day1-045

June 25, 2024

```python
[9]: import tensorflow as tf
     from tensorflow import keras
     from tensorflow.keras import layers
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     IMG_SIZE = 244
     BATCH_SIZE = 32
```

```python
[10]: train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2) #␣
      ↪Correct the typo here


      train_generator = train_datagen.flow_from_directory(
          '/content/drive/MyDrive/Covid19-dataset/train',
          target_size=(IMG_SIZE, IMG_SIZE),
          batch_size= BATCH_SIZE,
          class_mode='binary',
          subset='training'
          )
      val_generator = train_datagen.flow_from_directory(
            '/content/drive/MyDrive/Covid19-dataset/train',
              target_size=(IMG_SIZE,IMG_SIZE),
              batch_size=BATCH_SIZE,
              class_mode='binary',
              subset='validation'
          )
```

```
Found 112 images belonging to 2 classes.
Found 28 images belonging to 2 classes.
```

```python
[11]: # Define the model
      model = keras.Sequential([
      layers.Conv2D(32, (3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
      layers.MaxPooling2D(2,2),
      layers.Conv2D(64,(3,3),activation='relu'),
      layers.MaxPooling2D(2,2),
      layers.Conv2D(128,(3,3),activation='relu'),
      layers.MaxPooling2D(2,2),
      layers.Flatten(),
```

```python
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid')
])
```

```python
[13]: model.compile(optimizer='adam',loss='binary_crossentropy', metrics=['accuracy'])
```

```python
[14]: model.fit(train_generator,validation_data=val_generator,epochs=5)
```

```
Epoch 1/5
4/4 [==============================] - 73s 16s/step - loss: 1.2154 - accuracy:
0.4821 - val_loss: 0.6860 - val_accuracy: 0.5000
Epoch 2/5
4/4 [==============================] - 17s 4s/step - loss: 0.6262 - accuracy:
0.6250 - val_loss: 0.4910 - val_accuracy: 0.7500
Epoch 3/5
4/4 [==============================] - 18s 5s/step - loss: 0.3452 - accuracy:
0.8393 - val_loss: 0.2418 - val_accuracy: 0.8571
Epoch 4/5
4/4 [==============================] - 18s 4s/step - loss: 0.5784 - accuracy:
0.9018 - val_loss: 0.3016 - val_accuracy: 0.9286
Epoch 5/5
4/4 [==============================] - 17s 5s/step - loss: 0.2539 - accuracy:
0.8750 - val_loss: 0.2658 - val_accuracy: 0.8929
```

```
[14]: <keras.src.callbacks.History at 0x7f76d12dca30>
```

```python
[15]: model.save("model.h5","label.txt")
```

```
/usr/local/lib/python3.10/dist-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(
```

```
[ ]:
```

```python
[ ]: from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```python
[21]: from tensorflow.keras.models import load_model
      from tensorflow.keras.preprocessing import image
      import numpy as np

      model = load_model('model.h5')
```

```
test_image_path = '/content/drive/MyDrive/Covid19-dataset/test/Viral Pneumonia/
  ↪0103.jpeg'
img = image.load_img(test_image_path, target_size=(244, 244)) # Change target
  ↪size to 244x244
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)


img_array = img_array / 255.0


predictions = model.predict(img_array)
print(predictions)
```

```
1/1 [==============================] - 0s 123ms/step
[[0.9120578]]
```

[22]:
```
if predictions < 0.5:
        print('It is a Normal')
else:
        print('It is a viral phneumonia')
```

```
It is a viral phneumonia
```