data_aug_adam_val_95_64

October 8, 2021

1 Covid Classifier Model

1.0.1 Goals

Classify: - Normal CXR - Viral Pneumonia CXR - COVID CXR

1.1 Create Directories for Dataset

Separate the data to use later as generators.

[ ]:
```
# Matriz de confusion, cambiar learnings rates (learning rates dinamicos),␣
↪dropouts 0.3 & 0.2, Batch Normalization
# K-Fold o avg de modelos
import os

BASE_PATH = '/home/hivini/learn/research/new-covid'
ORIGINAL_DATASET_DIR = os.path.join(BASE_PATH, 'COVID-19_Radiography_Dataset')
ORIGINAL_VIRAL_DIR = os.path.join(ORIGINAL_DATASET_DIR, 'Viral Pneumonia')
ORIGINAL_COVID_DIR = os.path.join(ORIGINAL_DATASET_DIR, 'COVID')
ORIGINAL_NORMAL_DIR = os.path.join(ORIGINAL_DATASET_DIR, 'Normal')
DATASET_DIR = os.path.join(BASE_PATH, 'small_dataset')
TRAIN_DIR = os.path.join(DATASET_DIR, 'train')
VALIDATION_DIR = os.path.join(DATASET_DIR, 'validation')
TEST_DIR = os.path.join(DATASET_DIR, 'test')
TRAIN_VIRAL_DIR = os.path.join(TRAIN_DIR, 'viral_pneumonia')
TRAIN_COVID_DIR = os.path.join(TRAIN_DIR, 'covid')
TRAIN_NORMAL_DIR = os.path.join(TRAIN_DIR, 'normal')
VALIDATION_VIRAL_DIR = os.path.join(VALIDATION_DIR, 'viral_pneumonia')
VALIDATION_COVID_DIR = os.path.join(VALIDATION_DIR, 'covid')
VALIDATION_NORMAL_DIR = os.path.join(VALIDATION_DIR, 'normal')
TEST_VIRAL_DIR = os.path.join(TEST_DIR, 'viral_pneumonia')
TEST_COVID_DIR = os.path.join(TEST_DIR, 'covid')
TEST_NORMAL_DIR = os.path.join(TEST_DIR, 'normal')


def createDir(path: str) -> None:
    if not os.path.exists(path):
        os.mkdir(path)
```

```
createDir(DATASET_DIR)
createDir(TRAIN_DIR)
createDir(VALIDATION_DIR)
createDir(TEST_DIR)
createDir(TRAIN_VIRAL_DIR)
createDir(TRAIN_COVID_DIR)
createDir(TRAIN_NORMAL_DIR)
createDir(VALIDATION_VIRAL_DIR)
createDir(VALIDATION_COVID_DIR)
createDir(VALIDATION_NORMAL_DIR)
createDir(TEST_VIRAL_DIR)
createDir(TEST_COVID_DIR)
createDir(TEST_NORMAL_DIR)
```

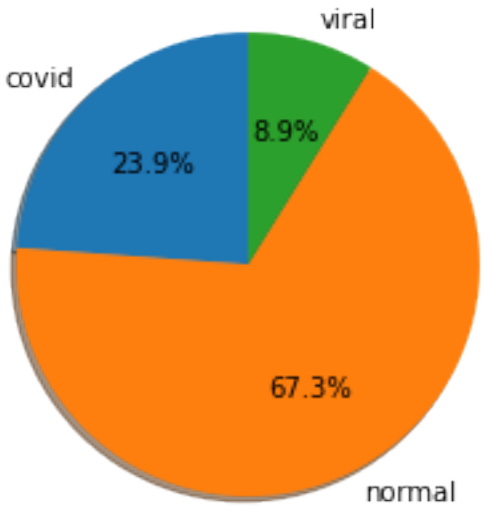
[ ]:
```
import numpy as np
import shutil


def generate_sets(source: str):
    allFiles = os.listdir(source)
    np.random.shuffle(allFiles)
    return np.split(np.array(allFiles), [int(len(allFiles)*0.7),␣
 ↪int(len(allFiles)*0.85)])


def saveAndSeparateFiles(src_dir: str, train_dir: str, val_dir: str, test_dir):
    train_fnames, val_fnames, test_fnames = generate_sets(src_dir)
    for fname in train_fnames:
        src = os.path.join(src_dir, fname)
        dst = os.path.join(train_dir, fname)
        shutil.copyfile(src, dst)

    for fname in val_fnames:
        src = os.path.join(src_dir, fname)
        dst = os.path.join(val_dir, fname)
        shutil.copyfile(src, dst)

    for fname in test_fnames:
        src = os.path.join(src_dir, fname)
        dst = os.path.join(test_dir, fname)
        shutil.copyfile(src, dst)

create = False
if create:
    saveAndSeparateFiles(ORIGINAL_NORMAL_DIR, TRAIN_NORMAL_DIR,
                         VALIDATION_NORMAL_DIR, TEST_NORMAL_DIR)
```

```
    saveAndSeparateFiles(ORIGINAL_COVID_DIR, TRAIN_COVID_DIR,
                         VALIDATION_COVID_DIR, TEST_COVID_DIR)
    saveAndSeparateFiles(ORIGINAL_VIRAL_DIR, TRAIN_VIRAL_DIR,
                         VALIDATION_VIRAL_DIR, TEST_VIRAL_DIR)
```

1.2 Counting our images

[ ]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
normal_train = tf.io.gfile.glob(TRAIN_NORMAL_DIR + '/*')
viral_train = tf.io.gfile.glob(TRAIN_VIRAL_DIR + '/*')
covid_train = tf.io.gfile.glob(TRAIN_COVID_DIR + '/*')

# Plotting Distribution of Each Classes
image_count = {'covid': len(covid_train), 'normal': len(
    normal_train), 'viral': len(viral_train)}
fig1, ax1 = plt.subplots()
ax1.pie(image_count.values(),
        labels=image_count.keys(),
        shadow=True,
        autopct='%1.1f%%',
        startangle=90)
plt.show()
```

```
2021-10-07 23:07:23.838838: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudart.so.10.1
```



1.3 Create our Covnet Model

In this case we are doing a multi class classification, our total clases are 3: - Viral CXR - Covid CXR - Normal CXR

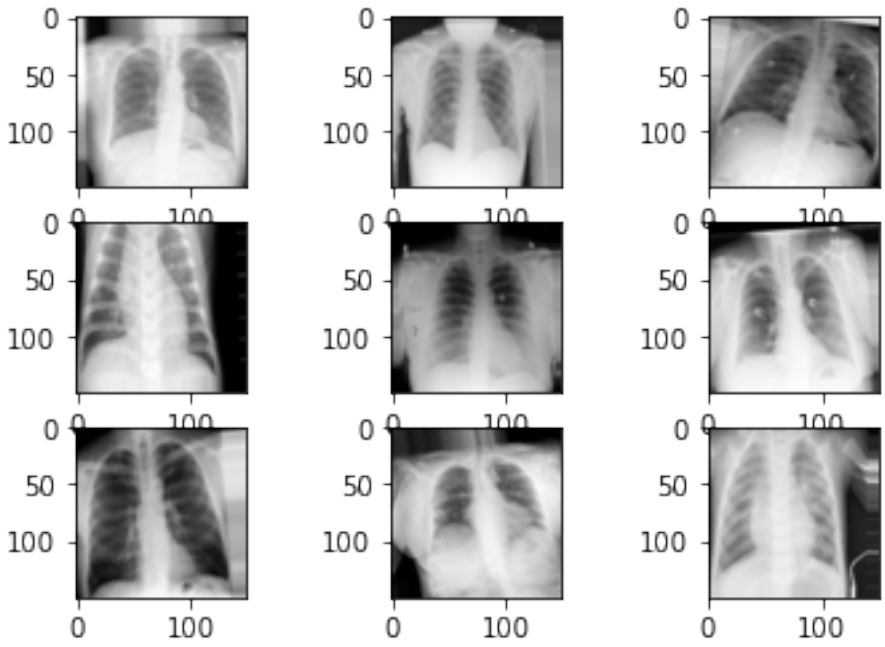
Our neural network will output neurons as 3 classes that will calculate the probability of being one using the softmax function.

[ ]:
```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    featurewise_center=False,  # set input mean to 0 over the dataset
    samplewise_center=False,  # set each sample mean to 0
    featurewise_std_normalization=False,  # divide inputs by std of the dataset
    samplewise_std_normalization=False,  # divide each input by its std
    zca_whitening=False,  # apply ZCA whitening
    # randomly rotate images in the range (degrees, 0 to 180)
    rotation_range=10,
    zoom_range=0.1,  # Randomly zoom image
    # randomly shift images horizontally (fraction of total width)
    width_shift_range=0.1,
    # randomly shift images vertically (fraction of total height)
    height_shift_range=0.1,
    horizontal_flip=False,  # randomly flip images
    vertical_flip=False  # randomly flip images
)

# train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
evaluate_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    TRAIN_DIR,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    color_mode='grayscale'
)

print(train_generator.class_indices)

validation_generator = test_datagen.flow_from_directory(
    VALIDATION_DIR,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    color_mode='grayscale'
```

```
)

print(validation_generator.class_indices)

test_generator = evaluate_datagen.flow_from_directory(
    TEST_DIR,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    color_mode='grayscale'
)

print(test_generator.class_indices)
```

```
Found 10606 images belonging to 3 classes.
{'covid': 0, 'normal': 1, 'viral_pneumonia': 2}
Found 2273 images belonging to 3 classes.
{'covid': 0, 'normal': 1, 'viral_pneumonia': 2}
Found 2274 images belonging to 3 classes.
{'covid': 0, 'normal': 1, 'viral_pneumonia': 2}
```

[ ]:
```
for X_batch, y_batch in train_generator:
        # create a grid of 3x3 images
        for i in range(0, 9):
                plt.subplot(330 + 1 + i)
                plt.imshow(X_batch[i].reshape(150, 150), cmap=plt.
 ↪get_cmap('gray'))
        # show the plot
        plt.show()
        break
```



[ ]:
```
from keras.layers import Conv2D, BatchNormalization, MaxPooling2D, Dropout,␣
↪Flatten, Dense
from keras.models import Sequential
from keras import backend

backend.clear_session()

model = Sequential()
model.add(Conv2D(64, (3, 3), activation='relu', input_shape=(150, 150, 1)))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(3, activation='softmax'))
```

```
model.summary()
```

```
2021-10-07 23:07:28.256635: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not
creating XLA devices, tf_xla_enable_xla_devices not set
2021-10-07 23:07:28.265024: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcuda.so.1
2021-10-07 23:07:28.654163: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.654442: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1720] Found device 0 with
properties:
pciBusID: 0000:01:00.0 name: NVIDIA GeForce RTX 2080 with Max-Q Design
computeCapability: 7.5
coreClock: 1.215GHz coreCount: 46 deviceMemorySize: 8.00GiB
deviceMemoryBandwidth: 357.69GiB/s
2021-10-07 23:07:28.654504: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudart.so.10.1
2021-10-07 23:07:28.700452: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcublas.so.10
2021-10-07 23:07:28.700542: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcublasLt.so.10
2021-10-07 23:07:28.727976: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcufft.so.10
2021-10-07 23:07:28.734165: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcurand.so.10
2021-10-07 23:07:28.781358: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcusolver.so.10
2021-10-07 23:07:28.794976: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcusparse.so.10
2021-10-07 23:07:28.891048: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudnn.so.7
2021-10-07 23:07:28.891989: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.893316: E
```

```
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.893738: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu
devices: 0
2021-10-07 23:07:28.894586: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2021-10-07 23:07:28.896693: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.897067: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1720] Found device 0 with
properties:
pciBusID: 0000:01:00.0 name: NVIDIA GeForce RTX 2080 with Max-Q Design
computeCapability: 7.5
coreClock: 1.215GHz coreCount: 46 deviceMemorySize: 8.00GiB
deviceMemoryBandwidth: 357.69GiB/s
2021-10-07 23:07:28.897118: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudart.so.10.1
2021-10-07 23:07:28.897162: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcublas.so.10
2021-10-07 23:07:28.897176: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcublasLt.so.10
2021-10-07 23:07:28.897187: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcufft.so.10
2021-10-07 23:07:28.897198: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcurand.so.10
2021-10-07 23:07:28.897209: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcusolver.so.10
2021-10-07 23:07:28.897221: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcusparse.so.10
2021-10-07 23:07:28.897231: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudnn.so.7
2021-10-07 23:07:28.898158: E
```

```
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.899312: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:28.899742: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu
devices: 0
2021-10-07 23:07:28.900428: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudart.so.10.1
2021-10-07 23:07:30.673375: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect
StreamExecutor with strength 1 edge matrix:
2021-10-07 23:07:30.673401: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]      0
2021-10-07 23:07:30.673407: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1280] 0:   N
2021-10-07 23:07:30.675516: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:30.675875: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1489] Could not identify NUMA
node of platform GPU id 0, defaulting to 0.  Your kernel may not have been built
with NUMA support.
2021-10-07 23:07:30.676731: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:30.677718: E
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:927] could not open file to
read NUMA node: /sys/bus/pci/devices/0000:01:00.0/numa_node
Your kernel may have been built without NUMA support.
2021-10-07 23:07:30.678051: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1406] Created TensorFlow device
(/job:localhost/replica:0/task:0/device:GPU:0 with 6575 MB memory) -> physical
GPU (device: 0, name: NVIDIA GeForce RTX 2080 with Max-Q Design, pci bus id:
0000:01:00.0, compute capability: 7.5)
2021-10-07 23:07:30.681952: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not
creating XLA devices, tf_xla_enable_xla_devices not set

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
```

```
conv2d (Conv2D)              (None, 148, 148, 64)      640
_________________________________________________________________
batch_normalization (BatchNo (None, 148, 148, 64)      256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 72, 72, 64)        36928
_________________________________________________________________
batch_normalization_1 (Batch (None, 72, 72, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 34, 34, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
batch_normalization_3 (Batch (None, 15, 15, 128)       512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dropout (Dropout)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 512)               3211776
_________________________________________________________________
dense_1 (Dense)              (None, 64)                32832
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 195
=================================================================
Total params: 3,505,347
Trainable params: 3,504,579
Non-trainable params: 768
_________________________________________________________________
```

[ ]:
```
from keras import optimizers

# opt = RMSprop(lr=0.0001, decay=1e-6)
opt = optimizers.Adam(learning_rate=1e-5, decay=1e-7)

model.compile(loss='categorical_crossentropy', optimizer=opt,␣
 ↪metrics=['accuracy'])
```

[ ]:
```
import numpy as np
from sklearn.utils import class_weight
from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint

classes = train_generator.classes
class_weights = class_weight.compute_class_weight(None,
                                                  np.unique(classes),
                                                  classes)
best_model_path = os.path.join(BASE_PATH, 'best_model.h5')
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)
mc = ModelCheckpoint(best_model_path, monitor='val_accuracy', mode='max',␣
 ↪verbose=1, save_best_only=True)
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // 32,
    epochs=150,
    validation_data=validation_generator,
    class_weight=dict(zip(np.unique(classes), class_weights)),
    callbacks=[es, mc]
)
```

```
/home/hivini/anaconda3/envs/tf-gpu/lib/python3.9/site-
packages/sklearn/utils/validation.py:67: FutureWarning: Pass classes=[0 1 2],
y=[0 0 0 … 2 2 2] as keyword args. From version 0.25 passing these as
positional arguments will result in an error
  warnings.warn("Pass {} as keyword args. From version 0.25 "
2021-10-07 23:07:31.660305: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2021-10-07 23:07:31.662515: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2208005000 Hz

Epoch 1/150

2021-10-07 23:07:32.451009: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcublas.so.10
2021-10-07 23:07:32.969887: I
tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully
opened dynamic library libcudnn.so.7
2021-10-07 23:07:35.163484: W tensorflow/stream_executor/gpu/asm_compiler.cc:63]
Running ptxas --version returned 256
2021-10-07 23:07:35.294879: W
tensorflow/stream_executor/gpu/redzone_allocator.cc:314] Internal: ptxas exited
with non-zero error code 256, output:
Relying on driver to perform ptx compilation.
```

```
Modify $PATH to customize ptxas location.
This message will be only logged once.

331/331 [==============================] - 86s 200ms/step - loss: 1.0255 -
accuracy: 0.6052 - val_loss: 1.9275 - val_accuracy: 0.2389

Epoch 00001: val_accuracy improved from -inf to 0.23889, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 2/150
331/331 [==============================] - 32s 98ms/step - loss: 0.6338 -
accuracy: 0.7207 - val_loss: 0.4280 - val_accuracy: 0.8196

Epoch 00002: val_accuracy improved from 0.23889 to 0.81962, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 3/150
331/331 [==============================] - 32s 97ms/step - loss: 0.5397 -
accuracy: 0.7630 - val_loss: 0.4066 - val_accuracy: 0.8245

Epoch 00003: val_accuracy improved from 0.81962 to 0.82446, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 4/150
331/331 [==============================] - 32s 97ms/step - loss: 0.5061 -
accuracy: 0.7832 - val_loss: 0.3740 - val_accuracy: 0.8412

Epoch 00004: val_accuracy improved from 0.82446 to 0.84118, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 5/150
331/331 [==============================] - 32s 97ms/step - loss: 0.4667 -
accuracy: 0.7959 - val_loss: 0.3426 - val_accuracy: 0.8601

Epoch 00005: val_accuracy improved from 0.84118 to 0.86010, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 6/150
331/331 [==============================] - 32s 98ms/step - loss: 0.4236 -
accuracy: 0.8133 - val_loss: 0.3115 - val_accuracy: 0.8715

Epoch 00006: val_accuracy improved from 0.86010 to 0.87154, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 7/150
331/331 [==============================] - 33s 100ms/step - loss: 0.4120 -
accuracy: 0.8213 - val_loss: 0.2977 - val_accuracy: 0.8759

Epoch 00007: val_accuracy improved from 0.87154 to 0.87593, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 8/150
331/331 [==============================] - 33s 100ms/step - loss: 0.3773 -
accuracy: 0.8411 - val_loss: 0.2642 - val_accuracy: 0.8927
```

```
Epoch 00008: val_accuracy improved from 0.87593 to 0.89265, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 9/150
331/331 [==============================] - 33s 100ms/step - loss: 0.3707 -
accuracy: 0.8509 - val_loss: 0.2773 - val_accuracy: 0.8847

Epoch 00009: val_accuracy did not improve from 0.89265
Epoch 10/150
331/331 [==============================] - 33s 100ms/step - loss: 0.3359 -
accuracy: 0.8618 - val_loss: 0.2780 - val_accuracy: 0.8847

Epoch 00010: val_accuracy did not improve from 0.89265
Epoch 11/150
331/331 [==============================] - 33s 100ms/step - loss: 0.3366 -
accuracy: 0.8637 - val_loss: 0.2589 - val_accuracy: 0.8883

Epoch 00011: val_accuracy did not improve from 0.89265
Epoch 12/150
331/331 [==============================] - 33s 101ms/step - loss: 0.3137 -
accuracy: 0.8742 - val_loss: 0.2393 - val_accuracy: 0.8997

Epoch 00012: val_accuracy improved from 0.89265 to 0.89969, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 13/150
331/331 [==============================] - 33s 101ms/step - loss: 0.3063 -
accuracy: 0.8746 - val_loss: 0.2263 - val_accuracy: 0.9072

Epoch 00013: val_accuracy improved from 0.89969 to 0.90717, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 14/150
331/331 [==============================] - 34s 102ms/step - loss: 0.2996 -
accuracy: 0.8814 - val_loss: 0.2378 - val_accuracy: 0.9041

Epoch 00014: val_accuracy did not improve from 0.90717
Epoch 15/150
331/331 [==============================] - 35s 106ms/step - loss: 0.2927 -
accuracy: 0.8827 - val_loss: 0.2155 - val_accuracy: 0.9142

Epoch 00015: val_accuracy improved from 0.90717 to 0.91421, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 16/150
331/331 [==============================] - 36s 109ms/step - loss: 0.2701 -
accuracy: 0.8903 - val_loss: 0.2318 - val_accuracy: 0.8997

Epoch 00016: val_accuracy did not improve from 0.91421
Epoch 17/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2690 -
accuracy: 0.8964 - val_loss: 0.2304 - val_accuracy: 0.9037
```

```
Epoch 00017: val_accuracy did not improve from 0.91421
Epoch 18/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2742 -
accuracy: 0.8893 - val_loss: 0.2675 - val_accuracy: 0.8883

Epoch 00018: val_accuracy did not improve from 0.91421
Epoch 19/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2678 -
accuracy: 0.8918 - val_loss: 0.1833 - val_accuracy: 0.9234

Epoch 00019: val_accuracy improved from 0.91421 to 0.92345, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 20/150
331/331 [==============================] - 33s 101ms/step - loss: 0.2544 -
accuracy: 0.8967 - val_loss: 0.2196 - val_accuracy: 0.9085

Epoch 00020: val_accuracy did not improve from 0.92345
Epoch 21/150
331/331 [==============================] - 33s 101ms/step - loss: 0.2517 -
accuracy: 0.8976 - val_loss: 0.1889 - val_accuracy: 0.9256

Epoch 00021: val_accuracy improved from 0.92345 to 0.92565, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 22/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2359 -
accuracy: 0.9094 - val_loss: 0.2026 - val_accuracy: 0.9173

Epoch 00022: val_accuracy did not improve from 0.92565
Epoch 23/150
331/331 [==============================] - 33s 101ms/step - loss: 0.2341 -
accuracy: 0.9131 - val_loss: 0.1975 - val_accuracy: 0.9234

Epoch 00023: val_accuracy did not improve from 0.92565
Epoch 24/150
331/331 [==============================] - 34s 101ms/step - loss: 0.2318 -
accuracy: 0.9109 - val_loss: 0.1721 - val_accuracy: 0.9309

Epoch 00024: val_accuracy improved from 0.92565 to 0.93093, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 25/150
331/331 [==============================] - 33s 101ms/step - loss: 0.2201 -
accuracy: 0.9112 - val_loss: 0.2292 - val_accuracy: 0.9107

Epoch 00025: val_accuracy did not improve from 0.93093
Epoch 26/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2262 -
accuracy: 0.9157 - val_loss: 0.1595 - val_accuracy: 0.9358
```

```
Epoch 00026: val_accuracy improved from 0.93093 to 0.93577, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 27/150
331/331 [==============================] - 33s 100ms/step - loss: 0.2169 -
accuracy: 0.9155 - val_loss: 0.1712 - val_accuracy: 0.9322

Epoch 00027: val_accuracy did not improve from 0.93577
Epoch 28/150
331/331 [==============================] - 34s 101ms/step - loss: 0.2185 -
accuracy: 0.9151 - val_loss: 0.1669 - val_accuracy: 0.9380

Epoch 00028: val_accuracy improved from 0.93577 to 0.93797, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 29/150
331/331 [==============================] - 34s 101ms/step - loss: 0.2051 -
accuracy: 0.9181 - val_loss: 0.1617 - val_accuracy: 0.9393

Epoch 00029: val_accuracy improved from 0.93797 to 0.93929, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 30/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1938 -
accuracy: 0.9247 - val_loss: 0.1571 - val_accuracy: 0.9349

Epoch 00030: val_accuracy did not improve from 0.93929
Epoch 31/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1998 -
accuracy: 0.9205 - val_loss: 0.1782 - val_accuracy: 0.9287

Epoch 00031: val_accuracy did not improve from 0.93929
Epoch 32/150
331/331 [==============================] - 33s 101ms/step - loss: 0.2039 -
accuracy: 0.9233 - val_loss: 0.1520 - val_accuracy: 0.9406

Epoch 00032: val_accuracy improved from 0.93929 to 0.94061, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 33/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1956 -
accuracy: 0.9200 - val_loss: 0.1519 - val_accuracy: 0.9362

Epoch 00033: val_accuracy did not improve from 0.94061
Epoch 34/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1914 -
accuracy: 0.9246 - val_loss: 0.1860 - val_accuracy: 0.9296

Epoch 00034: val_accuracy did not improve from 0.94061
Epoch 35/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1840 -
```

```
accuracy: 0.9289 - val_loss: 0.1591 - val_accuracy: 0.9402

Epoch 00035: val_accuracy did not improve from 0.94061
Epoch 36/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1734 -
accuracy: 0.9332 - val_loss: 0.1579 - val_accuracy: 0.9419

Epoch 00036: val_accuracy improved from 0.94061 to 0.94193, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 37/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1792 -
accuracy: 0.9319 - val_loss: 0.1439 - val_accuracy: 0.9424

Epoch 00037: val_accuracy improved from 0.94193 to 0.94237, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 38/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1699 -
accuracy: 0.9369 - val_loss: 0.1482 - val_accuracy: 0.9450

Epoch 00038: val_accuracy improved from 0.94237 to 0.94501, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 39/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1735 -
accuracy: 0.9354 - val_loss: 0.2238 - val_accuracy: 0.9138

Epoch 00039: val_accuracy did not improve from 0.94501
Epoch 40/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1864 -
accuracy: 0.9284 - val_loss: 0.1533 - val_accuracy: 0.9406

Epoch 00040: val_accuracy did not improve from 0.94501
Epoch 41/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1668 -
accuracy: 0.9400 - val_loss: 0.1546 - val_accuracy: 0.9441

Epoch 00041: val_accuracy did not improve from 0.94501
Epoch 42/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1654 -
accuracy: 0.9402 - val_loss: 0.2114 - val_accuracy: 0.9212

Epoch 00042: val_accuracy did not improve from 0.94501
Epoch 43/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1644 -
accuracy: 0.9378 - val_loss: 0.1511 - val_accuracy: 0.9384

Epoch 00043: val_accuracy did not improve from 0.94501
Epoch 44/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1757 -
```

```
accuracy: 0.9347 - val_loss: 0.1332 - val_accuracy: 0.9490

Epoch 00044: val_accuracy improved from 0.94501 to 0.94897, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 45/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1612 -
accuracy: 0.9439 - val_loss: 0.1533 - val_accuracy: 0.9432

Epoch 00045: val_accuracy did not improve from 0.94897
Epoch 46/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1590 -
accuracy: 0.9398 - val_loss: 0.1450 - val_accuracy: 0.9419

Epoch 00046: val_accuracy did not improve from 0.94897
Epoch 47/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1439 -
accuracy: 0.9500 - val_loss: 0.1610 - val_accuracy: 0.9380

Epoch 00047: val_accuracy did not improve from 0.94897
Epoch 48/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1484 -
accuracy: 0.9473 - val_loss: 0.1255 - val_accuracy: 0.9516

Epoch 00048: val_accuracy improved from 0.94897 to 0.95161, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 49/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1530 -
accuracy: 0.9434 - val_loss: 0.1383 - val_accuracy: 0.9485

Epoch 00049: val_accuracy did not improve from 0.95161
Epoch 50/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1469 -
accuracy: 0.9450 - val_loss: 0.1298 - val_accuracy: 0.9476

Epoch 00050: val_accuracy did not improve from 0.95161
Epoch 51/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1405 -
accuracy: 0.9463 - val_loss: 0.1594 - val_accuracy: 0.9393

Epoch 00051: val_accuracy did not improve from 0.95161
Epoch 52/150
331/331 [==============================] - 32s 97ms/step - loss: 0.1460 -
accuracy: 0.9485 - val_loss: 0.1443 - val_accuracy: 0.9481

Epoch 00052: val_accuracy did not improve from 0.95161
Epoch 53/150
331/331 [==============================] - 32s 97ms/step - loss: 0.1426 -
accuracy: 0.9475 - val_loss: 0.1195 - val_accuracy: 0.9569
```

```
Epoch 00053: val_accuracy improved from 0.95161 to 0.95689, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 54/150
331/331 [==============================] - 32s 98ms/step - loss: 0.1472 -
accuracy: 0.9427 - val_loss: 0.1168 - val_accuracy: 0.9556

Epoch 00054: val_accuracy did not improve from 0.95689
Epoch 55/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1288 -
accuracy: 0.9546 - val_loss: 0.1139 - val_accuracy: 0.9569

Epoch 00055: val_accuracy did not improve from 0.95689
Epoch 56/150
331/331 [==============================] - 34s 103ms/step - loss: 0.1376 -
accuracy: 0.9489 - val_loss: 0.1210 - val_accuracy: 0.9534

Epoch 00056: val_accuracy did not improve from 0.95689
Epoch 57/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1308 -
accuracy: 0.9527 - val_loss: 0.1193 - val_accuracy: 0.9551

Epoch 00057: val_accuracy did not improve from 0.95689
Epoch 58/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1237 -
accuracy: 0.9536 - val_loss: 0.2349 - val_accuracy: 0.9173

Epoch 00058: val_accuracy did not improve from 0.95689
Epoch 59/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1331 -
accuracy: 0.9481 - val_loss: 0.1265 - val_accuracy: 0.9490

Epoch 00059: val_accuracy did not improve from 0.95689
Epoch 60/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1314 -
accuracy: 0.9489 - val_loss: 0.1598 - val_accuracy: 0.9406

Epoch 00060: val_accuracy did not improve from 0.95689
Epoch 61/150
331/331 [==============================] - 33s 100ms/step - loss: 0.1291 -
accuracy: 0.9513 - val_loss: 0.1068 - val_accuracy: 0.9591

Epoch 00061: val_accuracy improved from 0.95689 to 0.95908, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 62/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1292 -
accuracy: 0.9542 - val_loss: 0.1414 - val_accuracy: 0.9454
```

```
Epoch 00062: val_accuracy did not improve from 0.95908
Epoch 63/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1273 -
accuracy: 0.9546 - val_loss: 0.1711 - val_accuracy: 0.9380

Epoch 00063: val_accuracy did not improve from 0.95908
Epoch 64/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1345 -
accuracy: 0.9513 - val_loss: 0.1099 - val_accuracy: 0.9551

Epoch 00064: val_accuracy did not improve from 0.95908
Epoch 65/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1276 -
accuracy: 0.9509 - val_loss: 0.1301 - val_accuracy: 0.9525

Epoch 00065: val_accuracy did not improve from 0.95908
Epoch 66/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1224 -
accuracy: 0.9526 - val_loss: 0.1289 - val_accuracy: 0.9494

Epoch 00066: val_accuracy did not improve from 0.95908
Epoch 67/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1196 -
accuracy: 0.9558 - val_loss: 0.1364 - val_accuracy: 0.9520

Epoch 00067: val_accuracy did not improve from 0.95908
Epoch 68/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1141 -
accuracy: 0.9583 - val_loss: 0.1098 - val_accuracy: 0.9573

Epoch 00068: val_accuracy did not improve from 0.95908
Epoch 69/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1257 -
accuracy: 0.9561 - val_loss: 0.0971 - val_accuracy: 0.9670

Epoch 00069: val_accuracy improved from 0.95908 to 0.96700, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 70/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1262 -
accuracy: 0.9519 - val_loss: 0.1164 - val_accuracy: 0.9547

Epoch 00070: val_accuracy did not improve from 0.96700
Epoch 71/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1278 -
accuracy: 0.9568 - val_loss: 0.1377 - val_accuracy: 0.9494

Epoch 00071: val_accuracy did not improve from 0.96700
Epoch 72/150
```

```
331/331 [==============================] - 34s 102ms/step - loss: 0.1048 -
accuracy: 0.9612 - val_loss: 0.1377 - val_accuracy: 0.9485

Epoch 00072: val_accuracy did not improve from 0.96700
Epoch 73/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1095 -
accuracy: 0.9566 - val_loss: 0.1178 - val_accuracy: 0.9573

Epoch 00073: val_accuracy did not improve from 0.96700
Epoch 74/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1172 -
accuracy: 0.9554 - val_loss: 0.1238 - val_accuracy: 0.9534

Epoch 00074: val_accuracy did not improve from 0.96700
Epoch 75/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1162 -
accuracy: 0.9564 - val_loss: 0.1235 - val_accuracy: 0.9534

Epoch 00075: val_accuracy did not improve from 0.96700
Epoch 76/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1082 -
accuracy: 0.9594 - val_loss: 0.0999 - val_accuracy: 0.9626

Epoch 00076: val_accuracy did not improve from 0.96700
Epoch 77/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1177 -
accuracy: 0.9576 - val_loss: 0.1245 - val_accuracy: 0.9551

Epoch 00077: val_accuracy did not improve from 0.96700
Epoch 78/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1145 -
accuracy: 0.9568 - val_loss: 0.1269 - val_accuracy: 0.9516

Epoch 00078: val_accuracy did not improve from 0.96700
Epoch 79/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1131 -
accuracy: 0.9601 - val_loss: 0.1149 - val_accuracy: 0.9542

Epoch 00079: val_accuracy did not improve from 0.96700
Epoch 80/150
331/331 [==============================] - 34s 102ms/step - loss: 0.1049 -
accuracy: 0.9603 - val_loss: 0.1155 - val_accuracy: 0.9560

Epoch 00080: val_accuracy did not improve from 0.96700
Epoch 81/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1081 -
accuracy: 0.9627 - val_loss: 0.1351 - val_accuracy: 0.9516
```

```
Epoch 00081: val_accuracy did not improve from 0.96700
Epoch 82/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1092 -
accuracy: 0.9591 - val_loss: 0.1091 - val_accuracy: 0.9586

Epoch 00082: val_accuracy did not improve from 0.96700
Epoch 83/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1061 -
accuracy: 0.9595 - val_loss: 0.1341 - val_accuracy: 0.9551

Epoch 00083: val_accuracy did not improve from 0.96700
Epoch 84/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1080 -
accuracy: 0.9590 - val_loss: 0.1094 - val_accuracy: 0.9591

Epoch 00084: val_accuracy did not improve from 0.96700
Epoch 85/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0992 -
accuracy: 0.9637 - val_loss: 0.1436 - val_accuracy: 0.9459

Epoch 00085: val_accuracy did not improve from 0.96700
Epoch 86/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1053 -
accuracy: 0.9625 - val_loss: 0.1006 - val_accuracy: 0.9622

Epoch 00086: val_accuracy did not improve from 0.96700
Epoch 87/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0985 -
accuracy: 0.9630 - val_loss: 0.1466 - val_accuracy: 0.9494

Epoch 00087: val_accuracy did not improve from 0.96700
Epoch 88/150
331/331 [==============================] - 34s 101ms/step - loss: 0.1015 -
accuracy: 0.9611 - val_loss: 0.0953 - val_accuracy: 0.9644

Epoch 00088: val_accuracy did not improve from 0.96700
Epoch 89/150
331/331 [==============================] - 33s 101ms/step - loss: 0.1073 -
accuracy: 0.9627 - val_loss: 0.1014 - val_accuracy: 0.9595

Epoch 00089: val_accuracy did not improve from 0.96700
Epoch 90/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0975 -
accuracy: 0.9638 - val_loss: 0.1020 - val_accuracy: 0.9639

Epoch 00090: val_accuracy did not improve from 0.96700
Epoch 91/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0929 -
```

```
accuracy: 0.9665 - val_loss: 0.1594 - val_accuracy: 0.9393

Epoch 00091: val_accuracy did not improve from 0.96700
Epoch 92/150
331/331 [==============================] - 36s 108ms/step - loss: 0.1061 -
accuracy: 0.9616 - val_loss: 0.1715 - val_accuracy: 0.9419

Epoch 00092: val_accuracy did not improve from 0.96700
Epoch 93/150
331/331 [==============================] - 36s 110ms/step - loss: 0.0997 -
accuracy: 0.9630 - val_loss: 0.0968 - val_accuracy: 0.9630

Epoch 00093: val_accuracy did not improve from 0.96700
Epoch 94/150
331/331 [==============================] - 36s 108ms/step - loss: 0.0930 -
accuracy: 0.9686 - val_loss: 0.1194 - val_accuracy: 0.9534

Epoch 00094: val_accuracy did not improve from 0.96700
Epoch 95/150
331/331 [==============================] - 35s 107ms/step - loss: 0.0908 -
accuracy: 0.9663 - val_loss: 0.1113 - val_accuracy: 0.9591

Epoch 00095: val_accuracy did not improve from 0.96700
Epoch 96/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0986 -
accuracy: 0.9640 - val_loss: 0.2489 - val_accuracy: 0.9160

Epoch 00096: val_accuracy did not improve from 0.96700
Epoch 97/150
331/331 [==============================] - 34s 103ms/step - loss: 0.0860 -
accuracy: 0.9708 - val_loss: 0.1737 - val_accuracy: 0.9375

Epoch 00097: val_accuracy did not improve from 0.96700
Epoch 98/150
331/331 [==============================] - 36s 110ms/step - loss: 0.0937 -
accuracy: 0.9649 - val_loss: 0.1326 - val_accuracy: 0.9516

Epoch 00098: val_accuracy did not improve from 0.96700
Epoch 99/150
331/331 [==============================] - 35s 106ms/step - loss: 0.0928 -
accuracy: 0.9664 - val_loss: 0.1795 - val_accuracy: 0.9371

Epoch 00099: val_accuracy did not improve from 0.96700
Epoch 100/150
331/331 [==============================] - 33s 100ms/step - loss: 0.0861 -
accuracy: 0.9669 - val_loss: 0.1133 - val_accuracy: 0.9582

Epoch 00100: val_accuracy did not improve from 0.96700
```

```
Epoch 101/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0874 -
accuracy: 0.9695 - val_loss: 0.1043 - val_accuracy: 0.9622

Epoch 00101: val_accuracy did not improve from 0.96700
Epoch 102/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0891 -
accuracy: 0.9694 - val_loss: 0.1033 - val_accuracy: 0.9622

Epoch 00102: val_accuracy did not improve from 0.96700
Epoch 103/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0867 -
accuracy: 0.9659 - val_loss: 0.1147 - val_accuracy: 0.9591

Epoch 00103: val_accuracy did not improve from 0.96700
Epoch 104/150
331/331 [==============================] - 35s 105ms/step - loss: 0.0798 -
accuracy: 0.9701 - val_loss: 0.1257 - val_accuracy: 0.9556

Epoch 00104: val_accuracy did not improve from 0.96700
Epoch 105/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0770 -
accuracy: 0.9704 - val_loss: 0.1098 - val_accuracy: 0.9604

Epoch 00105: val_accuracy did not improve from 0.96700
Epoch 106/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0935 -
accuracy: 0.9673 - val_loss: 0.0911 - val_accuracy: 0.9657

Epoch 00106: val_accuracy did not improve from 0.96700
Epoch 107/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0871 -
accuracy: 0.9704 - val_loss: 0.2003 - val_accuracy: 0.9314

Epoch 00107: val_accuracy did not improve from 0.96700
Epoch 108/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0770 -
accuracy: 0.9718 - val_loss: 0.1037 - val_accuracy: 0.9600

Epoch 00108: val_accuracy did not improve from 0.96700
Epoch 109/150
331/331 [==============================] - 34s 103ms/step - loss: 0.0813 -
accuracy: 0.9702 - val_loss: 0.1549 - val_accuracy: 0.9468

Epoch 00109: val_accuracy did not improve from 0.96700
Epoch 110/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0804 -
accuracy: 0.9714 - val_loss: 0.1163 - val_accuracy: 0.9586
```

```
Epoch 00110: val_accuracy did not improve from 0.96700
Epoch 111/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0871 -
accuracy: 0.9682 - val_loss: 0.2347 - val_accuracy: 0.9261

Epoch 00111: val_accuracy did not improve from 0.96700
Epoch 112/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0834 -
accuracy: 0.9691 - val_loss: 0.1228 - val_accuracy: 0.9516

Epoch 00112: val_accuracy did not improve from 0.96700
Epoch 113/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0816 -
accuracy: 0.9692 - val_loss: 0.1161 - val_accuracy: 0.9569

Epoch 00113: val_accuracy did not improve from 0.96700
Epoch 114/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0787 -
accuracy: 0.9724 - val_loss: 0.1227 - val_accuracy: 0.9578

Epoch 00114: val_accuracy did not improve from 0.96700
Epoch 115/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0771 -
accuracy: 0.9719 - val_loss: 0.1330 - val_accuracy: 0.9538

Epoch 00115: val_accuracy did not improve from 0.96700
Epoch 116/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0770 -
accuracy: 0.9693 - val_loss: 0.0965 - val_accuracy: 0.9648

Epoch 00116: val_accuracy did not improve from 0.96700
Epoch 117/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0830 -
accuracy: 0.9700 - val_loss: 0.0926 - val_accuracy: 0.9661

Epoch 00117: val_accuracy did not improve from 0.96700
Epoch 118/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0908 -
accuracy: 0.9688 - val_loss: 0.1577 - val_accuracy: 0.9441

Epoch 00118: val_accuracy did not improve from 0.96700
Epoch 119/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0716 -
accuracy: 0.9746 - val_loss: 0.1597 - val_accuracy: 0.9454

Epoch 00119: val_accuracy did not improve from 0.96700
Epoch 120/150
```

```
331/331 [==============================] - 34s 101ms/step - loss: 0.0760 -
accuracy: 0.9720 - val_loss: 0.2751 - val_accuracy: 0.9164

Epoch 00120: val_accuracy did not improve from 0.96700
Epoch 121/150
331/331 [==============================] - 35s 104ms/step - loss: 0.0797 -
accuracy: 0.9705 - val_loss: 0.0990 - val_accuracy: 0.9639

Epoch 00121: val_accuracy did not improve from 0.96700
Epoch 122/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0878 -
accuracy: 0.9678 - val_loss: 0.1162 - val_accuracy: 0.9591

Epoch 00122: val_accuracy did not improve from 0.96700
Epoch 123/150
331/331 [==============================] - 34s 104ms/step - loss: 0.0726 -
accuracy: 0.9729 - val_loss: 0.1006 - val_accuracy: 0.9617

Epoch 00123: val_accuracy did not improve from 0.96700
Epoch 124/150
331/331 [==============================] - 34s 103ms/step - loss: 0.0740 -
accuracy: 0.9733 - val_loss: 0.1382 - val_accuracy: 0.9512

Epoch 00124: val_accuracy did not improve from 0.96700
Epoch 125/150
331/331 [==============================] - 34s 103ms/step - loss: 0.0693 -
accuracy: 0.9738 - val_loss: 0.0838 - val_accuracy: 0.9688

Epoch 00125: val_accuracy improved from 0.96700 to 0.96876, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 126/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0726 -
accuracy: 0.9729 - val_loss: 0.1036 - val_accuracy: 0.9591

Epoch 00126: val_accuracy did not improve from 0.96876
Epoch 127/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0689 -
accuracy: 0.9751 - val_loss: 0.1578 - val_accuracy: 0.9463

Epoch 00127: val_accuracy did not improve from 0.96876
Epoch 128/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0686 -
accuracy: 0.9761 - val_loss: 0.1350 - val_accuracy: 0.9529

Epoch 00128: val_accuracy did not improve from 0.96876
Epoch 129/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0675 -
accuracy: 0.9761 - val_loss: 0.1101 - val_accuracy: 0.9604
```

```
Epoch 00129: val_accuracy did not improve from 0.96876
Epoch 130/150
331/331 [==============================] - 35s 104ms/step - loss: 0.0687 -
accuracy: 0.9740 - val_loss: 0.1276 - val_accuracy: 0.9560

Epoch 00130: val_accuracy did not improve from 0.96876
Epoch 131/150
331/331 [==============================] - 34s 101ms/step - loss: 0.0681 -
accuracy: 0.9741 - val_loss: 0.1695 - val_accuracy: 0.9397

Epoch 00131: val_accuracy did not improve from 0.96876
Epoch 132/150
331/331 [==============================] - 34s 102ms/step - loss: 0.0652 -
accuracy: 0.9778 - val_loss: 0.1702 - val_accuracy: 0.9406

Epoch 00132: val_accuracy did not improve from 0.96876
Epoch 133/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0756 -
accuracy: 0.9718 - val_loss: 0.1204 - val_accuracy: 0.9556

Epoch 00133: val_accuracy did not improve from 0.96876
Epoch 134/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0708 -
accuracy: 0.9737 - val_loss: 0.1219 - val_accuracy: 0.9578

Epoch 00134: val_accuracy did not improve from 0.96876
Epoch 135/150
331/331 [==============================] - 33s 101ms/step - loss: 0.0723 -
accuracy: 0.9730 - val_loss: 0.1682 - val_accuracy: 0.9437

Epoch 00135: val_accuracy did not improve from 0.96876
Epoch 136/150
331/331 [==============================] - 33s 100ms/step - loss: 0.0708 -
accuracy: 0.9743 - val_loss: 0.2359 - val_accuracy: 0.9309

Epoch 00136: val_accuracy did not improve from 0.96876
Epoch 137/150
331/331 [==============================] - 33s 100ms/step - loss: 0.0646 -
accuracy: 0.9751 - val_loss: 0.1461 - val_accuracy: 0.9507

Epoch 00137: val_accuracy did not improve from 0.96876
Epoch 138/150
331/331 [==============================] - 35s 104ms/step - loss: 0.0681 -
accuracy: 0.9748 - val_loss: 0.1991 - val_accuracy: 0.9371

Epoch 00138: val_accuracy did not improve from 0.96876
Epoch 139/150
```

```
331/331 [==============================] - 35s 106ms/step - loss: 0.0692 -
accuracy: 0.9740 - val_loss: 0.0900 - val_accuracy: 0.9692

Epoch 00139: val_accuracy improved from 0.96876 to 0.96920, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 140/150
331/331 [==============================] - 35s 107ms/step - loss: 0.0725 -
accuracy: 0.9761 - val_loss: 0.0979 - val_accuracy: 0.9595

Epoch 00140: val_accuracy did not improve from 0.96920
Epoch 141/150
331/331 [==============================] - 34s 104ms/step - loss: 0.0731 -
accuracy: 0.9747 - val_loss: 0.1665 - val_accuracy: 0.9384

Epoch 00141: val_accuracy did not improve from 0.96920
Epoch 142/150
331/331 [==============================] - 35s 105ms/step - loss: 0.0599 -
accuracy: 0.9781 - val_loss: 0.0778 - val_accuracy: 0.9758

Epoch 00142: val_accuracy improved from 0.96920 to 0.97580, saving model to
/home/hivini/learn/research/new-covid/best_model.h5
Epoch 143/150
331/331 [==============================] - 38s 116ms/step - loss: 0.0684 -
accuracy: 0.9740 - val_loss: 0.1101 - val_accuracy: 0.9613

Epoch 00143: val_accuracy did not improve from 0.97580
Epoch 144/150
331/331 [==============================] - 33s 98ms/step - loss: 0.0716 -
accuracy: 0.9731 - val_loss: 0.1172 - val_accuracy: 0.9582

Epoch 00144: val_accuracy did not improve from 0.97580
Epoch 145/150
331/331 [==============================] - 32s 98ms/step - loss: 0.0659 -
accuracy: 0.9756 - val_loss: 0.0985 - val_accuracy: 0.9648

Epoch 00145: val_accuracy did not improve from 0.97580
Epoch 146/150
331/331 [==============================] - 32s 98ms/step - loss: 0.0678 -
accuracy: 0.9750 - val_loss: 0.1296 - val_accuracy: 0.9569

Epoch 00146: val_accuracy did not improve from 0.97580
Epoch 147/150
331/331 [==============================] - 32s 98ms/step - loss: 0.0643 -
accuracy: 0.9774 - val_loss: 0.1282 - val_accuracy: 0.9569

Epoch 00147: val_accuracy did not improve from 0.97580
Epoch 148/150
331/331 [==============================] - 33s 100ms/step - loss: 0.0662 -
```

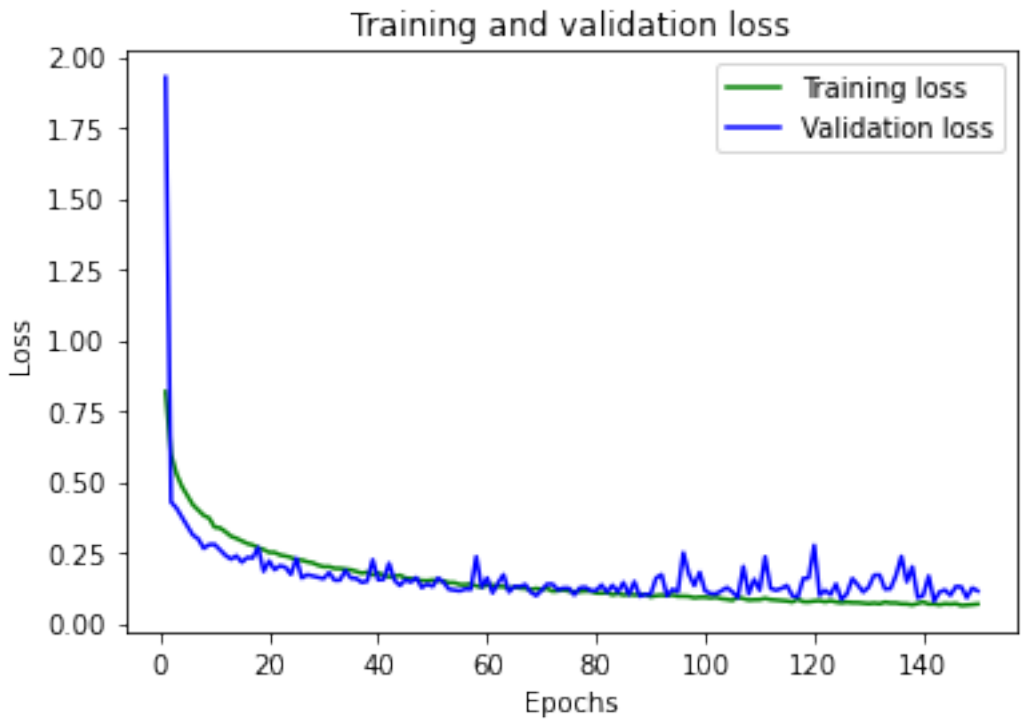
```
accuracy: 0.9762 - val_loss: 0.0898 - val_accuracy: 0.9683

Epoch 00148: val_accuracy did not improve from 0.97580
Epoch 149/150
331/331 [==============================] - 32s 98ms/step - loss: 0.0652 -
accuracy: 0.9760 - val_loss: 0.1243 - val_accuracy: 0.9569

Epoch 00149: val_accuracy did not improve from 0.97580
Epoch 150/150
331/331 [==============================] - 32s 98ms/step - loss: 0.0606 -
accuracy: 0.9770 - val_loss: 0.1138 - val_accuracy: 0.9617

Epoch 00150: val_accuracy did not improve from 0.97580
```

[ ]:
```
model.save(os.path.join(BASE_PATH, 'covid_classifier_result.h5'))
```

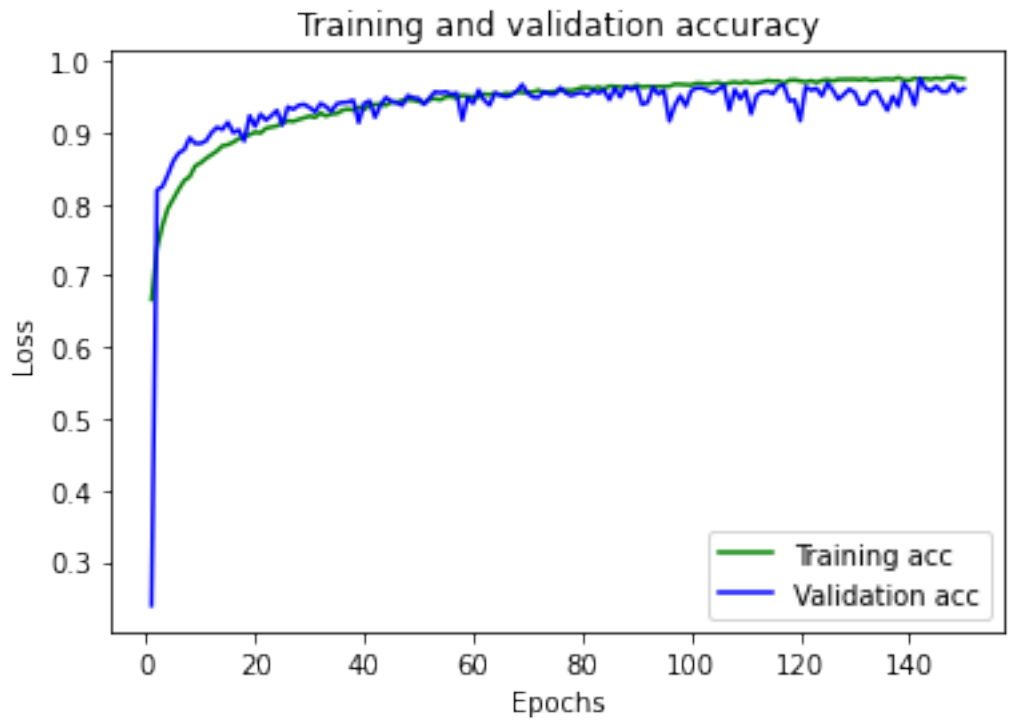
[ ]:
```
test_loss, test_acc = model.evaluate(test_generator)
print("Loss on test set: ", test_loss)
print("Accuracy on test set: ", test_acc)
```

```
72/72 [==============================] - 13s 180ms/step - loss: 0.1171 -
accuracy: 0.9565
Loss on test set:  0.11710795760154724
Accuracy on test set:  0.9564643502235413
```

[ ]:
```
import matplotlib.pyplot as plt

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(acc) + 1)
# bo is for blue dot.
plt.plot(epochs, loss, 'g', label='Training loss')
# b is for solid blue line
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

plt.plot(epochs, acc, 'g', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



[ ]:
```
from sklearn.metrics import classification_report, confusion_matrix

X_test = []
Y_test = []
# Extract the data
for X, Y in test_generator:
    X_test.append(X)
    Y_test.append(Y)
X_test = np.array(X_test)
Y_test = np.array(Y_test)

predictions = model.predict_classes(X_test)
predictions = predictions.reshape(1, -1)[0]

print(classification_report(Y_test, predictions, target_names=[
      'Covid (Class 0)', 'Normal (Class 1)', 'Viral Pneumonia (Class 2)']))
```

[ ]:
```
import pandas as pd
import seaborn as sns

labels = ['covid', 'normal', 'viral_pneumonia']

cm = confusion_matrix(Y_test,predictions)
```

```
cm = pd.DataFrame(cm , index = ['0','1'] , columns = ['0','1'])
plt.figure(figsize = (10,10))
sns.heatmap(cm,cmap= "Reds", linecolor = 'black' , linewidth = 1 , annot =␣
 ↪True, fmt='',xticklabels = labels,yticklabels = labels)
```