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

Layer (type) Output Shape Param #

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

input\_2 (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 (Flatten) (None, 25088) 0

dense (Dense) (None, 1) 25089

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

Total params: 14,739,777

Trainable params: 25,089

Non-trainable params: 14,714,688

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

Epoch 1/100

62/62 [==============================] - 349s 5s/step - loss: 0.2681 - accuracy: 0.8821 - val\_loss: 0.1133 - val\_accuracy: 0.9617 - lr: 0.0010

Epoch 2/100

62/62 [==============================] - 56s 899ms/step - loss: 0.0973 - accuracy: 0.9723 - val\_loss: 0.0771 - val\_accuracy: 0.9819 - lr: 0.0010

Epoch 3/100

62/62 [==============================] - 56s 910ms/step - loss: 0.0731 - accuracy: 0.9803 - val\_loss: 0.0801 - val\_accuracy: 0.9698 - lr: 0.0010

Epoch 4/100

62/62 [==============================] - 57s 923ms/step - loss: 0.0608 - accuracy: 0.9859 - val\_loss: 0.0568 - val\_accuracy: 0.9839 - lr: 0.0010

Epoch 5/100

62/62 [==============================] - 57s 913ms/step - loss: 0.0380 - accuracy: 0.9909 - val\_loss: 0.0484 - val\_accuracy: 0.9839 - lr: 0.0010

Epoch 6/100

62/62 [==============================] - 55s 890ms/step - loss: 0.0319 - accuracy: 0.9960 - val\_loss: 0.0560 - val\_accuracy: 0.9819 - lr: 0.0010

Epoch 7/100

62/62 [==============================] - 55s 884ms/step - loss: 0.0269 - accuracy: 0.9955 - val\_loss: 0.0420 - val\_accuracy: 0.9899 - lr: 0.0010

Epoch 8/100

62/62 [==============================] - 54s 878ms/step - loss: 0.0219 - accuracy: 0.9970 - val\_loss: 0.0410 - val\_accuracy: 0.9859 - lr: 0.0010

Epoch 9/100

62/62 [==============================] - 54s 874ms/step - loss: 0.0183 - accuracy: 0.9975 - val\_loss: 0.0384 - val\_accuracy: 0.9879 - lr: 0.0010

Epoch 10/100

62/62 [==============================] - 54s 873ms/step - loss: 0.0163 - accuracy: 0.9990 - val\_loss: 0.0387 - val\_accuracy: 0.9879 - lr: 0.0010

Epoch 11/100

62/62 [==============================] - 55s 884ms/step - loss: 0.0145 - accuracy: 0.9990 - val\_loss: 0.0370 - val\_accuracy: 0.9879 - lr: 0.0010

Epoch 12/100

62/62 [==============================] - 55s 892ms/step - loss: 0.0121 - accuracy: 1.0000 - val\_loss: 0.0378 - val\_accuracy: 0.9899 - lr: 0.0010

Epoch 13/100

62/62 [==============================] - 56s 904ms/step - loss: 0.0108 - accuracy: 0.9995 - val\_loss: 0.0328 - val\_accuracy: 0.9899 - lr: 0.0010

Epoch 14/100

62/62 [==============================] - 56s 904ms/step - loss: 0.0103 - accuracy: 1.0000 - val\_loss: 0.0425 - val\_accuracy: 0.9819 - lr: 0.0010

Epoch 15/100

62/62 [==============================] - 54s 875ms/step - loss: 0.0086 - accuracy: 1.0000 - val\_loss: 0.0418 - val\_accuracy: 0.9839 - lr: 0.0010

Epoch 16/100

62/62 [==============================] - 54s 873ms/step - loss: 0.0086 - accuracy: 1.0000 - val\_loss: 0.0546 - val\_accuracy: 0.9819 - lr: 0.0010

Epoch 17/100

62/62 [==============================] - 54s 869ms/step - loss: 0.0075 - accuracy: 1.0000 - val\_loss: 0.0323 - val\_accuracy: 0.9879 - lr: 1.0000e-04

Epoch 18/100

62/62 [==============================] - 54s 879ms/step - loss: 0.0069 - accuracy: 1.0000 - val\_loss: 0.0325 - val\_accuracy: 0.9879 - lr: 1.0000e-04

Epoch 19/100

62/62 [==============================] - 55s 878ms/step - loss: 0.0068 - accuracy: 1.0000 - val\_loss: 0.0329 - val\_accuracy: 0.9899 - lr: 1.0000e-04

Epoch 20/100

62/62 [==============================] - 54s 870ms/step - loss: 0.0068 - accuracy: 1.0000 - val\_loss: 0.0323 - val\_accuracy: 0.9879 - lr: 1.0000e-04

Epoch 21/100

62/62 [==============================] - 55s 893ms/step - loss: 0.0067 - accuracy: 1.0000 - val\_loss: 0.0325 - val\_accuracy: 0.9899 - lr: 1.0000e-05

Epoch 22/100

62/62 [==============================] - 56s 896ms/step - loss: 0.0067 - accuracy: 1.0000 - val\_loss: 0.0327 - val\_accuracy: 0.9899 - lr: 1.0000e-05

Epoch 23/100

62/62 [==============================] - 54s 872ms/step - loss: 0.0066 - accuracy: 1.0000 - val\_loss: 0.0328 - val\_accuracy: 0.9899 - lr: 1.0000e-05

Epoch 24/100

62/62 [==============================] - 54s 876ms/step - loss: 0.0066 - accuracy: 1.0000 - val\_loss: 0.0328 - val\_accuracy: 0.9899 - lr: 1.0000e-06

Epoch 25/100

62/62 [==============================] - 54s 870ms/step - loss: 0.0066 - accuracy: 1.0000 - val\_loss: 0.0328 - val\_accuracy: 0.9899 - lr: 1.0000e-06

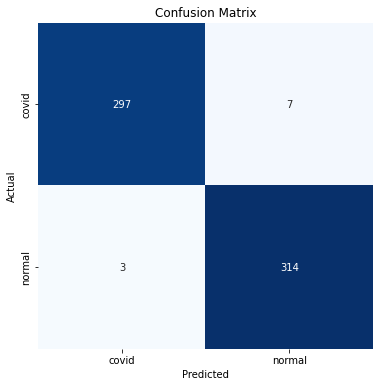
INFO:tensorflow:Assets written to: /content/drive/MyDrive/CTVGG16Split0.8noAug/assets

Test Loss: 0.06887

Test Accuracy: 98.39%

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:123: 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 0.99 0.98 0.98 304

normal 0.98 0.99 0.98 317

accuracy 0.98 621

macro avg 0.98 0.98 0.98 621

weighted avg 0.98 0.98 0.98 621

