\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_10 (InputLayer) [(None, 224, 224, 3)] 0

block1\_conv1 (Conv2D) (None, 224, 224, 64) 1792

block1\_conv2 (Conv2D) (None, 224, 224, 64) 36928

block1\_pool (MaxPooling2D) (None, 112, 112, 64) 0

block2\_conv1 (Conv2D) (None, 112, 112, 128) 73856

block2\_conv2 (Conv2D) (None, 112, 112, 128) 147584

block2\_pool (MaxPooling2D) (None, 56, 56, 128) 0

block3\_conv1 (Conv2D) (None, 56, 56, 256) 295168

block3\_conv2 (Conv2D) (None, 56, 56, 256) 590080

block3\_conv3 (Conv2D) (None, 56, 56, 256) 590080

block3\_pool (MaxPooling2D) (None, 28, 28, 256) 0

block4\_conv1 (Conv2D) (None, 28, 28, 512) 1180160

block4\_conv2 (Conv2D) (None, 28, 28, 512) 2359808

block4\_conv3 (Conv2D) (None, 28, 28, 512) 2359808

block4\_pool (MaxPooling2D) (None, 14, 14, 512) 0

block5\_conv1 (Conv2D) (None, 14, 14, 512) 2359808

block5\_conv2 (Conv2D) (None, 14, 14, 512) 2359808

block5\_conv3 (Conv2D) (None, 14, 14, 512) 2359808

block5\_pool (MaxPooling2D) (None, 7, 7, 512) 0

flatten\_4 (Flatten) (None, 25088) 0

dense\_4 (Dense) (None, 1) 25089

=================================================================

Total params: 14,739,777

Trainable params: 25,089

Non-trainable params: 14,714,688

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Epoch 1/100

143/143 [==============================] - 59s 409ms/step - loss: 0.2731 - accuracy: 0.8799 - val\_loss: 0.1960 - val\_accuracy: 0.9165 - lr: 0.0010

Epoch 2/100

143/143 [==============================] - 55s 383ms/step - loss: 0.1493 - accuracy: 0.9431 - val\_loss: 0.1607 - val\_accuracy: 0.9359 - lr: 0.0010

Epoch 3/100

143/143 [==============================] - 54s 378ms/step - loss: 0.1072 - accuracy: 0.9625 - val\_loss: 0.1174 - val\_accuracy: 0.9561 - lr: 0.0010

Epoch 4/100

143/143 [==============================] - 55s 383ms/step - loss: 0.0740 - accuracy: 0.9807 - val\_loss: 0.1221 - val\_accuracy: 0.9552 - lr: 0.0010

Epoch 5/100

143/143 [==============================] - 53s 372ms/step - loss: 0.0602 - accuracy: 0.9851 - val\_loss: 0.0991 - val\_accuracy: 0.9640 - lr: 0.0010

Epoch 6/100

143/143 [==============================] - 54s 374ms/step - loss: 0.0472 - accuracy: 0.9903 - val\_loss: 0.0979 - val\_accuracy: 0.9622 - lr: 0.0010

Epoch 7/100

143/143 [==============================] - 53s 373ms/step - loss: 0.0388 - accuracy: 0.9941 - val\_loss: 0.1117 - val\_accuracy: 0.9525 - lr: 0.0010

Epoch 8/100

143/143 [==============================] - 54s 374ms/step - loss: 0.0358 - accuracy: 0.9936 - val\_loss: 0.0936 - val\_accuracy: 0.9649 - lr: 0.0010

Epoch 9/100

143/143 [==============================] - 54s 374ms/step - loss: 0.0239 - accuracy: 0.9982 - val\_loss: 0.0988 - val\_accuracy: 0.9631 - lr: 0.0010

Epoch 10/100

143/143 [==============================] - 54s 374ms/step - loss: 0.0210 - accuracy: 0.9993 - val\_loss: 0.0906 - val\_accuracy: 0.9684 - lr: 0.0010

Epoch 11/100

143/143 [==============================] - 54s 373ms/step - loss: 0.0179 - accuracy: 0.9996 - val\_loss: 0.0932 - val\_accuracy: 0.9613 - lr: 0.0010

Epoch 12/100

143/143 [==============================] - 53s 372ms/step - loss: 0.0150 - accuracy: 0.9996 - val\_loss: 0.0875 - val\_accuracy: 0.9710 - lr: 0.0010

Epoch 13/100

143/143 [==============================] - 54s 373ms/step - loss: 0.0139 - accuracy: 0.9998 - val\_loss: 0.0932 - val\_accuracy: 0.9657 - lr: 0.0010

Epoch 14/100

143/143 [==============================] - 53s 370ms/step - loss: 0.0122 - accuracy: 0.9996 - val\_loss: 0.0922 - val\_accuracy: 0.9640 - lr: 0.0010

Epoch 15/100

143/143 [==============================] - 53s 368ms/step - loss: 0.0110 - accuracy: 0.9996 - val\_loss: 0.0871 - val\_accuracy: 0.9710 - lr: 0.0010

Epoch 16/100

143/143 [==============================] - 53s 371ms/step - loss: 0.0107 - accuracy: 0.9996 - val\_loss: 0.0883 - val\_accuracy: 0.9701 - lr: 0.0010

Epoch 17/100

143/143 [==============================] - 53s 370ms/step - loss: 0.0099 - accuracy: 0.9996 - val\_loss: 0.0875 - val\_accuracy: 0.9666 - lr: 0.0010

Epoch 18/100

143/143 [==============================] - 53s 371ms/step - loss: 0.0081 - accuracy: 0.9998 - val\_loss: 0.0961 - val\_accuracy: 0.9640 - lr: 0.0010

Epoch 19/100

143/143 [==============================] - 54s 377ms/step - loss: 0.0064 - accuracy: 0.9996 - val\_loss: 0.0869 - val\_accuracy: 0.9701 - lr: 1.0000e-04

Epoch 20/100

143/143 [==============================] - 54s 376ms/step - loss: 0.0061 - accuracy: 0.9998 - val\_loss: 0.0874 - val\_accuracy: 0.9710 - lr: 1.0000e-04

Epoch 21/100

143/143 [==============================] - 54s 376ms/step - loss: 0.0061 - accuracy: 0.9998 - val\_loss: 0.0867 - val\_accuracy: 0.9692 - lr: 1.0000e-04

Epoch 22/100

143/143 [==============================] - 54s 375ms/step - loss: 0.0061 - accuracy: 0.9996 - val\_loss: 0.0885 - val\_accuracy: 0.9701 - lr: 1.0000e-04

Epoch 23/100

143/143 [==============================] - 54s 376ms/step - loss: 0.0060 - accuracy: 0.9996 - val\_loss: 0.0868 - val\_accuracy: 0.9692 - lr: 1.0000e-04

Epoch 24/100

143/143 [==============================] - 54s 378ms/step - loss: 0.0059 - accuracy: 0.9998 - val\_loss: 0.0870 - val\_accuracy: 0.9692 - lr: 1.0000e-04

Epoch 25/100

143/143 [==============================] - 54s 376ms/step - loss: 0.0057 - accuracy: 0.9998 - val\_loss: 0.0870 - val\_accuracy: 0.9701 - lr: 1.0000e-05

Epoch 26/100

143/143 [==============================] - 54s 373ms/step - loss: 0.0057 - accuracy: 0.9998 - val\_loss: 0.0870 - val\_accuracy: 0.9692 - lr: 1.0000e-05

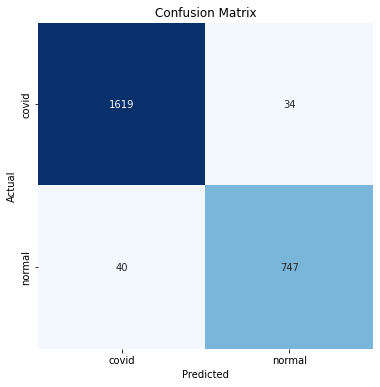
INFO:tensorflow:Assets written to: /content/drive/MyDrive/MendelyCTVGG16Split0.7noAug9000/assets

Test Loss: 0.09286

Test Accuracy: 96.97%

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:127: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you wish to review your current use, check the release note link for additional information.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>



Classification Report:

----------------------

precision recall f1-score support

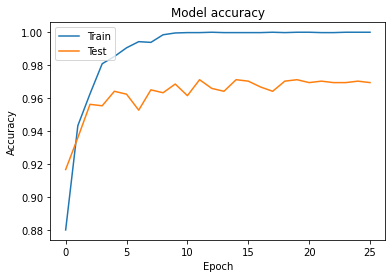
covid 0.98 0.98 0.98 1653

normal 0.96 0.95 0.95 787

accuracy 0.97 2440

macro avg 0.97 0.96 0.97 2440

weighted avg 0.97 0.97 0.97 2440



<matplotlib.legend.Legend at 0x7f9780ba6910>

