

day1-11

June 25, 2024

```
[ ]: 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
```

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

```
[ ]: train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Covid_Dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Covid_Dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

Found 54 images belonging to 4 classes.

Found 13 images belonging to 4 classes.

```
[ ]: # 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(),
layers.Dense(128,activation='relu'),
layers.Dense(1,activation='sigmoid') #output layer
])

```

```

[ ]: # Compile the model
model.compile(optimizer='adam',loss='binary_crossentropy', metrics=['accuracy'])

```

```

[ ]: model.fit(train_generator, validation_data=val_generator,epochs=5) # Fixed typo:
    ↪ train_genertor -> train_generator

```

```

Epoch 1/5
2/2 [=====] - 22s 10s/step - loss: 3.0716 - accuracy:
0.5556 - val_loss: 1.9279 - val_accuracy: 0.7500
Epoch 2/5
2/2 [=====] - 9s 4s/step - loss: 1.1207 - accuracy:
0.7500 - val_loss: 0.7385 - val_accuracy: 0.2500
Epoch 3/5
2/2 [=====] - 10s 6s/step - loss: 0.6941 - accuracy:
0.5463 - val_loss: 0.6129 - val_accuracy: 0.7500
Epoch 4/5
2/2 [=====] - 8s 4s/step - loss: 0.5962 - accuracy:
0.7500 - val_loss: 0.5629 - val_accuracy: 0.7500
Epoch 5/5
2/2 [=====] - 9s 5s/step - loss: 0.5803 - accuracy:
0.7500 - val_loss: 0.5775 - val_accuracy: 0.7500

```

```

[ ]: <keras.src.callbacks.History at 0x7a7201e6b4c0>

```

```

[ ]: 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(

```

```

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

#load the model
model = load_model('Model.h5')

```

```

#Load and preprocess the last image
test_image_path = '/content/drive/MyDrive/Covid_Dataset/Viral Pneumonia/0101.
    ↳jpeg' # store the path as a string
img = image.load_img(test_image_path, target_size=(244, 244)) # load image from
    ↳the path
img_array= image.img_to_array(img)
img_array= np.expand_dims(img_array, axis=0) # Fix: expand img_array, not img

#Add batch dimension
img_array = img_array / 255.0 #normalize the pixel value, create a copy

#Make predictions
prediction = model.predict(img_array)

#print the prediction
print(prediction)

```

```

1/1 [=====] - 0s 188ms/step
[[0.17097087]]

```

```

[ ]: # ... previous code ...

# Make predictions and store the result
predictions = model.predict(img_array)

# Print the prediction
print(predictions)

# Access the predicted class (assuming binary classification)
if predictions[0][0] < 0.5:
    print('It is a covid')
else:
    print('It is a viral pneumonia')

```

```

1/1 [=====] - 0s 52ms/step
[[0.17097087]]
It is a covid

```