0.1772
- val_accuracy: 0.9407
Epoch 9/25
63/63 [=====] - 108s 2s/step - loss: 0.0665 - accuracy: 0.9765 - val_loss: 0.5190
- val_accuracy: 0.8519
Epoch 10/25
63/63 [=====] - 107s 2s/step - loss: 0.0656 - accuracy: 0.9745 - val_loss: 0.1798
- val_accuracy: 0.9444
Epoch 11/25
63/63 [=====] - 107s 2s/step - loss: 0.0565 - accuracy: 0.9795 - val_loss: 0.1752
- val_accuracy: 0.9444
Epoch 12/25
63/63 [=====] - 107s 2s/step - loss: 0.0472 - accuracy: 0.9875 - val_loss: 0.1585
- val_accuracy: 0.9556
Epoch 13/25
63/63 [=====] - 108s 2s/step - loss: 0.0366 - accuracy: 0.9855 - val_loss: 0.1269
- val_accuracy: 0.9741
Epoch 14/25
63/63 [=====] - 109s 2s/step - loss: 0.0460 - accuracy: 0.9825 - val_loss: 0.1977
- val_accuracy: 0.9481
Epoch 15/25
63/63 [=====] - 108s 2s/step - loss: 0.0466 - accuracy: 0.9840 - val_loss: 0.1094
- val_accuracy: 0.9815
Epoch 16/25
63/63 [=====] - 109s 2s/step - loss: 0.0365 - accuracy: 0.9875 - val_loss: 0.3212
- val_accuracy: 0.9185
Epoch 17/25
63/63 [=====] - 109s 2s/step - loss: 0.0444 - accuracy: 0.9870 - val_loss: 0.0993
- val_accuracy: 0.9741
Epoch 18/25
63/63 [=====] - 109s 2s/step - loss: 0.0311 - accuracy: 0.9885 - val_loss: 0.1710
- val_accuracy: 0.9667
Epoch 19/25
63/63 [=====] - 109s 2s/step - loss: 0.0327 - accuracy: 0.9885 - val_loss: 0.1196
- val_accuracy: 0.9741
Epoch 20/25
63/63 [=====] - 108s 2s/step - loss: 0.0323 - accuracy: 0.9890 - val_loss: 0.2211
- val_accuracy: 0.9481
Epoch 21/25
63/63 [=====] - 109s 2s/step - loss: 0.0355 - accuracy: 0.9890 - val_loss: 0.2126
- val_accuracy: 0.9407
Epoch 22/25
63/63 [=====] - 109s 2s/step - loss: 0.0248 - accuracy: 0.9915 - val_loss: 0.3793
- val_accuracy: 0.9074
Epoch 23/25
63/63 [=====] - 108s 2s/step - loss: 0.0345 - accuracy: 0.9860 - val_loss: 0.2350
- val_accuracy: 0.9519
Epoch 24/25
63/63 [=====] - 108s 2s/step - loss: 0.0337 - accuracy: 0.9880 - val_loss: 0.2888
- val_accuracy: 0.9370
Epoch 25/25
63/63 [=====] - 109s 2s/step - loss: 0.0299 - accuracy: 0.9900 - val_loss: 0.3512
- val_accuracy: 0.9148
```

Out[15]: <tensorflow.python.keras.callbacks.History at 0x22b7a93d888>

In [ ]: