


```
In [1]: import os # OS Library can be used for reading folders, sub-folders or files inside a directory.  
import cv2  
import tensorflow as tf  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sn; sn.set(font_scale=1.4)  
from sklearn.utils import shuffle  
from tqdm import tqdm # Can be used as a progress tracker
```

```
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype(["qint8", np.int8, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint16 = np.dtype(["qint16", np.int16, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype(["qint32", np.int32, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_resource = np.dtype(["resource", np.ubyte, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype(["qint8", np.int8, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint16 = np.dtype(["qint16", np.int16, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing
```

```
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype(["qint32", np.int32, 1])
C:\Users\shash\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_resource = np.dtype(["resource", np.ubyte, 1])
```

```
In [2]: class_names = ['ARDS', 'Bacteria', 'COVID', 'Normal', 'SARS', 'Streptococcus', 'Virus']
class_names_label = {class_name:i for i, class_name in enumerate(class_names)}
IMAGE_SIZE = (150, 150)
class_names_label
```

```
Out[2]: {'ARDS': 0,
'Bacteria': 1,
'COVID': 2,
'Normal': 3,
'SARS': 4,
'Streptococcus': 5,
'Virus': 6}
```

```
In [3]: paths = [r'S:\VIT AP\SummerInternship1\COVID 19\train', r'S:\VIT AP\SummerInternship1\COVID 19\test']
```

```
In [4]: def load_data(paths):
        output = []

        #Iterating through Train & Test Data Set.
        for path in paths:

            images = []
            labels = []

            print("Loading Started for ",path)

            # Iterate through each folder corresponding to a category
            for folder in os.listdir(path): # 6 times for each folder
                label = class_names_label[folder]

                # Iterate through each image in our folder
                # Example Path: ../Data/seg_train/seg_train/buildings
                for file in tqdm(os.listdir(os.path.join(path, folder))):

                    # Get the path name of the image
                    img_path = os.path.join(os.path.join(path, folder), file)

                    # Open and resize the img
                    image = cv2.imread(img_path)
                    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    image = cv2.resize(image, IMAGE_SIZE)

                    # Append the image and its corresponding label to the output
                    images.append(image)
                    labels.append(label)

            images = np.array(images, dtype = 'float32')
            labels = np.array(labels, dtype = 'int32')

            output.append((images, labels))

        return output
```

```
In [5]: (train_images, train_labels), (test_images, test_labels) = load_data(paths)
```

```
0%|                                     | 0/2 [00:00<?, ?it/
s]
```

Loading Started for S:\VIT AP\SummerInternship1\COVID 19\train

```
100%|████████████████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 9.16it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 73/73 [00:02<00:00, 28.19it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 69/69 [00:04<00:00, 14.51it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 84/84 [00:05<00:00, 14.12it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 4/4 [00:00<00:00, 12.50it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 4/4 [00:00<00:00, 13.85it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 53/53 [00:02<00:00, 25.11it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 6.72it/
s]
```

```
0%|                                     | 0/24 [00:00<?, ?it/
s]
```

Loading Started for S:\VIT AP\SummerInternship1\COVID 19\test

```
100%|████████████████████████████████████████████████████████████████████████████████| 24/24 [00:00<00:00, 37.17it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 12/12 [00:01<00:00, 10.44it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 24/24 [00:02<00:00, 11.22it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 11.48it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 17.96it/
s]
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 13/13 [00:00<00:00, 33.16it/
s]
```

```
In [6]: train_images, train_labels = shuffle(train_images, train_labels, random_state=25)
```

```
In [7]: train_images = train_images / 255.0  
test_images = test_images / 255.0
```

```
In [8]: model_scratch = tf.keras.Sequential([  
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),  
    tf.keras.layers.MaxPooling2D(2,2),  
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
    tf.keras.layers.Flatten(),  
    tf.keras.layers.Dense(128, activation=tf.nn.relu),  
    tf.keras.layers.Dense(84, activation=tf.nn.relu),  
    tf.keras.layers.Dense(7, activation=tf.nn.softmax)  
])
```

WARNING:tensorflow:From C:\Users\shash\anaconda3\lib\site-packages\tensorflow\python\ops\init_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.

Instructions for updating:

Call initializer instance with the dtype argument instead of passing it to the constructor

In [9]: `model_scratch.summary()`

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 128)	5308544
dense_1 (Dense)	(None, 84)	10836
dense_2 (Dense)	(None, 7)	595
Total params: 5,330,119		
Trainable params: 5,330,119		
Non-trainable params: 0		

In [10]: `model_scratch.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy'])`


```
In [11]: history = model_scratch.fit(train_images, train_labels, batch_size=1, epochs=20, validation_split = 0.15)
```

Train on 245 samples, validate on 44 samples

Epoch 1/20

245/245 [=====] - 14s 59ms/sample - loss: 1.5027 - acc: 0.4245 - val_loss: 1.5490 - val_acc: 0.4773

Epoch 2/20

245/245 [=====] - 14s 59ms/sample - loss: 1.2062 - acc: 0.5959 - val_loss: 1.3012 - val_acc: 0.5682

Epoch 3/20

245/245 [=====] - 14s 58ms/sample - loss: 0.8111 - acc: 0.6939 - val_loss: 1.0721 - val_acc: 0.7273

Epoch 4/20

245/245 [=====] - 14s 58ms/sample - loss: 0.5545 - acc: 0.7878 - val_loss: 0.9598 - val_acc: 0.7500

Epoch 5/20

245/245 [=====] - 14s 58ms/sample - loss: 0.3856 - acc: 0.8735 - val_loss: 0.9861 - val_acc: 0.7727

Epoch 6/20

245/245 [=====] - 15s 59ms/sample - loss: 0.5547 - acc: 0.8571 - val_loss: 1.2000 - val_acc: 0.6818

Epoch 7/20

245/245 [=====] - 14s 58ms/sample - loss: 0.2744 - acc: 0.9102 - val_loss: 1.3838 - val_acc: 0.7727

Epoch 8/20

245/245 [=====] - 14s 59ms/sample - loss: 0.2001 - acc: 0.9510 - val_loss: 1.6743 - val_acc: 0.7273

Epoch 9/20

245/245 [=====] - 14s 59ms/sample - loss: 0.0570 - acc: 0.9796 - val_loss: 2.3934 - val_acc: 0.6818

Epoch 10/20

245/245 [=====] - 14s 59ms/sample - loss: 0.0261 - acc: 0.9959 - val_loss: 2.0636 - val_acc: 0.7273

Epoch 11/20

245/245 [=====] - 14s 58ms/sample - loss: 0.0099 - acc: 1.0000 - val_loss: 2.4087 - val_acc: 0.79550s - loss: 0.0103 - a

Epoch 12/20

245/245 [=====] - 14s 59ms/sample - loss: 0.0325 - acc: 0.9837 - val_loss: 2.2127 - val_acc: 0.6591

Epoch 13/20

245/245 [=====] - 14s 58ms/sample - loss: 0.3549 - acc: 0.9265 - val_loss: 1.9850 - val_acc: 0.6136

Epoch 14/20

```
245/245 [=====] - 14s 58ms/sample - loss: 0.2220 - acc: 0.9224 - val_loss: 2.1951 - val_acc: 0.7727
Epoch 15/20
245/245 [=====] - 14s 58ms/sample - loss: 0.0261 - acc: 0.9918 - val_loss: 2.5387 - val_acc: 0.6364
Epoch 16/20
245/245 [=====] - 14s 57ms/sample - loss: 0.0040 - acc: 1.0000 - val_loss: 3.0415 - val_acc: 0.7273
Epoch 17/20
245/245 [=====] - 14s 58ms/sample - loss: 8.6185e-04 - acc: 1.0000 - val_loss: 3.4446 - val_acc: 0.7045
Epoch 18/20
245/245 [=====] - 14s 59ms/sample - loss: 3.0196e-04 - acc: 1.0000 - val_loss: 3.3584 - val_acc: 0.7045
Epoch 19/20
245/245 [=====] - 15s 61ms/sample - loss: 1.5438e-04 - acc: 1.0000 - val_loss: 3.5180 - val_acc: 0.7045
Epoch 20/20
245/245 [=====] - 15s 59ms/sample - loss: 9.5786e-05 - acc: 1.0000 - val_loss: 3.6061 - val_acc: 0.7273
```

```
In [12]: test_loss = model_scratch.evaluate(test_images, test_labels)
```

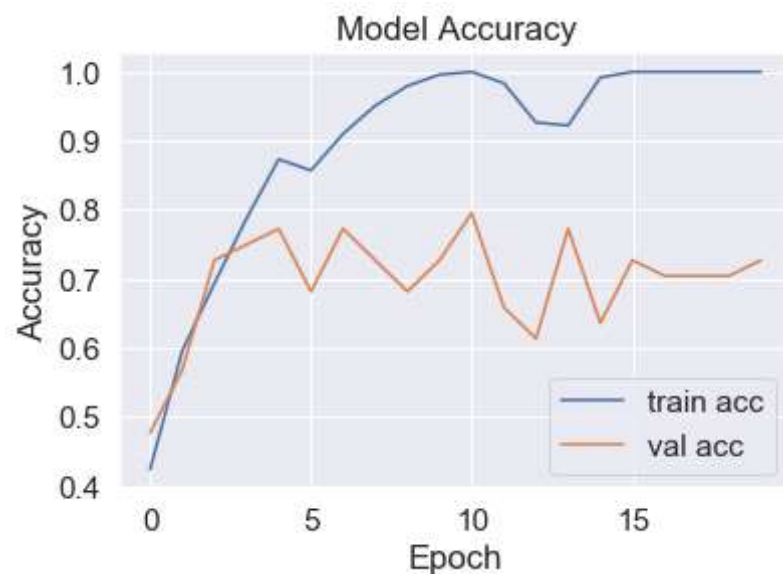
```
76/76 [=====] - 0s 4ms/sample - loss: 6.3880 - acc: 0.5263
```

```
In [14]: # Loss
plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.legend()
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()
plt.savefig('LossVal_loss')
```



<Figure size 432x288 with 0 Axes>

```
In [15]: # accuracies
plt.plot(history.history['acc'], label='train acc')
plt.plot(history.history['val_acc'], label='val acc')
plt.legend()
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.show()
plt.savefig('AccVal_acc')
```



<Figure size 432x288 with 0 Axes>

```
In [16]: model_scratch.save('Covidchest.h5')
```

Done!