covid_classifier_4_150

September 30, 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.

[ ]:
```
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)
```

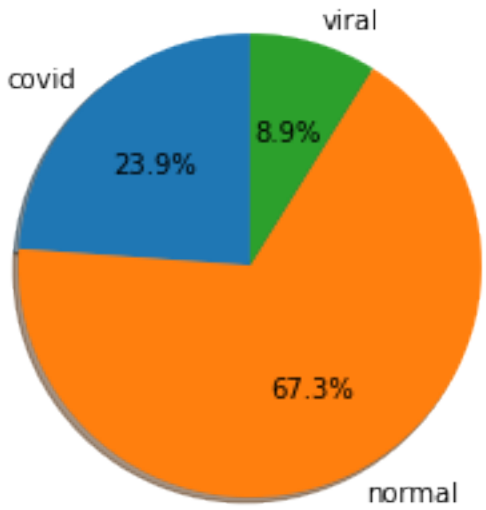
```
createDir(VALIDATION_DIR)
createDir(TEST_DIR)
createDir(TRAIN_VIRAL_DIR)
createDir(TRAIN_COVID_DIR)
createDir(TRAIN_NORMAL_DIR)
createDir(VALIDATION_VIRAL_DIR)
createDir(VALIDATION_COVID_DIR)
createDir(VALIDATION_NORMAL_DIR)
createDir(TEST_VIRAL_DIR)
createDir(TEST_COVID_DIR)
createDir(TEST_NORMAL_DIR)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

[ ]:
```
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 = True
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()
```



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 import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(150, 150,␣
↪1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))
model.summary()
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_32 (Conv2D)           (None, 148, 148, 64)      640
_________________________________________________________________
max_pooling2d_32 (MaxPooling (None, 74, 74, 64)        0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 72, 72, 64)        36928
_________________________________________________________________
max_pooling2d_33 (MaxPooling (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_34 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_35 (MaxPooling (None, 7, 7, 128)         0
_________________________________________________________________
flatten_8 (Flatten)          (None, 6272)              0
_________________________________________________________________
dropout_8 (Dropout)          (None, 6272)              0
_________________________________________________________________
dense_26 (Dense)             (None, 512)               3211776
```

```
_________________________________________________________________
dense_27 (Dense)             (None, 256)               131328
_________________________________________________________________
dense_28 (Dense)             (None, 64)                16448
_________________________________________________________________
dense_29 (Dense)             (None, 3)                 195
=================================================================
Total params: 3,618,755
Trainable params: 3,618,755
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
from keras import optimizers

model.compile(loss='categorical_crossentropy', optimizer=optimizers.
 ↪RMSprop(learning_rate=1e-4), metrics=['accuracy'])
```

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

train_datagen = ImageDataGenerator(
    rescale=1./255,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3
)

# 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'
)

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

test_generator = evaluate_datagen.flow_from_directory(
```

```
    TEST_DIR,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    color_mode='grayscale'
)
```

```
Found 10606 images belonging to 3 classes.
Found 2273 images belonging to 3 classes.
Found 2274 images belonging to 3 classes.
```

[ ]:
```
import numpy as np
from sklearn.utils import class_weight

classes = train_generator.classes
class_weights = class_weight.compute_class_weight(None,
                                                  np.unique(classes),
                                                  classes)
history = model.fit(
    train_generator,
    steps_per_epoch=100,
    epochs=150,
    validation_data=validation_generator,
    validation_steps=50,
    class_weight=dict(zip(np.unique(classes), class_weights))
)
```

```
Epoch 1/150
100/100 [==============================] - 12s 114ms/step - loss: 0.8318 -
accuracy: 0.6770 - val_loss: 0.6814 - val_accuracy: 0.6888
Epoch 2/150
100/100 [==============================] - 11s 111ms/step - loss: 0.7167 -
accuracy: 0.6745 - val_loss: 0.5848 - val_accuracy: 0.7262
Epoch 3/150
100/100 [==============================] - 11s 111ms/step - loss: 0.6480 -
accuracy: 0.6919 - val_loss: 0.5649 - val_accuracy: 0.7244
Epoch 4/150
100/100 [==============================] - 11s 112ms/step - loss: 0.5920 -
accuracy: 0.7204 - val_loss: 0.5072 - val_accuracy: 0.7775
Epoch 5/150
100/100 [==============================] - 12s 122ms/step - loss: 0.5650 -
accuracy: 0.7474 - val_loss: 0.5219 - val_accuracy: 0.7719
Epoch 6/150
100/100 [==============================] - 12s 124ms/step - loss: 0.5897 -
accuracy: 0.7139 - val_loss: 0.4743 - val_accuracy: 0.7825
Epoch 7/150
100/100 [==============================] - 11s 110ms/step - loss: 0.5381 -
accuracy: 0.7654 - val_loss: 0.4852 - val_accuracy: 0.7669
```

```
Epoch 8/150
100/100 [==============================] - 12s 118ms/step - loss: 0.5474 -
accuracy: 0.7370 - val_loss: 0.4421 - val_accuracy: 0.8131
Epoch 9/150
100/100 [==============================] - 12s 116ms/step - loss: 0.5267 -
accuracy: 0.7629 - val_loss: 0.4295 - val_accuracy: 0.8081
Epoch 10/150
100/100 [==============================] - 11s 113ms/step - loss: 0.5037 -
accuracy: 0.7730 - val_loss: 0.4217 - val_accuracy: 0.8181
Epoch 11/150
100/100 [==============================] - 11s 110ms/step - loss: 0.4856 -
accuracy: 0.7918 - val_loss: 0.4850 - val_accuracy: 0.7663
Epoch 12/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4840 -
accuracy: 0.7946 - val_loss: 0.3887 - val_accuracy: 0.8406
Epoch 13/150
100/100 [==============================] - 11s 108ms/step - loss: 0.4743 -
accuracy: 0.7900 - val_loss: 0.4052 - val_accuracy: 0.8050
Epoch 14/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4520 -
accuracy: 0.8045 - val_loss: 0.4709 - val_accuracy: 0.7862
Epoch 15/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4531 -
accuracy: 0.7984 - val_loss: 0.4011 - val_accuracy: 0.8087
Epoch 16/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4268 -
accuracy: 0.8132 - val_loss: 0.3870 - val_accuracy: 0.8294
Epoch 17/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4217 -
accuracy: 0.8249 - val_loss: 0.3598 - val_accuracy: 0.8487
Epoch 18/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4301 -
accuracy: 0.8062 - val_loss: 0.3530 - val_accuracy: 0.8375
Epoch 19/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4149 -
accuracy: 0.8188 - val_loss: 0.3412 - val_accuracy: 0.8706
Epoch 20/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4160 -
accuracy: 0.8131 - val_loss: 0.3199 - val_accuracy: 0.8706
Epoch 21/150
100/100 [==============================] - 11s 110ms/step - loss: 0.4217 -
accuracy: 0.8178 - val_loss: 0.3205 - val_accuracy: 0.8700
Epoch 22/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4333 -
accuracy: 0.8136 - val_loss: 0.3000 - val_accuracy: 0.8775
Epoch 23/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4010 -
accuracy: 0.8279 - val_loss: 0.3239 - val_accuracy: 0.8650
```

```
Epoch 24/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3974 -
accuracy: 0.8226 - val_loss: 0.2990 - val_accuracy: 0.8750
Epoch 25/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3750 -
accuracy: 0.8435 - val_loss: 0.3062 - val_accuracy: 0.8781
Epoch 26/150
100/100 [==============================] - 11s 108ms/step - loss: 0.4073 -
accuracy: 0.8178 - val_loss: 0.2861 - val_accuracy: 0.8856
Epoch 27/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3831 -
accuracy: 0.8286 - val_loss: 0.3057 - val_accuracy: 0.8687
Epoch 28/150
100/100 [==============================] - 11s 108ms/step - loss: 0.3880 -
accuracy: 0.8349 - val_loss: 0.3027 - val_accuracy: 0.8850
Epoch 29/150
100/100 [==============================] - 11s 109ms/step - loss: 0.4033 -
accuracy: 0.8325 - val_loss: 0.2526 - val_accuracy: 0.9050
Epoch 30/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3882 -
accuracy: 0.8313 - val_loss: 0.2666 - val_accuracy: 0.8950
Epoch 31/150
100/100 [==============================] - 11s 113ms/step - loss: 0.3522 -
accuracy: 0.8545 - val_loss: 0.2848 - val_accuracy: 0.8756
Epoch 32/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3935 -
accuracy: 0.8299 - val_loss: 0.2465 - val_accuracy: 0.9000
Epoch 33/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3464 -
accuracy: 0.8556 - val_loss: 0.2625 - val_accuracy: 0.8963
Epoch 34/150
100/100 [==============================] - 11s 112ms/step - loss: 0.3637 -
accuracy: 0.8393 - val_loss: 0.2402 - val_accuracy: 0.9150
Epoch 35/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3639 -
accuracy: 0.8438 - val_loss: 0.2416 - val_accuracy: 0.9044
Epoch 36/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3559 -
accuracy: 0.8482 - val_loss: 0.2530 - val_accuracy: 0.8925
Epoch 37/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3434 -
accuracy: 0.8603 - val_loss: 0.2559 - val_accuracy: 0.9062
Epoch 38/150
100/100 [==============================] - 11s 108ms/step - loss: 0.3437 -
accuracy: 0.8516 - val_loss: 0.2485 - val_accuracy: 0.8963
Epoch 39/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3329 -
accuracy: 0.8665 - val_loss: 0.2375 - val_accuracy: 0.9119
```

```
Epoch 40/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3317 -
accuracy: 0.8663 - val_loss: 0.2313 - val_accuracy: 0.9075
Epoch 41/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3198 -
accuracy: 0.8709 - val_loss: 0.2290 - val_accuracy: 0.9094
Epoch 42/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3667 -
accuracy: 0.8422 - val_loss: 0.2208 - val_accuracy: 0.9125
Epoch 43/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3298 -
accuracy: 0.8678 - val_loss: 0.2193 - val_accuracy: 0.9131
Epoch 44/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3309 -
accuracy: 0.8678 - val_loss: 0.2210 - val_accuracy: 0.9125
Epoch 45/150
100/100 [==============================] - 11s 109ms/step - loss: 0.3145 -
accuracy: 0.8659 - val_loss: 0.2413 - val_accuracy: 0.9019
Epoch 46/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3069 -
accuracy: 0.8801 - val_loss: 0.2191 - val_accuracy: 0.9112
Epoch 47/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2920 -
accuracy: 0.8840 - val_loss: 0.2099 - val_accuracy: 0.9194
Epoch 48/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2900 -
accuracy: 0.8896 - val_loss: 0.2231 - val_accuracy: 0.9087
Epoch 49/150
100/100 [==============================] - 11s 109ms/step - loss: 0.2946 -
accuracy: 0.8770 - val_loss: 0.1961 - val_accuracy: 0.9194
Epoch 50/150
100/100 [==============================] - 11s 110ms/step - loss: 0.3007 -
accuracy: 0.8794 - val_loss: 0.2639 - val_accuracy: 0.8981
Epoch 51/150
100/100 [==============================] - 11s 109ms/step - loss: 0.2955 -
accuracy: 0.8793 - val_loss: 0.2295 - val_accuracy: 0.9100
Epoch 52/150
100/100 [==============================] - 11s 109ms/step - loss: 0.2876 -
accuracy: 0.8795 - val_loss: 0.2796 - val_accuracy: 0.8813
Epoch 53/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2956 -
accuracy: 0.8764 - val_loss: 0.2515 - val_accuracy: 0.8931
Epoch 54/150
100/100 [==============================] - 11s 112ms/step - loss: 0.2857 -
accuracy: 0.8876 - val_loss: 0.2079 - val_accuracy: 0.9169
Epoch 55/150
100/100 [==============================] - 11s 111ms/step - loss: 0.2666 -
accuracy: 0.8929 - val_loss: 0.2039 - val_accuracy: 0.9256
```

```
Epoch 56/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2941 -
accuracy: 0.8868 - val_loss: 0.1918 - val_accuracy: 0.9206
Epoch 57/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2776 -
accuracy: 0.8892 - val_loss: 0.1972 - val_accuracy: 0.9137
Epoch 58/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2788 -
accuracy: 0.8872 - val_loss: 0.1863 - val_accuracy: 0.9337
Epoch 59/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2453 -
accuracy: 0.9057 - val_loss: 0.1669 - val_accuracy: 0.9394
Epoch 60/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2587 -
accuracy: 0.8928 - val_loss: 0.2051 - val_accuracy: 0.9206
Epoch 61/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2650 -
accuracy: 0.8936 - val_loss: 0.2014 - val_accuracy: 0.9237
Epoch 62/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2539 -
accuracy: 0.9068 - val_loss: 0.2040 - val_accuracy: 0.9137
Epoch 63/150
100/100 [==============================] - 11s 106ms/step - loss: 0.2674 -
accuracy: 0.8973 - val_loss: 0.2022 - val_accuracy: 0.9169
Epoch 64/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2552 -
accuracy: 0.9036 - val_loss: 0.1772 - val_accuracy: 0.9281
Epoch 65/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2509 -
accuracy: 0.8988 - val_loss: 0.2931 - val_accuracy: 0.8800
Epoch 66/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2495 -
accuracy: 0.9017 - val_loss: 0.1719 - val_accuracy: 0.9388
Epoch 67/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2575 -
accuracy: 0.9058 - val_loss: 0.1769 - val_accuracy: 0.9312
Epoch 68/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2898 -
accuracy: 0.8872 - val_loss: 0.1859 - val_accuracy: 0.9325
Epoch 69/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2601 -
accuracy: 0.8946 - val_loss: 0.2304 - val_accuracy: 0.9087
Epoch 70/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2487 -
accuracy: 0.9001 - val_loss: 0.1727 - val_accuracy: 0.9350
Epoch 71/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2494 -
accuracy: 0.8947 - val_loss: 0.3003 - val_accuracy: 0.8750
```

```
Epoch 72/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2575 -
accuracy: 0.9008 - val_loss: 0.3079 - val_accuracy: 0.8662
Epoch 73/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2588 -
accuracy: 0.8926 - val_loss: 0.1912 - val_accuracy: 0.9194
Epoch 74/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2320 -
accuracy: 0.9049 - val_loss: 0.1706 - val_accuracy: 0.9331
Epoch 75/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2413 -
accuracy: 0.9008 - val_loss: 0.1841 - val_accuracy: 0.9319
Epoch 76/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2278 -
accuracy: 0.9062 - val_loss: 0.2325 - val_accuracy: 0.9038
Epoch 77/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2296 -
accuracy: 0.9167 - val_loss: 0.2021 - val_accuracy: 0.9256
Epoch 78/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2293 -
accuracy: 0.9105 - val_loss: 0.1932 - val_accuracy: 0.9144
Epoch 79/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2499 -
accuracy: 0.9069 - val_loss: 0.1776 - val_accuracy: 0.9337
Epoch 80/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2155 -
accuracy: 0.9161 - val_loss: 0.2161 - val_accuracy: 0.9119
Epoch 81/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2349 -
accuracy: 0.9043 - val_loss: 0.2487 - val_accuracy: 0.8944
Epoch 82/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2269 -
accuracy: 0.9111 - val_loss: 0.1935 - val_accuracy: 0.9244
Epoch 83/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2085 -
accuracy: 0.9211 - val_loss: 0.1821 - val_accuracy: 0.9231
Epoch 84/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2473 -
accuracy: 0.9074 - val_loss: 0.1489 - val_accuracy: 0.9394
Epoch 85/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2325 -
accuracy: 0.9154 - val_loss: 0.1700 - val_accuracy: 0.9362
Epoch 86/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2378 -
accuracy: 0.9059 - val_loss: 0.1602 - val_accuracy: 0.9306
Epoch 87/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2179 -
accuracy: 0.9186 - val_loss: 0.1509 - val_accuracy: 0.9456
```

```
Epoch 88/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2248 -
accuracy: 0.9195 - val_loss: 0.1828 - val_accuracy: 0.9362
Epoch 89/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2296 -
accuracy: 0.9124 - val_loss: 0.1689 - val_accuracy: 0.9312
Epoch 90/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2106 -
accuracy: 0.9232 - val_loss: 0.2973 - val_accuracy: 0.8813
Epoch 91/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2042 -
accuracy: 0.9291 - val_loss: 0.1670 - val_accuracy: 0.9369
Epoch 92/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2175 -
accuracy: 0.9182 - val_loss: 0.1892 - val_accuracy: 0.9250
Epoch 93/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2038 -
accuracy: 0.9220 - val_loss: 0.1417 - val_accuracy: 0.9444
Epoch 94/150
100/100 [==============================] - 11s 107ms/step - loss: 0.1814 -
accuracy: 0.9337 - val_loss: 0.1767 - val_accuracy: 0.9281
Epoch 95/150
100/100 [==============================] - 11s 107ms/step - loss: 0.1990 -
accuracy: 0.9228 - val_loss: 0.1272 - val_accuracy: 0.9525
Epoch 96/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2206 -
accuracy: 0.9193 - val_loss: 0.1296 - val_accuracy: 0.9469
Epoch 97/150
100/100 [==============================] - 11s 107ms/step - loss: 0.2155 -
accuracy: 0.9197 - val_loss: 0.1821 - val_accuracy: 0.9325
Epoch 98/150
100/100 [==============================] - 11s 107ms/step - loss: 0.1922 -
accuracy: 0.9320 - val_loss: 0.1432 - val_accuracy: 0.9544
Epoch 99/150
100/100 [==============================] - 11s 108ms/step - loss: 0.1901 -
accuracy: 0.9298 - val_loss: 0.1508 - val_accuracy: 0.9381
Epoch 100/150
100/100 [==============================] - 11s 111ms/step - loss: 0.2115 -
accuracy: 0.9163 - val_loss: 0.1781 - val_accuracy: 0.9250
Epoch 101/150
100/100 [==============================] - 11s 109ms/step - loss: 0.2105 -
accuracy: 0.9225 - val_loss: 0.1884 - val_accuracy: 0.9244
Epoch 102/150
100/100 [==============================] - 11s 109ms/step - loss: 0.2017 -
accuracy: 0.9202 - val_loss: 0.1215 - val_accuracy: 0.9619
Epoch 103/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2061 -
accuracy: 0.9200 - val_loss: 0.2010 - val_accuracy: 0.9081
```

```
Epoch 104/150
100/100 [==============================] - 11s 113ms/step - loss: 0.1723 -
accuracy: 0.9366 - val_loss: 0.2252 - val_accuracy: 0.9075
Epoch 105/150
100/100 [==============================] - 12s 116ms/step - loss: 0.1982 -
accuracy: 0.9237 - val_loss: 0.1685 - val_accuracy: 0.9306
Epoch 106/150
100/100 [==============================] - 11s 112ms/step - loss: 0.2148 -
accuracy: 0.9133 - val_loss: 0.1211 - val_accuracy: 0.9519
Epoch 107/150
100/100 [==============================] - 11s 108ms/step - loss: 0.1939 -
accuracy: 0.9298 - val_loss: 0.1445 - val_accuracy: 0.9556
Epoch 108/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1846 -
accuracy: 0.9260 - val_loss: 0.1715 - val_accuracy: 0.9337
Epoch 109/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1879 -
accuracy: 0.9339 - val_loss: 0.1211 - val_accuracy: 0.9619
Epoch 110/150
100/100 [==============================] - 11s 108ms/step - loss: 0.2037 -
accuracy: 0.9219 - val_loss: 0.1831 - val_accuracy: 0.9312
Epoch 111/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1964 -
accuracy: 0.9295 - val_loss: 0.1499 - val_accuracy: 0.9400
Epoch 112/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1839 -
accuracy: 0.9336 - val_loss: 0.1196 - val_accuracy: 0.9613
Epoch 113/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1888 -
accuracy: 0.9360 - val_loss: 0.1543 - val_accuracy: 0.9375
Epoch 114/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1894 -
accuracy: 0.9307 - val_loss: 0.1554 - val_accuracy: 0.9450
Epoch 115/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1671 -
accuracy: 0.9404 - val_loss: 0.1139 - val_accuracy: 0.9619
Epoch 116/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2007 -
accuracy: 0.9204 - val_loss: 0.1081 - val_accuracy: 0.9606
Epoch 117/150
100/100 [==============================] - 11s 110ms/step - loss: 0.2193 -
accuracy: 0.9206 - val_loss: 0.1192 - val_accuracy: 0.9569
Epoch 118/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1825 -
accuracy: 0.9412 - val_loss: 0.2079 - val_accuracy: 0.9325
Epoch 119/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1942 -
accuracy: 0.9220 - val_loss: 0.1243 - val_accuracy: 0.9600
```

```
Epoch 120/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1819 -
accuracy: 0.9308 - val_loss: 0.1191 - val_accuracy: 0.9531
Epoch 121/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1640 -
accuracy: 0.9378 - val_loss: 0.1627 - val_accuracy: 0.9350
Epoch 122/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1773 -
accuracy: 0.9298 - val_loss: 0.1511 - val_accuracy: 0.9362
Epoch 123/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1583 -
accuracy: 0.9412 - val_loss: 0.1603 - val_accuracy: 0.9425
Epoch 124/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1639 -
accuracy: 0.9400 - val_loss: 0.1563 - val_accuracy: 0.9362
Epoch 125/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1677 -
accuracy: 0.9373 - val_loss: 0.1303 - val_accuracy: 0.9494
Epoch 126/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1824 -
accuracy: 0.9392 - val_loss: 0.1848 - val_accuracy: 0.9250
Epoch 127/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1811 -
accuracy: 0.9360 - val_loss: 0.1055 - val_accuracy: 0.9600
Epoch 128/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1628 -
accuracy: 0.9396 - val_loss: 0.1893 - val_accuracy: 0.9275
Epoch 129/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1618 -
accuracy: 0.9321 - val_loss: 0.1248 - val_accuracy: 0.9538
Epoch 130/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1771 -
accuracy: 0.9325 - val_loss: 0.1685 - val_accuracy: 0.9369
Epoch 131/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1828 -
accuracy: 0.9322 - val_loss: 0.1352 - val_accuracy: 0.9550
Epoch 132/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1462 -
accuracy: 0.9496 - val_loss: 0.1365 - val_accuracy: 0.9550
Epoch 133/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1551 -
accuracy: 0.9501 - val_loss: 0.1437 - val_accuracy: 0.9450
Epoch 134/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1730 -
accuracy: 0.9371 - val_loss: 0.1160 - val_accuracy: 0.9588
Epoch 135/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1741 -
accuracy: 0.9372 - val_loss: 0.1358 - val_accuracy: 0.9500
```

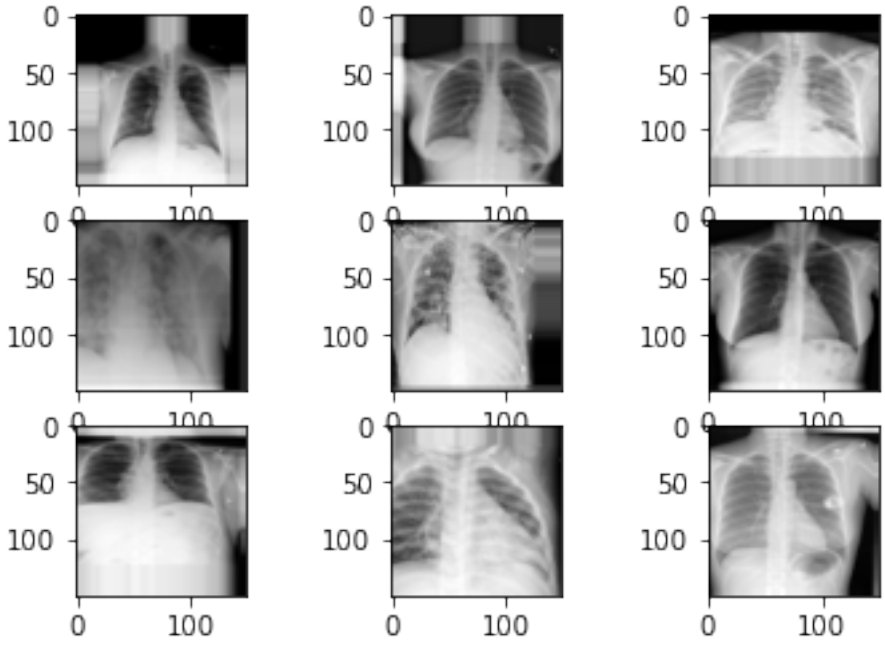
```
Epoch 136/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1521 -
accuracy: 0.9436 - val_loss: 0.1074 - val_accuracy: 0.9556
Epoch 137/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1508 -
accuracy: 0.9470 - val_loss: 0.1255 - val_accuracy: 0.9569
Epoch 138/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1835 -
accuracy: 0.9405 - val_loss: 0.1306 - val_accuracy: 0.9513
Epoch 139/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1503 -
accuracy: 0.9434 - val_loss: 0.1113 - val_accuracy: 0.9550
Epoch 140/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1668 -
accuracy: 0.9396 - val_loss: 0.1115 - val_accuracy: 0.9631
Epoch 141/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1687 -
accuracy: 0.9380 - val_loss: 0.1209 - val_accuracy: 0.9556
Epoch 142/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1664 -
accuracy: 0.9426 - val_loss: 0.1267 - val_accuracy: 0.9538
Epoch 143/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1499 -
accuracy: 0.9433 - val_loss: 0.1065 - val_accuracy: 0.9638
Epoch 144/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1498 -
accuracy: 0.9496 - val_loss: 0.1720 - val_accuracy: 0.9463
Epoch 145/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1874 -
accuracy: 0.9271 - val_loss: 0.1126 - val_accuracy: 0.9581
Epoch 146/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1543 -
accuracy: 0.9480 - val_loss: 0.1111 - val_accuracy: 0.9656
Epoch 147/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1712 -
accuracy: 0.9336 - val_loss: 0.1466 - val_accuracy: 0.9469
Epoch 148/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1544 -
accuracy: 0.9394 - val_loss: 0.1177 - val_accuracy: 0.9531
Epoch 149/150
100/100 [==============================] - 11s 110ms/step - loss: 0.1537 -
accuracy: 0.9331 - val_loss: 0.1059 - val_accuracy: 0.9625
Epoch 150/150
100/100 [==============================] - 11s 109ms/step - loss: 0.1516 -
accuracy: 0.9434 - val_loss: 0.1279 - val_accuracy: 0.9444
```

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

[ ]:
```
test_loss, test_acc = model.evaluate(test_generator)
```

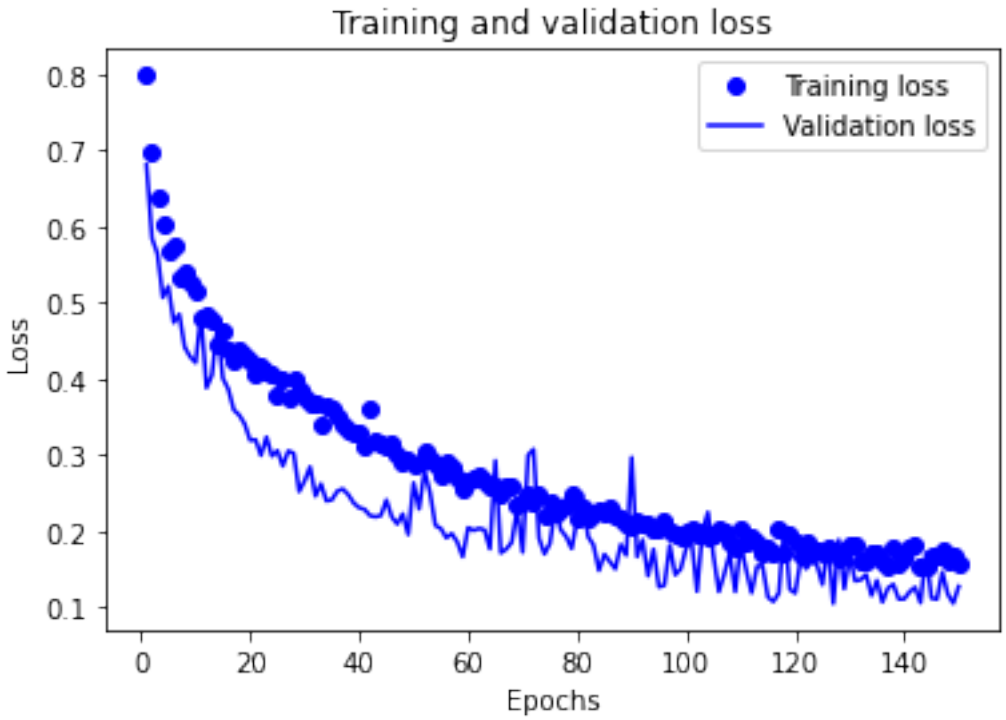
```
72/72 [==============================] - 4s 54ms/step - loss: 0.1600 - accuracy:
0.9437
```

[ ]:
```
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
```

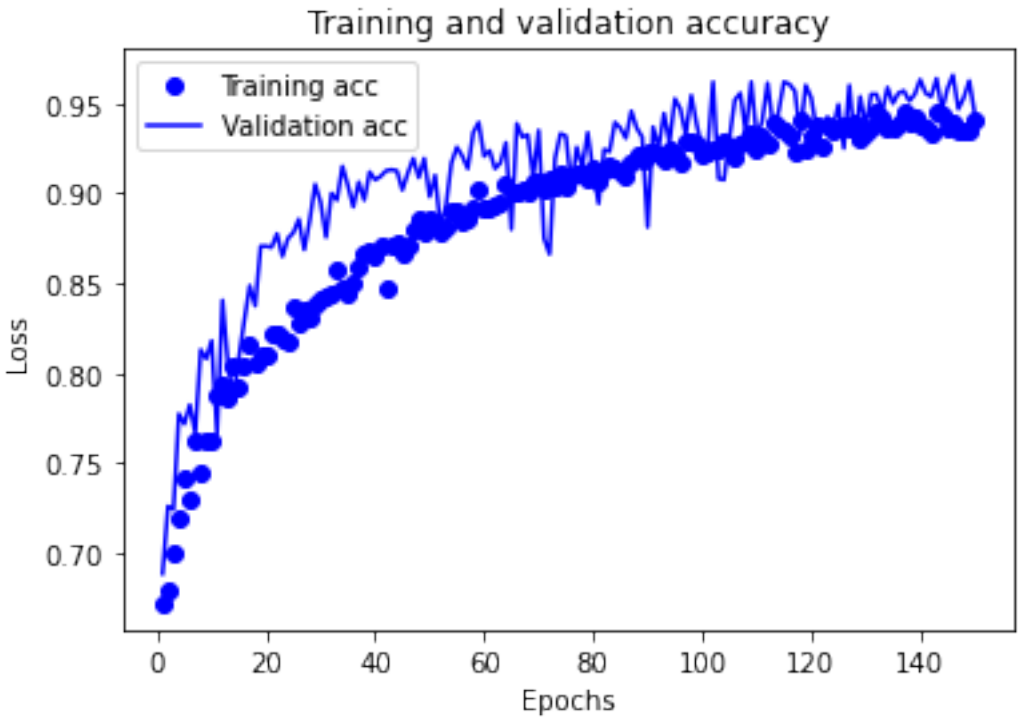


[ ]:
```
import matplotlib.pyplot as plt

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, 'bo', 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, 'bo', 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()
```


Training and validation accuracy
Training acc
Validation acc
Loss
0.95
0.90
0.85
0.80
0.75
0.70
0
20
40
60
80
100
120
140
Epochs