Script: rede neural aplicada a imagens de covid no pulmão.

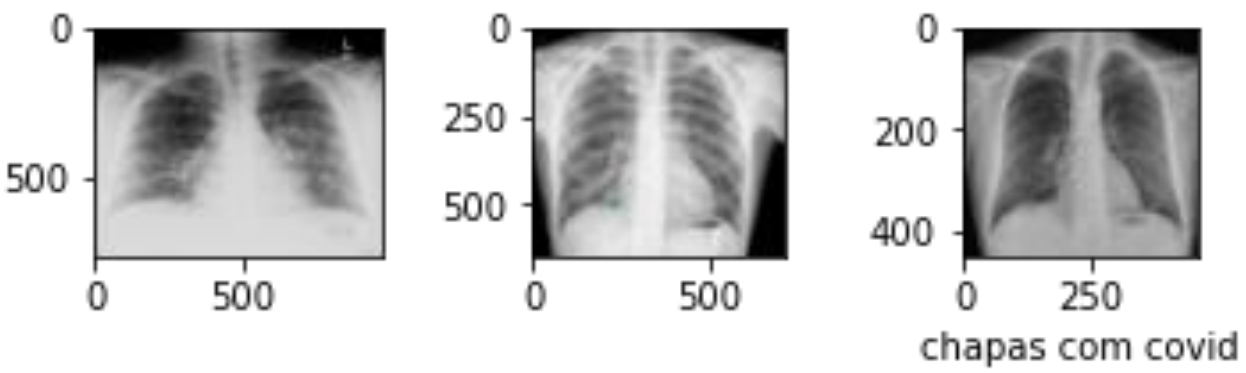
Neste script em linguagem python é usada um rede neural com 3 camadas profundas para identificar fotografias do pulmão com covid

Prof. Dourival Júnior

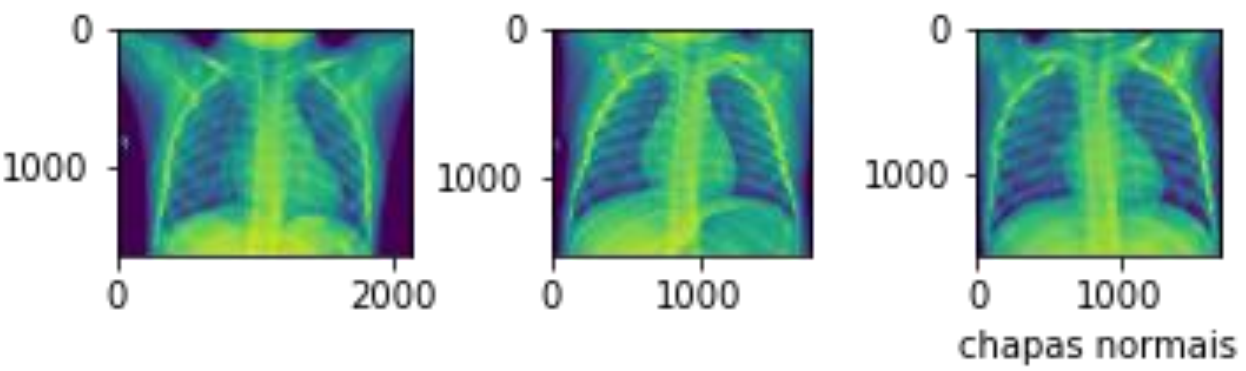
Prof.dourival.junior@gmail.com

código

```
# plot de fotos do pulmão com covid
from matplotlib import pyplot
from matplotlib.image import imread
# local do dataset
folder = 'C:/Users/qualquer/Downloads/covid/train/covid/'
# plot de 3 imagens
for i in range(3):
        pyplot.subplot(330 + 1 + i)
        filename = folder + '000001-'+str(i+1)+'.jpg'
        image = imread(filename)
        pyplot.imshow(image)
# mostra a figura
pyplot.xlabel('chapas com covid')
pyplot.show()
```



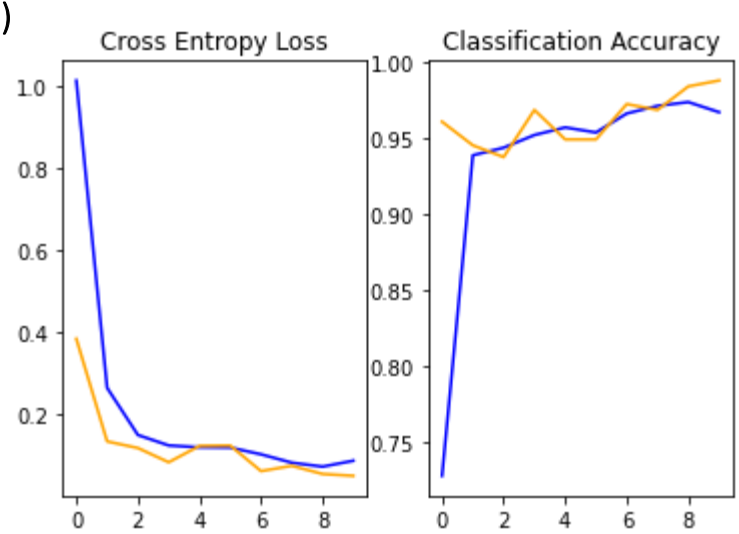

```
# plot de fotos do pulmão sem covid
from matplotlib import pyplot
from matplotlib.image import imread
# local do dataset
folder = 'C:/Users/qualquer/Downloads/covid/train/normal/'
# plot de 3 imagens
for i in range(3):
        pyplot.subplot(330 + 1 + i)
        filename = folder + '000001-'+str(i+1)+'.jpg'
        image = imread(filename)
        pyplot.imshow(image)
# mostra a figura
pyplot.xlabel('chapas normais')
pyplot.show()
```



Código principal

```
import sys
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dense
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator
# definindo o modelo de rede neural convolucional
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_uniform', padding='same', input_shape=(224, 224,
3)))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu',
kernel_initializer='he_uniform'))
model.add(Dense(1, activation='sigmoid'))
# compilando o modelo
opt = SGD(learning_rate=0.001, momentum=0.9)
model.compile(optimizer=opt, loss='binary_crossentropy',
metrics=['accuracy'])
datagen = ImageDataGenerator(rescale=1.0/255.0)
train_it =
datagen.flow_from_directory('C:/Users/qualquer/Downloads/covid/trai
n/',class_mode='binary',
                             batch_size=64, target_size=(224, 224))
test_it =
datagen.flow_from_directory('C:/Users/qualquer/Downloads/covid/tes
t/',class_mode='binary',
                            batch_size=64, target_size=(224, 224))
# aplicando o model
history =
model.fit_generator(train_it,steps_per_epoch=len(train_it),validation_d
ata=test_it,
validation_steps=len(test_it), epochs=1, verbose=1)
# calculando a exatidão do modelo
_, acc = model.evaluate_generator(test_it, steps=len(test_it), verbose=1)
print('> %.3f' % (acc * 100.0))
#salvando o modelo num diretório
model.save('C:/Users/qualquer/Downloads/covid/final_covid_model_1.
h5')
```

```
# plotando a ‘perda’ loss e a ‘acurácia’ accuracy do modelo
# plot loss
pyplot.subplot(121)
pyplot.title('Cross Entropy Loss')
pyplot.plot(history.history['loss'], color='blue', label='train')
pyplot.plot(history.history['val_loss'], color='orange', label='test')
# plot accuracy
pyplot.subplot(122)
pyplot.title('Classification Accuracy')
pyplot.plot(history.history['accuracy'], color='blue', label='train')
pyplot.plot(history.history['val_accuracy'], color='orange', label='test')
# save plot to file
#filename = sys.argv[0].split('/')[-1]
#pyplot.savefig(filename + '_plot.png')
pyplot.show()
```

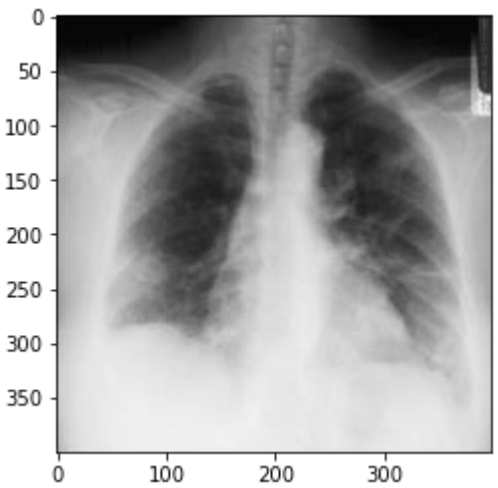


 # o modelo tem uma exatidão de 98%

Continuação do código

- **# realizando uma predição com uma ‘nova’ imagem.**
- **#from keras.preprocessing.image import load_img**
- **#from keras.preprocessing.image import img_to_array**
- **from keras.models import load_model**
- **from tensorflow.keras.preprocessing.image import load_img, img_to_array**
- **from matplotlib import pyplot**
- **# carregando uma imagem**
- **folder = 'C:/Users/qualquer/Downloads/covid/'**
- **#filename='C:/Users/qualquer/Downloads/covid/chapa_normal.jpg'**
- **filename='C:/Users/qualquer/Downloads/covid/chapa_covid.jpg'**
- **# plot da imagem escolhida ‘normal’ ou ‘covid’**
- **img = load_img(filename, target_size=(224, 224))**
- **# convert to array**
- **img = img_to_array(img)**
- **# reshape into a single sample with 3 channels**
- **img = img.reshape(1, 224, 224, 3)**
- **# center pixel data**
- **img = img.astype('float32')**
- **img = img - [123.68, 116.779, 103.939]**

```
#Continuação ...
# chamando o 'melhor' modelo salvo no diretório.
model =
load_model('C:/Users/qualquer/Downloads/covid/final_covid_model_1.h5')
            # predict the class
result = model.predict(img)
print(result[0])
prev=result[0]
if prev== 1:
   print ("normal")
else:
   print ("covid")
```




```
1/1 [==============================] - 1s 996ms/step
[0.] covid
```

O modelo de rede neural prevê com certeza de 98%. Essa fotografia é de alguém que tem o covid-19.

O modelo anterior usa somente três camadas profundas. Usando um modelo com 16 camadas chega-se a uma exatidão de 99%.

- Não fiz ainda essa adaptação.