

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e

Clique aqui para baixar o código fonte desta publicação



Neste tutorial, você aprenderá como detectar automaticamente o COVID-19 em um conjunto de dados de imagem de raio-X criado à mão usando Keras, TensorFlow e Deep Learning.

Como a maioria das pessoas no mundo atualmente, **estou genuinamente preocupado**

com o COVID-19. Encontro-me constantemente analisando minha saúde pessoal e me perguntando se / quando a contratarei.

Quanto mais eu me preocupo com isso, mais ele se transforma em um doloroso jogo mental de sintomas legítimos combinados com hipocondria:

- Acordei esta manhã sentindo-me um pouco dolorido e acabei correndo.
- Quando me levantei da cama, notei que meu nariz estava escorrendo (embora [agora seja relatado](#) que o nariz escorrendo *não* é um sintoma do COVID-19).
- Quando cheguei ao banheiro para pegar um lenço de papel, eu estava tossindo também.

No começo, não pensei muito nisso - tenho alergias ao pólen e, devido ao clima quente na costa leste dos Estados Unidos, a primavera chegou no início deste ano. Minhas alergias provavelmente estavam apenas agindo.

Mas meus sintomas não melhoraram ao longo do dia.

Na verdade, estou sentado aqui, escrevendo este tutorial, com um termômetro na boca; e olhando para baixo, vejo que se lê 99,4 ° Fahrenheit.

Meu corpo está um pouco mais frio do que a maioria, normalmente na faixa de 97,4 ° F. Qualquer coisa acima de 99 ° F é uma febre baixa para mim.

Tosse e febre baixa? Isso poderia ser COVID-19 ... ou poderia ser simplesmente minhas alergias.

É impossível saber sem um teste, e que "não saber" é o que torna essa situação tão assustadora do nível humano visceral.

Como seres humanos, não há nada mais aterrorizante do que o desconhecido.

Apesar das minhas ansiedades, tento racionalizá-las. Estou com 30 e poucos anos, muito em forma e meu sistema imunológico é forte. Vou me colocar em quarentena (apenas no caso), descansar e seguir em frente - o COVID-19 não me assusta da minha perspectiva *pessoal* de saúde (pelo menos é o que eu continuo dizendo a mim mesmo).

Dito isto, *estou* preocupado com meus parentes mais velhos, incluindo qualquer pessoa que tenha condições pré-existentes ou em um lar de idosos ou hospital. Eles são vulneráveis e seria realmente devastador vê-los partir devido ao COVID-19.

Em vez de ficar ocioso e deixar tudo o que me aflige me mantenha em baixo (seja alergias, COVID-19 ou minhas próprias ansiedades pessoais), decidi fazer o que faço de melhor - **concentre-se na comunidade geral de CV / DL escrevendo código , executando experimentos e educando outras pessoas sobre como usar a visão por computador e o aprendizado profundo em aplicativos práticos do mundo real.**

Dito isto, vou ser sincero, este *não* é o artigo mais científico que já escrevi. Longe disso, de fato. Os métodos e conjuntos de dados usados *não* seriam dignos de publicação. Mas eles servem como ponto de partida para aqueles que precisam sentir que estão fazendo *algopara* ajudar.

Eu me preocupo com você e me preocupo com esta comunidade. **Quero fazer o que puder para ajudar - este post no blog é minha maneira de lidar mentalmente com um momento difícil, ao mesmo tempo em que ajudo outras pessoas em uma situação semelhante.**

Espero que você veja isso como tal.

Dentro do tutorial de hoje, você aprenderá como:

1. Faça uma amostra de um conjunto de dados de imagens de raios-X de código aberto para pacientes com resultado positivo para COVID-19
2. Amostra de imagens de raios X “normais” (não infectadas) de pacientes saudáveis
3. Treine uma CNN para detectar automaticamente o COVID-19 em imagens de raios-X através do conjunto de dados que criamos
4. Avalie os resultados de uma perspectiva educacional

Isenção de responsabilidade: *Eu já sugeri isso, mas vou dizer explicitamente aqui. Os métodos e técnicas usados nesta publicação são apenas para fins educacionais. Este **não** é um estudo cientificamente rigoroso, nem será publicado em uma revista. Este artigo é para leitores que estão interessados em (1) Visão Computacional / Aprendizado Profundo e desejam aprender através de métodos práticos e práticos e (2) são inspirados nos eventos atuais. Peço gentilmente que você o trate como tal.*

Para saber como você pode detectar o COVID-19 em imagens de raios-X usando Keras, TensorFlow e Deep Learning, *continue lendo!*



Procurando o código fonte para este post?

[VÁ DIRETO PARA A SEÇÃO DE DOWNLOADS →](#)

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

Na primeira parte deste tutorial, discutiremos como o COVID-19 pode ser detectado nas radiografias de pacientes.

A partir daí, revisaremos nosso conjunto de dados de raios-X de tórax COVID-19.

Em seguida, mostrarei como treinar um modelo de aprendizado profundo usando o

Em seguida, mostrarei como treinar um modelo de aprendizado profundo usando o Keras e o TensorFlow para prever o COVID-19 em nosso conjunto de dados de imagens.

aviso Legal

Esta postagem no blog sobre detecção automática de COVID-19 é apenas para fins educacionais. Não se destina a ser um sistema de diagnóstico confiável e altamente preciso do COVID-19, nem foi avaliado profissional ou academicamente.

Meu objetivo é simplesmente inspirá-lo e abrir os olhos para como estudar a visão computacional / aprendizado profundo e aplicar esse conhecimento à área médica pode causar um *grande impacto* no mundo.

Simplificando: **você não precisa de um diploma em medicina para causar impacto no campo da medicina** - profissionais de aprendizado profundo que trabalham em estreita colaboração com médicos e profissionais da área médica podem resolver problemas complexos, salvar vidas e tornar o mundo um lugar melhor.

Minha esperança é que este tutorial o inspire a fazer exatamente isso.

Mas com isso dito, pesquisadores, curadores de periódicos e sistemas de revisão por pares estão sendo *sobrecarregados* com envios contendo modelos de previsão COVID-19 de qualidade questionável. **Por favor, não pegue o código / modelo desta postagem e envie-o para uma revista ou Open Science - você só aumentará o ruído.**

Além disso, se você pretende realizar pesquisas usando esta postagem (ou qualquer outro artigo da COVID-19 que encontrar on-line), consulte as [diretrizes](#) do [TRIPOD para relatar modelos preditivos](#).

Como você provavelmente sabe, a inteligência artificial aplicada ao domínio médico pode ter *consequências muito reais*. Publique ou implante esses modelos apenas se você for um especialista médico ou se estiver consultando de perto com um.

Como o COVID-19 pode ser detectado em imagens de raios-X?



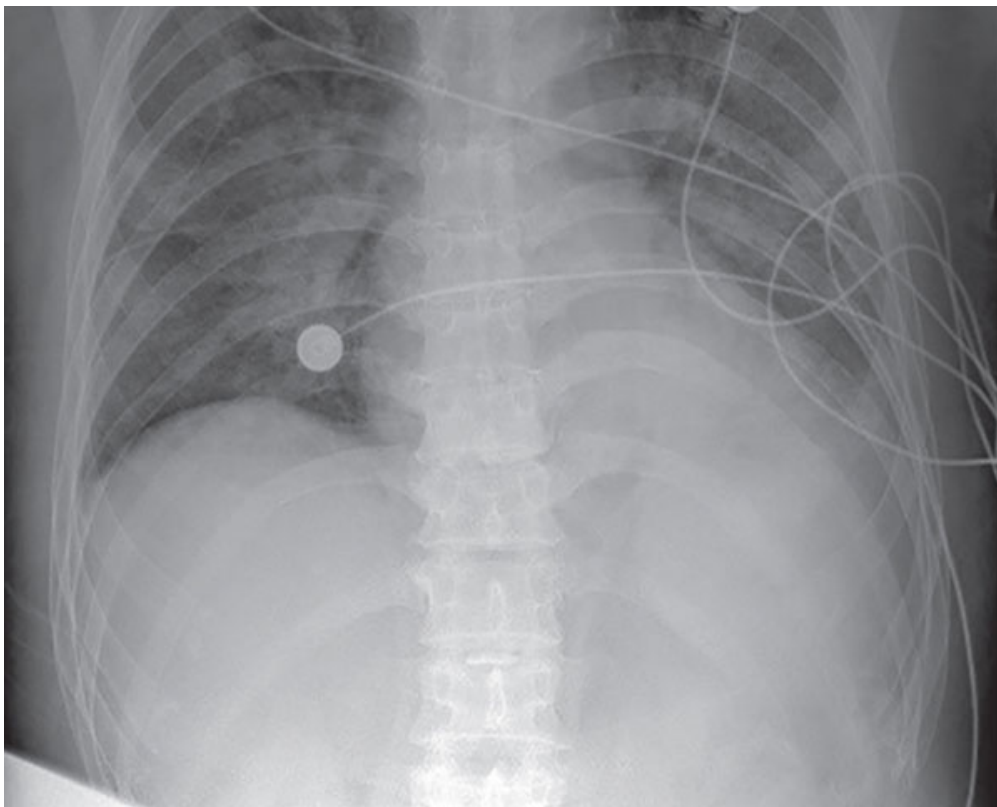


Figura 1: Exemplo de uma imagem de raios-X tirada de um paciente com teste positivo para COVID-19. Usando imagens de raios X, podemos treinar um classificador de aprendizado de máquina para detectar o COVID-19 usando Keras e TensorFlow.

Atualmente, os testes COVID-19 são difíceis de encontrar - [eles simplesmente não são suficientes](#) e não podem ser fabricados com rapidez suficiente, o que está causando pânico.

Quando há pânico, há pessoas nefastas tentando tirar vantagem de outras pessoas, [ou seja, vendendo kits de teste COVID-19 falsos](#) depois de [encontrar vítimas em plataformas de mídia social e aplicativos de bate-papo](#).

Dado que existem limitados kits de teste COVID-19, precisamos confiar em outras medidas de diagnóstico.

Para os fins deste tutorial, pensei em explorar imagens de raios-X, pois os médicos frequentemente usam raios-X e tomografias computadorizadas para diagnosticar pneumonia, inflamação pulmonar, abscessos e / ou linfonodos aumentados.

Como o COVID-19 ataca as células epiteliais que revestem nosso trato respiratório, podemos usar raios-X para analisar a saúde dos pulmões de um paciente.

E, visto que quase todos os hospitais possuem máquinas de raios-X, seria possível usar raios-X para testar o COVID-19 *sem* os kits de teste dedicados.

Uma desvantagem é que a análise de raios-X requer um especialista em radiologia e leva um tempo significativo - o que é precioso quando as pessoas estão doentes em todo o mundo. **Portanto, é necessário desenvolver um sistema de análise automatizado para economizar tempo valioso aos profissionais médicos.**

Nota: Existem publicações mais recentes que sugerem que as tomografias são melhores para diagnosticar o COVID-19, mas tudo o que temos para trabalhar neste tutorial é um conjunto de dados de imagens de raios-X. Em segundo lugar, não sou especialista em medicina e presumo que existem outros métodos mais confiáveis que médicos e profissionais da área médica usarão para detectar o COVID-19 fora dos kits de teste dedicados.

Nosso conjunto de dados de imagem de raios X de pacientes COVID-19

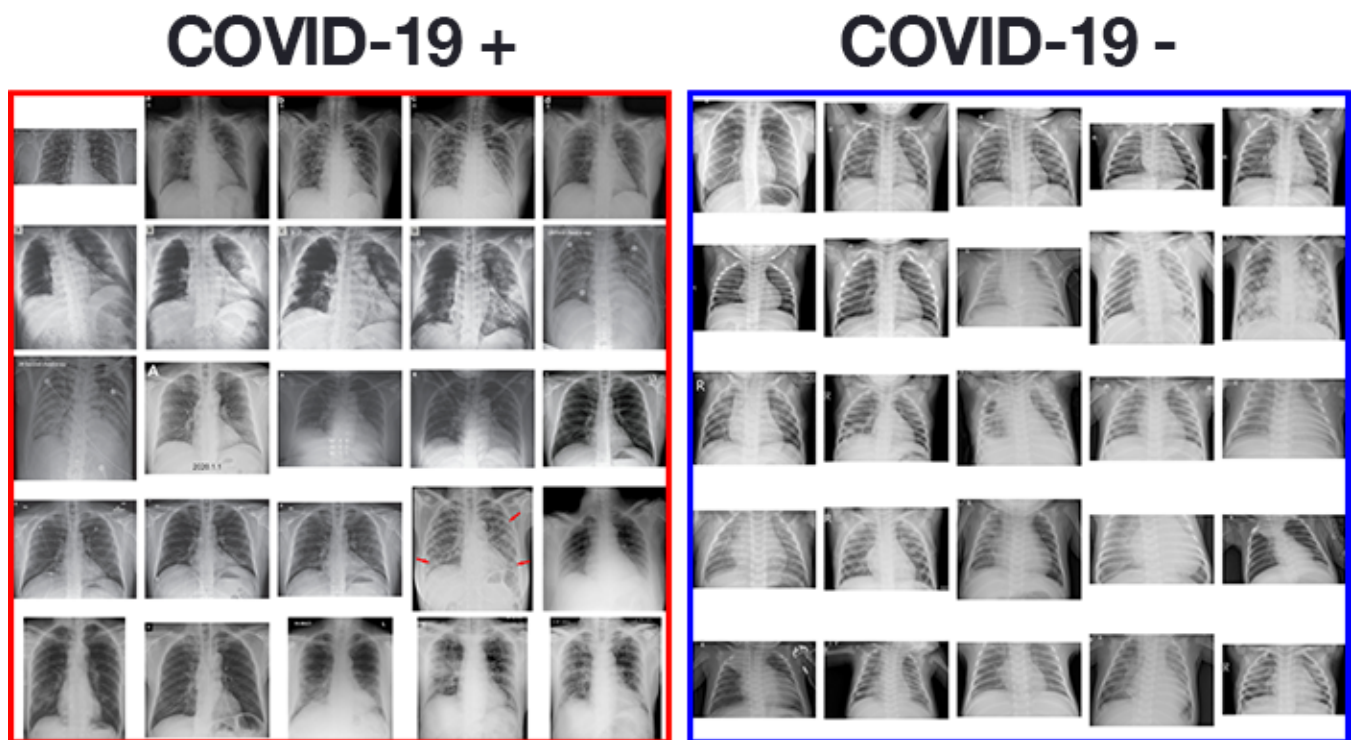


Figura 2: Dados da imagem de radiografia de tórax do CoronaVirus (COVID-19). À esquerda, temos imagens de raios X positivas (ou seja, infectadas), enquanto à direita, temos amostras negativas. Essas imagens são usadas para treinar um modelo de aprendizado profundo com o TensorFlow e o Keras para prever automaticamente se um paciente tem COVID-19 (isto é, coronavírus).

O conjunto de dados de imagem de raios-X COVID-19 que usaremos para este tutorial foi curado pelo [Dr. Joseph Cohen](#), um pós-doutorado na Universidade de Montreal.

Há uma semana, o Dr. Cohen começou a coletar imagens de raios-X de casos COVID-19 e publicá-las no repositório [GitHub](#) a [seguir](#).

Dentro do repositório, você encontrará exemplos de casos COVID-19, além de MERS, SARS e ARDS.

Para criar o conjunto de dados de imagem de raio-x COVID-19 para este tutorial, eu:

1. Analisou o arquivo `metadata.csv` encontrado no repositório do Dr. Cohen.
2. Selecionadas todas as linhas que são:
 1. Positivo para COVID-19 (ou seja, ignorando os casos MERS, SARS e ARDS).

2. Vista posterior-posterior (AP) dos pulmões. Eu usei a visão da AP como, que eu saiba, essa era a visão usada para os meus casos "saudáveis", conforme discutido abaixo; no entanto, tenho certeza de que um profissional médico será capaz de me esclarecer e me corrigir se eu estiver incorreto (o que muito bem posso ser, isso é apenas um exemplo).

No total, isso me deixou com **25 imagens de raios-X de casos positivos de COVID-19** (**Figura 2** , à esquerda).

O próximo passo foi amostrar imagens de raios-X de pacientes saudáveis .

Para fazer isso, usei [o conjunto de dados de imagens de raios X do tórax de Kaggle \(Pneumonia\)](#) e coletei **25 imagens de raios X de pacientes saudáveis** (**Figura 2** , à direita). Existem vários problemas com o conjunto de dados de radiografia de tórax do Kaggle, ou seja, rótulos ruidosos / incorretos, mas serviu como um ponto de partida suficientemente bom para esta prova de conceito do detector COVID-19.

Após reunir meu conjunto de dados, fiquei com **50 imagens no total** , divididas igualmente com 25 imagens de raios-X positivos para COVID-19 e 25 imagens de raios-X de pacientes saudáveis.

Incluí meu conjunto de dados de amostra na seção "Downloads" deste tutorial, para que você não precise recriá-lo.

Além disso, incluí meus scripts Python usados para gerar o conjunto de dados nos downloads também, mas esses scripts não serão revisados neste tutorial, pois estão fora do escopo da publicação.

Estrutura do projeto

Vá em frente e pegue o código e os dados de hoje na seção **"Downloads"** deste tutorial. A partir daí, extraia os arquivos e você verá a seguinte estrutura de diretórios:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > $ tree --dirsfirst --filelimit 10 </span> $ tree --dirsfirst --filelimit 10 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > . < / span> . </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | — conjunto de dados </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | | — oculto [ 25 cadastros] </span> C | — oculto [ 25 cadastros] </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | | — normal [ 25 cadastros] </span> Normal | — normal [ 25 cadastros] </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | — build_covid_dataset.py </span> Build— build_covid_dataset.py </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | — sample_kaggle_dataset.py </span> Exemplo— sample_kaggle_dataset.py </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span class = "google-src-text" style = "direção: ltr; text-align: left" > | — train_covid19.py </span> | — train_covid19.py </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
```

```
class = "google-src-text" style = "direção: ltr; text-align: left" > | - plot.png </span>
| - plot.png </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > | - covid19.model
</span> | - covid19.model </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > 3 diretórios , 5
arquivos </ span> 3 diretórios, 5 arquivos </ span>
```

Nossos dados de radiografia de tórax de coronavírus (COVID-19) estão no diretório

dataset/ que nossas duas classes de dados são separadas em covid/ e normal/ .

Ambos os meus scripts de construção de conjunto de dados são fornecidos; no entanto, não os revisaremos hoje.

Em vez disso, revisaremos o script

train_covid19.py que treina nosso detector COVID-19.

Vamos mergulhar e trabalhar!

Implementando nosso script de treinamento COVID-19 usando Keras e TensorFlow

Agora que analisamos nosso conjunto de dados de imagens junto com a estrutura de diretórios correspondente ao nosso projeto, passemos ao ajuste fino de uma Rede Neural Convolutiva para diagnosticar automaticamente o COVID-19 usando Keras, TensorFlow e aprendizado profundo.

Abra o arquivo

train_covid19.py na sua estrutura de diretórios e insira o seguinte código:

```
Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # import os pacote
necessários </span> # importar os pacotes necessários </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.preprocessing.image import ImageDataGenerator < / span > de
tensorflow.keras.preprocessing.image import ImageDataGenerator < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.applications import VGG16 < / span > do tensorflow.keras.applications import VGG1
< / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.layers import AveragePooling2D < / span > de tensorflow.keras.layers import
AveragePooling2D < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.layers import Dropout < / span > de tensorflow.keras.layers import Dropout < /
span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.layers import Flatten < / span > de tensorflow.keras.layers import Flatten < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
```



```

.keras.layers import Dense < / span > de tensorflow.keras.layers import Dense < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.layers import Input < / span > de tensorflow.keras.layers import Input < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.models importar Modelo < / span > não tensorflow.keras.models importar Modelo < / spa
>
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.optimizers importam Adam < / span > de tensorflow.keras.optimizers importam Adam < /
span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do tensorflow
.keras.utils import to_categorical < / span > de tensorflow.keras.utils import to_categorica
< / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do sklearn .processir
import LabelBinarizer < / span > de sklearn.preprocessing import LabelBinarizer < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do sklearn
.model_selection import train_test_split < / span > de sklearn.model_selection import
train_test_split < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do sklearn .metrics
importar classification_report < / span > de sklearn.metrics importar classification_report
/ span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > do sklearn .metrics
import confusion_matrix < / span > de sklearn.metrics import confusion_matrix < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > from imutils caminhos c
importação < / span > de caminhos de importação imutils < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > importar matplotlib
.pyplot como plt < / span > importar matplotlib.pyplot como plt < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > importar numpy como r
< / span > importar numpy como np < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > importar argparse < /
span > argumento de importação < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > importar cv2 < / span >
importação cv2 < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > importar os < / span >
importação os < / span >

```

Esse script aproveita as bibliotecas de aprendizado profundo TensorFlow 2.0 e Keras através de uma seleção de importações

```
tensorflow.keras .
```

Além disso, usamos o [scikit-learn](#) , a biblioteca Python *de fato* para aprendizado de máquina, [matplotlib](#) para plotagem e [OpenCV](#) para carregar e pré-processar imagens no conjunto de dados.

Para aprender a instalar o TensorFlow 2.0 (incluindo as bibliotecas relevantes scikit-learn, OpenCV e matplotlib), basta seguir o meu guia [Ubuntu](#) ou [macOS](#) .

Com nossas importações cuidadas, em seguida analisaremos os argumentos da linha de comando e inicializaremos os hiperparâmetros:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # construção c

```

```

span class = "google-src-text" style = "direção: ltr; text-align: left" / # construção o
analizador de argumentos e analise os argumentos </span> # construa o analisador de
argumentos e analise os argumentos </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > ap = argparse.
ArgumentParser () < / span > ap = argparse. ArgumentParser () < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > ap. add_argument ( "-d"
, "--dataset" , obrigatório = True , < / span > ap. add_argument ( "- d" , "--dataset" ,
obrigatório = True , < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > help = "caminho
para conjunto de dados de entrada" ) < / span > help = "caminho para inserir o Conjunto de
dados" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > ap. add_argument ( "-p"
, "--plot" , tipo = str, padrão = "plot.png" , < / span > ap. add_argument ( "- p" , "--plot
, digite = str, padrão = "plot.png" , < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > help = "caminho
para o gráfico de perda / precisão de saída" ) < / span > help = "caminho para gráfico de
perda / precisão de saída" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > ap. add_argument ( "-m"
, "--model" , tipo = str, padrão = "covid19.model" , < / span > ap. add_argument ( "- m" , "
-model" , tipo = str, padrão = "covid19.model", < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > help = "caminho
para o gráfico de perda / precisão de saída" ) < / span > help = "caminho para gráfico de
perda / precisão de saída" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > args = vars ( ap.
parse_args () ) < / span > args = vars ( ap. parse_args () ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # initialize a taxa de
aprendizado inicial, número de épocas para treinamento, </span> # inicialize um taxa de
aprendizado inicial, número de épocas para treinamento, </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # e tamanho do lote
</span> # e tamanho do lote </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > INIT_LR = 1e-3 < / spar
> INIT_LR = 1e-3 < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > EPOCHS = 25 < / span >
EPOCHS = 25 < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > BS = 8 < / span > BS =
< / span >

```

Nossos três [argumentos de linha de comando](#) (linhas 24 a 31) incluem:

- `--dataset` : o caminho para o nosso conjunto de dados de entrada de imagens de radiografia de tórax.
- `--plot` : um caminho opcional para um gráfico de histórico de treinamento de saída. Por padrão, o plot é nomeado `plot.png` menos que seja especificado de outra forma através da linha de comando.
- `--model` : o caminho opcional para o nosso modelo COVID-19 de saída; por padrão, ele será nomeado `covid19.model` .

A partir daí, inicializamos nossa taxa de aprendizado inicial, número de épocas de treinamento e hiperparâmetros de tamanho de lote (**Linhas 35-37**).

Agora estamos prontos para carregar e pré-processar nossos dados de raio-X:

```

< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # grab a lista de
imagens em nosso diretório de conjunto de dados e, em seguida, inicialize </span> # pegue um
lista de imagens no diretório de conjunto de dados e inicialize </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # the lista de dados
(imagens) e imagens de classe </span> # a lista de dados (imagens) e imagens de classe
</span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "[INFO]
carregando imagens ..." ) < / span > print ( "[INFO] carregando imagens ..." ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > imagePath = list (
caminhos. list_imagens ( args [ "conjunto de dados" ])) < / span > imagePath = list (
caminhos. list_imagens ( args [ "conjunto de dados" ]))) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > data = [] < / span >
dados = [] < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > labels = [] < / span >
etiquetas = [] < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # loop sobre os caminhos
da imagem </span> # volta pelos caminhos da imagem </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > para imagePath em
imagePaths: < / span > para imagePath em imagePath: < / span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # extract o rótulo
da classe do nome do arquivo </span> # extrai o rótulo da classe do nome do arquivo </span>
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > label = imagePath.
split ( os.path.sep ) [ -2 ] < / span > label = imagePath. split ( os.path.sep ) [ -2 ] < /
span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # load a imagem,
troque canais de cores e redimensione-a para ser uma imagem fixa </span> # carrega uma
imagem, canais de núcleos e redimensionados - para uma imagem fixa </span>
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direction: ltr; text-align: left" > # 224x224 pixels
enquanto ignora a proporção de aspecto </span> # 224x224 pixels enquanto ignora uma proporçã
</span>
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > image = CV2. imread
( caminho de imagem ) < / span > image = cv2. imread ( Caminho de Imagem ) < / span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > image = CV2.
cvtColor ( imagem, cv2.COLOR_BGR2RGB ) < / span > imagem = cv2. cvtColor ( imagem,
cv2.COLOR_BGR2RGB ) < / span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > image = CV2.
redimensionar ( imagem, ( 224 , 224 ) ) < / span > imagem = cv2. redimensionar ( imagem, ( 22
, 224 ) ) < / span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # update as listas
de dados e etiquetas, respectivamente </span> # atualizar as listas de dados e etiquetas,
respectivamente </span>
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > dados. anexar (
imagem ) < / span > dados. acrescentar ( imagem ) < / span >
    < span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direction: ltr; text-align: left" > rótulos. anexar
etiqueta ) < / span > etiquetas. acrescentar ( etiqueta ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # convert os dados e
rótulos para matrizes NumPy ao dimensionar o pixel </span> # converte os dados e rótulos em
matrizes NumPy enquanto dimensiona o pixel </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direction: ltr; text-align: left" > # intensidades no
intervalo [0, 1] </span> # intensidades no intervalo [0, 1] </span>

```

```

< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > data = np. array ( data
) / 255.0 < / span > data = matriz np ( dados ) / 255 , 0 < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > labels = np. matriz (
etiquetas ) < / span > etiquetas = np. matriz ( rótulos ) < / span >

```

Para carregar nossos dados, pegamos todos os caminhos para as imagens no diretório

--dataset (**Linhas 42**). Então, para cada imagePath , nós:

- Extraia o label da classe (covid ou normal) do caminho (**Linha 49**).
- Carregue a image e pré-processe-a convertendo para a ordenação de canais RGB e redimensionando-a para 224 × 224 pixels, para que fique pronta para a nossa Rede Neural Convolutiva (**Linhas 53-55**).
- Atualize nossas listas de data e labels respectivamente (**linhas 58 e 59**).

Em seguida, escalamos as intensidades de pixel para o intervalo [0, 1] e convertemos nossos

data e labels para o formato de matriz NumPy (**Linhas 63 e 64**).

Em seguida, codificaremos nossas

labels única e criaremos nossas divisões de treinamento / teste:

Deteção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # execute
codificação one-hot nas etiquetas </span> # executar codificação etiquetas one-hot nas
</span>
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > lb = LabelBinarizer ()
/ span > lb = LabelBinarizer () < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > labels = lb.
fit_transform ( etiquetas ) < / span > labels = lb. fit_transform ( etiquetas ) < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > labels = to_categorical
( etiquetas ) < / span > labels = to_categorical ( etiquetas ) < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # partição os dados par
treinamento e teste são divididos usando 80% de </span> # particionamento de dados em
treinamentos e testes de divisão usando 80% dos </span>
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # the dados para
treinamento e os 20% restantes para testes </span> # os dados para treinamento e os 20%
restantes para teste </span>
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > ( trainX , testX,
trainY, testY ) = train_test_split ( dados, etiquetas, < / span > ( trainX, testX, trainY,
testY ) = train_test_split ( dados, rótulos, < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > test_size = 0,20 ,
estratificar = etiquetas, estado_aleatório = 42 ) < / span > test_size = 0,20 , estratifica
= etiquetas, estado_aleatório = 42 ) < / extensão >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # initialize o objeto c
aumento de dados de treinamento </span> # inicialize o objeto de aumento de dados de
treinamento </span>
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa

```

```

class = "google-src-text" style = "direção: ltr; text-align: left" > trainAug =
ImageDataGenerator (< / span > trainAug = ImageDataGenerator (< / span >
    < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > rotation_range = 1
, < / span > intervalo_ rotação = 15 , < / span >
    < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda" >
fill_mode = "mais próximo" ) < / span > fill_mode = "mais próximo" ) < / span >

```

A codificação de

labels quente ocorre nas **linhas 67 a 69**, o que significa que nossos dados estarão no seguinte formato:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [[ 0. < / span > [
0. < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout =
"_tipoff ()" > < span class = "google-src-text" style = "direction: ltr; alinhamento de
texto: left" > 1. ] < / span > 1. ] < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [ 0 . < / span > [
0 < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > < span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda"
> 1. ] < / span > 1. ] < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [ 0 . < / span > [
0 < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > < span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda"
> 1. ] < / span > 1. ] < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direction: ltr; text-align: left" > ... < / span >
... < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [ 1 . < / span > [
1 < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > < span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda"
> 0. ] < / span > 0. ] < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [ 1 . < / span > [
1 < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > < span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda"
> 0. ] < / span > 0. ] < / span >
< span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > [ 1 . < / span > [
1 < / span > < span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > < span class = "google-src-text" style = "direção: ltr; alinhamento de texto: esquerda"
> 0. ]] < / span > 0. ]] < / span >

```

Cada etiqueta codificada consiste em uma matriz de dois elementos com um dos elementos sendo "quente" (ou seja,

1) versus "não" (ou seja, 0).

As linhas 73 e 74 constroem nossa divisão de dados, reservando 80% dos dados para treinamento e 20% para testes.

Para garantir a generalização do nosso modelo, realizamos o [aumento de dados](#) definindo a rotação aleatória da imagem em 15 graus no sentido horário ou anti-horário.

As linhas 77-79 inicializam o objeto gerador de aumento de dados.

A partir daqui, inicializaremos nosso modelo VGGNet e o configuraremos para o [ajuste fino](#) :

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # load a rede
VGG16, garantindo que os conjuntos de camadas FC principais sejam deixados </span> # carregue
uma rede VGG16, permitindo que os conjuntos de camadas FC principais sejam deixados </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # desativado </span> #
fóruns </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > baseModel = VGG16 (
pesos = "imagenet" , include_top = Falso , < / span > baseModel = VGG16 ( pesos = "imagenet"
, include_top = Falso , < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > input_tensor =
Entrada ( formato = ( 224 , 224 , 3 )) < / span > input_tensor = Entrada ( forma = ( 224 ,
224 , 3 )) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # construção a cabeça c
modelo que será colocada em cima da </span> # construção da cabeça do modelo que será
colocada em cima da </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # the modelo base
</span> # o modelo base </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel =
baseModel.output < / span > headModel = baseModel.output < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel =
AveragePooling2D ( POOL_SIZE = ( 4 , 4 )) ( headModel ) < / span > headModel =
AveragePooling2D ( POOL_SIZE = ( 4 , 4 )) ( headModel ) </ span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel = Achatar (
nome = "achatar" ) ( headModel ) < / span > headModel = Achatar ( nome = "achatar" ) (
headModel ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel = Denso ( 64
ativação = "relu" ) ( headModel ) < / span > headModel = Denso ( 64 , ativação = " relu " )
headModel ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel = Dropout (
0,5 ) ( headModel ) < / span > headModel = Dropout ( 0,5 ) ( headModel ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > headModel = Denso ( 2 ,
ativação = "softmax" ) ( headModel ) < / span > headModel = Denso ( 2 , ativação = "softmax"
) ( headModel ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # place o modelo FC
principal na parte superior do modelo base (isso se tornará </span> # coloque o modelo FC
principal sobre o modelo base (isso se tornará </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # the modelo real que
iremos treinar) </span> # o modelo real que treinaremos) </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > model = Model ( entradas
= baseModel.input, saídas = headModel ) < / span > model = Modelo ( entradas =
baseModel.input, exportações = headModel ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # loop em todas as

camadas no modelo base e congelá-las para que elas </span> # passem por todas as camadas no
modelo base e congele - como para elas </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # * não * será
atualizado durante o primeiro processo de treinamento </span> # * not * será atualizado
durante o primeiro processo de treinamento </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
```

```

class = "google-src-text" style = "direção: ltr; text-align: left" > para a camada em
baseModel.layers: < / span > para uma camada em baseModel.layers: < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > camada.trainable
Falso < / span > layer.trainable = Falso < / span >

```

As linhas 83 e 84 instanciam a rede VGG16 com pesos pré-treinados no ImageNet, deixando a cabeça da camada FC.

A partir daí, construímos uma nova cabeça de camada totalmente conectada, consistindo em camadas

POOL => FC = SOFTMAX (**linhas 88-93**) e **anexamos -a** no topo do VGG16 (**linha 97**).

Em seguida, congelamos os pesos

CONV do VGG16, de modo que *apenas* a cabeça da camada FC seja treinada (**Linhas 101-102**); isso completa nossa configuração de ajuste fino.

Agora estamos prontos para compilar e treinar nosso modelo de aprendizado profundo COVID-19 (coronavírus):

Deteção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # compile nosso
modelo </span> # compile nosso modelo </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "[INFO]
compilando modelo ..." ) < / span > print ( "[INFO] compilando modelo ..." ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > opt = Adam ( lr =
INIT_LR, decaimento = INIT_LR / EPOCHS ) < / span > opt = Adam ( lr = INIT_LR, decaimento =
INIT_LR / EPOCHS ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > modelo.compile ( perc
= "binary_crossentropy" , otimizador = opt, < / span > modelo.compile ( perda =
"binary_crossentropy" , otimizador = opt, < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > metrics = [
"precisão" ]) < / span > métricas = [ "precisão" ]) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > # train o chefe da rede
</span> # treinar o chefe da rede </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "[INFO] cabeça
de treinamento ..." ) < / span > print ( "cabeça de treinamento [INFO] ..." ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < span
class = "google-src-text" style = "direção: ltr; text-align: left" > H = modelo.
fit_generator (< / span > H = modelo.fit_generator (< / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > trainAug.fluxo (
trainX, trainY, batch_size = BS ) , < / span > trainAug.fluxo ( trainX, trainY, batch_size
BS ) , < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > steps_per_epoch =
len ( trainX ) // BS, < / span > steps_per_epoch = len ( trainX ) // BS, < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > validation_data =
testX, testY ) , < / span > validation_data = ( testX, testY ) , < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > validation_steps =
len ( testX ) // BS, < / span > validation_steps = len ( testX ) // BS, < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > epochs = EPOCHS )

```

| / span > épocas = EPOCHS) < / span >

As linhas 106-108 compilam a rede com o [declínio da taxa de aprendizado](#) e o otimizador

Adam . Dado que este é um problema de 2 classes, usamos a perda "binary_crossentropy" em vez da crossentropy categórica.

Para iniciar nosso processo de treinamento em rede neural COVID-19, fazemos uma chamada para [o método fit_generator de Keras](#), enquanto transmitimos nossos dados de radiografia de tórax por meio de nosso objeto de aumento de dados (Linhas 112-117).

A seguir, avaliaremos nosso modelo:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # make previsões
sobre o conjunto de testes </span> # faça tocar no conjunto de testes </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "[INFO]
avaliando a rede ..." ) < / span > print ( "[INFO] avaliando uma rede ..." ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > predIdxs = modelo.
predizer ( testX, lote_size = BS ) < / span > predIdxs = modelo. predizer ( testX, batch_siz
= BS ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # para cada imagem no
conjunto de testes, precisamos encontrar o índice do </span> # para cada imagem no conjunto
de testes, encontrar o índice do </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # label com a maior
probabilidade prevista correspondente </span> # etiqueta com a maior probabilidade
correspondente </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > predIdxs = np. argmax (
predIdxs, eixo = 1 ) < / span > predIdxs = np. argmax ( predIdxs, eixo = 1 ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # show um relatório de
classificação bem formatado </span> # mostrar um relatório de classificação bem formatado
</span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print (
classification_report ( tESTY. argmax ( eixo = 1 ) , predIdxs, < / span > print ( Relatório
de classificaçã o ( tESTY. argmax ( eixo = 1 ) , predIdxs,< / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > target_names =
lb.classes_ )) < / span > target_names = lb.classes_ )) < / span >
```

Para avaliação, primeiro fazemos previsões no conjunto de testes e obtemos os índices de previsão (Linhas 121-125).

Em seguida, geramos e imprimimos um relatório de classificação usando o utilitário auxiliar do scikit-learn (Linhas 128 e 129).

Em seguida, calcularemos uma matriz de confusão para uma avaliação estatística adicional:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # computação a
```

```

span class = "google-src-text" style = "direção: ltr; text-align: left" / # computação a
matriz de confusão ee use-a para derivar a matéria-prima </span> # calcular uma matriz de
confusão e usar-a para derivar uma matéria prima </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # precisão ,
sensibilidade e especificidade </span> # precisão, sensibilidade e especificidade </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > cm = confusion_matrix (
testY. argmax ( eixo = 1 ) , predIdxs ) < / span > cm = confusion_matrix ( testY. argmax (
eixo = 1 ) , predIdxs ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > total = soma ( soma ( c
)) < / extensão > total = soma ( soma ( cm )) < / extensão >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > acc = ( cm [ 0 , 0 ] +
cm [ 1 , 1 ]) / total < / span > acc = ( cm [ 0 , 0 ] + cm [ 1 ,1 ]) / total < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > sensibilidade = cm [ 0
0 ] / ( cm [ 0 , 0 ] + cm [ 0 , 1 ]) < / span > sensibilidade = cm [ 0 ,0 ] / ( cm [ 0 , 0 ]
cm [ 0 , 1 ]) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > specificity = cm [ 1 ,
] / ( cm [ 1 , 0 ] + cm [ 1 , 1 ]) < / span > especificidade = cm [ 1 ,1 ] / ( cm [ 1 , 0 ] +
cm [ 1 , 1 ]) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > # show a matriz de
confusão, precisão, sensibilidade e especificidade </span> # mostra uma matriz de confusão,
precisão, sensibilidade e especificidade </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( cm ) < / span >
IMPRESSA o ( cm ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "acc: {:.4f}"
formato ( acc )) < / span > print ( "acc: {:.4f}" . formato ( acc )) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "sensibilidade:
{:.4f}" . formato ( sensibilidade )) < / span > print ( "sensibilidade: {:.4f}" . formato
sensibilidade )) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "especificidade
{:.4f}" . formato ( especificidade )) < / span > print ( "especificidade: {:.4f}" . format
( especificidade )) < / span >

```

Aqui nós:

- Gere uma matriz de confusão (**linha 133**)
- Use a matriz de confusão para derivar a precisão, sensibilidade e especificidade (**linhas 135-137**) e imprima cada uma dessas métricas (**linhas 141-143**)

Em seguida, plotamos nosso histórico de precisão / perda de treinamento para inspeção, produzindo o plot para um arquivo de imagem:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # plot a perda e a
precisão do treinamento </span> # traçar a perda e a precisão do treinamento </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > N = EPOCHS < / span > N
= EPOCHS < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. estilo. use (
"ggplot" ) < / span > plt.style. use ( "ggplot" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. figure () < / spar
> plt. figure () < / span >
< span class = "notranslate" onmouseover = " tipon (this)" onmouseout = " tipoff ()" > < spa

```

```

class = "google-src-text" style = "direção: ltr; text-align: left" > plt._plot ( np. arange
0 , N ) , H.history [ "loss" ] , label = "train_loss" ) < / span > plt. plot ( np. arange (
, N ) , H.history [ "perda" ] , label = "train_loss" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt._plot ( np. arange
0 , N ) , H.history [ "val_loss" ] , label = "val_loss" ) < / span > plt. plot ( np. arange
0 , N ) , H.history [ "val_loss" ] , label = "val_loss" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt._plot ( np. arange
0 , N ) , H.history [ " precision " ] , label = "train_acc" ) < / span > plt. plot ( np.
arange ( 0 , N ) , H.history [ "precisão" ] , label = "train_acc" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt._plot ( np. arange
0 , N ) , H.history [ "val_accuracy" ] , label = "val_acc" ) < / span > plt. plot ( np.
arange ( 0 , N ) , H.history [ "val_accuracy" ] , label = "val_acc" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. title ( "Perda e
precisão do treinamento no conjunto de dados COVID-19" ) < / span > plt. title ( "Perda e
precisão do treinamento no conjunto de dados COVID-19" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. xlabel ( "Epoch #"
) < / span > plt. xlabel ( "Época #" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. ylabel ( "Perda /
precisão" ) < / span > plt. rótulo ( "Perda / precisão" ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. legenda ( loc =
"canto inferior esquerdo" ) < / span > plt. legenda ( loc = "canto inferior esquerdo" ) < /
span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > plt. savefig ( args [
"plot" ] ) < / span > plt. savefig ( args [ "plot" ] ) < / span >

```

Finalmente, serializamos nosso modelo de classificador

tf.keras COVID-19 em disco:

```

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <
span class = "google-src-text" style = "direção: ltr; text-align: left" > # serialize o
modelo para o disco </span> # serialize o modelo em disco </span>
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > print ( "[INFO] salvanc
modelo COVID-19 detector ..." ) < / span > print ( "[INFO] salvando o modelo fazer detector
COVID-19 ..." ) < / span >
< span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > < spa
class = "google-src-text" style = "direção: ltr; text-align: left" > modelo. save ( args [
"modelo" ] , save_format = "h5" ) < / span > modelo. save ( args [ "modelo" ] , save_format
"h5" ) < / span >

```

Treinando nosso detector COVID-19 com Keras e TensorFlow

Com nosso script

train_covid19.py implementado, agora estamos prontos para treinar nosso detector COVID-19 automático.

Certifique-se de usar a seção **“Downloads”** deste tutorial para baixar o código-fonte, o conjunto de dados de raios-X COVID-19 e o modelo pré-treinado.

A partir daí, abra um terminal e execute o seguinte comando para treinar o detector COVID-19:

Detecção de COVID-19 em imagens de raios-X com Keras, TensorFlow e Deep Learning

```

<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > $ python
train_covid19.py - conjunto de dados do conjunto de dados </span> $ python train_covid19.py
conjunto de dados do conjunto de dados </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [INFO ] carregando
imagens ... </span> [INFO] carregando imagens ... </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [INFO ] modelo de
compilação ... </span> Modelo de compilação [INFO] ... </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [INFO ] chefe de
treinamento ... </span> [INFO] chefe de treinamento ... </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 1 / 25 </ span>
Época 1 / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 20s 4s / step - perda: 0. 7169 - precisão: 0. 6000 -
val_loss: 0. 6590 - val_accuracy: 0. 5000 </ span> 5 / 5 [=====] -
20s 4s / step - perda: 0 , 7169 - precisão: 0 , 6000 - val_loss: 0.6590- precisão val .: 0.
5000 </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > época 2 / 25 </ span>
Época 2 / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 0s 86ms / passo - perda: 0. 8088 - precisão: 0. 4250 -
val_loss: 0. 6112 - val_accuracy: 0. 9000 </ span> 5 / 5 [=====] -
0s 86ms / etapa - perda: 0. 8088 - precisão: 0. 4250 - perda de valor: 0. 6112 - precisão de
valor:0. 9000</span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 3 / 25 </ span>
Época 3 / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 0s 99ms / step - perda: 0. 6809 - precisão: 0. 5500 -
val_loss: 0. 6054 - val_accuracy: 0. 5000 </ span> 5 / 5 [=====] -
0s 99ms / step - perda: 0. 6809 - precisão: 0. 5500 - perda de valor: 0. 6054 - precisão de
valor:0. 5000</span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > época 4 / 25 </ span>
Época 4 / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 1s 100ms / step - perda: 0. 6723 - precisão: 0. 6000 -
val_loss: 0. 5771 - val_accuracy: 0. 6000 </ span> 5 / 5 [=====] -
1s 100ms / step - perda: 0 , 6723 - precisão: 0 , 6000 - val_loss:0. 5771- precisão val .: (
6000 </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > .. . </span> ...
</span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 22 / 25 </ spar
Época 22 de / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 0s 99ms / step - perda: 0. 3271 - precisão: 0. 9250 -
val_loss: 0. 2902 - val_accuracy: 0. 9000 </ span> 5 / 5 [=====] -
0s 99ms / step - perda: 0. 3271 - precisão: 0. 9250 - val_loss: 0 , 2902- precisão val .: 0
9000 </span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 23 / 25 </ spar
Época 23 de / 25 </ span>
<span class = "notranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 0s 99ms / step - perda: 0. 3634 - precisão: 0. 9250 -
val_loss: 0. 2690 - val_accuracy: 0. 9000 </ span> 5 / 5 [=====] -
0s 99ms / step - perda: 0 , 3634 - precisão: 0 , 9250 - val_loss: 0.2690- precisão val .: 0.

```

```

9000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 24 / 25 </ spar
Época 24 / 25 </ span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 27s 5s / step - perda: 0. 3175 - precisão: 0. 9250 -
val_loss: 0. 2395 - val_accuracy: 0. 9000 </ span> 5 / 5 [===== ] -
27s 5s / step - perda: 0 , 3175 - precisão: 0 , 9250 - val_loss: 0.2395- precisão val .: 0.
9000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > Época 25 / 25 </ spar
Época 25 / 25 </ span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > 5 / 5
[=====] - 1s 101ms / step - perda: 0. 3655 - precisão: 0. 8250 -
val_loss: 0. 2522 - val_accuracy: 0. 9000 </ span> 5 / 5 [===== ] -
1s 101ms / step - perda: 0 , 3655 - precisão: 0 , 8250 - val_loss:0. 2522- precisão val .: (
9000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [INFO ] avaliando a rec
... </span> [INFO] avaliando rede ... </span>
    <span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipof
()" > <span class = "google-src-text" style = "direction: ltr; text-align: left" >
recuperação de precisão suporte para pontuação f1 </span> suporte de precisão pontuação para
f1 </span>
    <span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" >
<span class = "google-src-text" style = "direction: ltr; text-align: left" > covid 0 . 83 1.
00 0. 91 5 </ span> covid 0 , 83 1 , 00 0 , 91 5 </ span>
    <span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" >
<span class = "google-src-text" style = "direção: ltr; text-align: left" > normal 1 . 00 0.
80 0. 89 5 </ span> normal de 1 , 00 0 , 80 0 , 89 5 </ span>
    <span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" >
<span class = "google-src-text" style = "direção: ltr; text-align: left" > precisão 0 . 90 1
</ span> Precisão 0 , 90 10 </ span>
    <span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" >
<span class = "google-src-text" style = "direction: ltr; text-align: left" > macro avg 0. 92
0. 90 0. 90 10 </span> média 0 , 92 0 , 90 0 , 90 10 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direction: ltr; text-align: left" > média ponderada 0. 92
0. 90 0. 90 10 </span> média ponderada 0 , 92 0 , 90 0 , 90 10 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [[ 5 0 ] </span> [ 5 0
</span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [ 1 4 ]] </span> [ 1 4
]] </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > acc: 0. 9000 </span>
acc: 0. 9000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > sensibilidade: 1. 0000
</span> sensibilidade: 1. 0000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > especificidade: 0. 8000
</span> especificidade: 0. 8000 </span>
<span class = "nottranslate" onmouseover = "_tipon (this)" onmouseout = "_tipoff ()" > <span
class = "google-src-text" style = "direção: ltr; text-align: left" > [INFO ] salvar COVID- 1
modelo de detector ... </ span> [INFO] salvando o modelo de detector COVID- 19 ... </ span>

```

Diagnóstico automático de COVID-19 a partir de resultados de imagens de raios-X

Isonção de responsabilidade: A seção a seguir não reivindica, nem pretende

"resolver" a detecção COVID-19. Está escrito apenas no contexto e a partir dos resultados deste tutorial. É um exemplo para profissionais de visão computacional e aprendizado profundo, para que eles possam aprender sobre várias métricas, incluindo precisão bruta, sensibilidade e especificidade (e as vantagens e desvantagens que devemos considerar ao trabalhar com aplicativos médicos). Novamente, esta seção / tutorial **não pretende resolver a detecção de COVID-19**.

Como você pode ver pelos resultados acima, nosso detector COVID-19 automático está obtendo ~ **90-92% de precisão** em nosso conjunto de dados de amostra com base *apenas* em imagens de raios-X - nenhum outro dado, incluindo localização geográfica, densidade populacional etc. foi usado para treinar esse modelo.

Também estamos obtendo **100% de sensibilidade** e **80% de especificidade**, o que implica que:

- Dos pacientes que possuem COVID-19 (isto é, verdadeiros positivos), **poderíamos identificá-los com precisão como "positivos para COVID-19" 100% do tempo** usando nosso modelo.
- Dos pacientes que *não* possuem COVID-19 (isto é, verdadeiros negativos), **poderíamos identificá-los com precisão como "COVID-19 negativo" apenas 80% do tempo** usando nosso modelo.

Como mostra nosso gráfico de histórico de treinamento, nossa rede não está adaptando demais, apesar de ter dados de treinamento *muito limitados* :



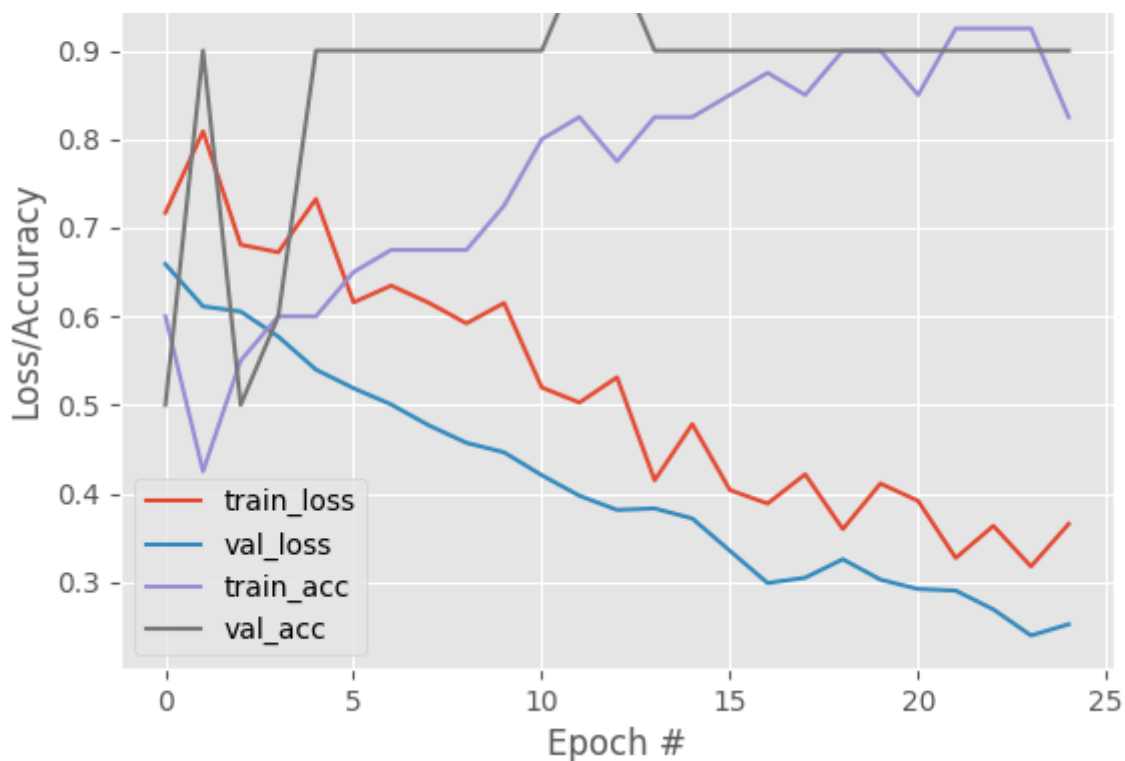


Figura 3: Este gráfico de histórico de treinamento de aprendizado profundo, mostrando curvas de precisão e perda, demonstra que nosso modelo não está adaptado, apesar dos dados limitados de treinamento em raios-X COVID-19 usados em nosso modelo Keras / TensorFlow.

Ser capaz de detectar com precisão o COVID-19 com 100% de precisão é ótimo; no entanto, nossa verdadeira taxa negativa é um pouco preocupante - **não queremos classificar alguém como "COVID-19 negativo" quando é "COVID-19 positivo"**.

De fato, a *última* coisa que queremos fazer é dizer a um paciente que eles são negativos para COVID-19 e depois levá-los para casa e infectar sua família e amigos; transmitindo assim a doença ainda mais.

Também queremos ter *muito cuidado* com nossa taxa de falsos positivos - **não queremos classificar alguém como "COVID-19 positivo" por engano, colocá-lo em quarentena com outros pacientes positivos com COVID-19 e infectar uma pessoa que nunca teve o vírus. vírus.**

Equilibrar sensibilidade e especificidade é *incrivelmente desafiador* quando se trata de aplicações médicas, *especialmente* doenças infecciosas que podem ser transmitidas rapidamente, como o COVID-19.

Quando se trata de visão médica computadorizada e aprendizado profundo, devemos sempre estar atentos ao fato de que nossos modelos preditivos podem ter *consequências muito reais* - **um diagnóstico perdido pode custar vidas**.

Novamente, esses resultados são coletados apenas para *fins educacionais*. Este artigo e os resultados que o acompanham *não* se destinam a ser um artigo de periódico nem estão em conformidade com as [diretrizes](#) do [TRIPOD sobre relatórios de modelos preditivos](#). Eu sugiro que você consulte estas diretrizes para obter mais informações, se você estiver interessado.

Limitações, melhorias e trabalhos futuros



Figura 4: Atualmente, especialistas em inteligência artificial (IA) e profissionais de aprendizado profundo estão sofrendo com a falta de dados COVID-19 de qualidade para treinar efetivamente sistemas automáticos de detecção baseados em imagem. ([fonte da imagem](#))

Uma das maiores limitações do método discutido neste tutorial são os dados.

Simplesmente não temos dados suficientes (confiáveis) para treinar um detector COVID-19.

Os hospitais já estão sobrecarregados com o número de casos COVID-19 e, dados os direitos e a confidencialidade dos pacientes, fica *ainda mais difícil* montar conjuntos de dados de imagens médicas de qualidade em tempo hábil.

Imagino que nos próximos 12 a 18 meses teremos mais conjuntos de dados de imagem COVID-19 de alta qualidade; mas, por enquanto, só podemos nos contentar com o que temos.

Fiz o meu melhor (considerando meu estado mental e saúde física atuais) para montar um tutorial para meus leitores interessados em aplicar visão computacional e aprendizado profundo à pandemia do COVID-19, devido ao meu tempo e recursos limitados; no entanto, devo lembrá-lo de que **não sou especialista médico treinado**.

Para que o detector COVID-19 seja implantado em campo, ele precisaria passar por testes *rigorosos* por profissionais médicos treinados, trabalhando lado a lado com profissionais especialistas em aprendizado profundo. O método aqui coberto hoje certamente *não* é um método desse tipo e destina-se apenas a fins educacionais.

Além disso, precisamos nos preocupar com o que o modelo está realmente “aprendendo”.

Como discuti no [tutorial Grad-CAM da semana passada](#), é possível que nosso modelo seja de padrões de aprendizado que *não* sejam *relevantes* para o COVID-19 e, em vez disso, sejam apenas variações entre as duas divisões de dados (ou seja, diagnóstico positivo versus negativo do COVID-19).

Seria necessário um profissional médico treinado e testes rigorosos para validar os resultados do nosso detector COVID-19.

E finalmente, os futuros (e melhores) detectores COVID-19 serão multimodais.

No momento, estamos usando *apenas* dados de imagem (raios X) - os melhores detectores automáticos de COVID-19 devem aproveitar várias fontes de dados não limitadas *apenas* às imagens, incluindo dados vitais do paciente, densidade populacional, localização geográfica etc. Os **dados de imagem por si só são normalmente não é suficiente para esses tipos de aplicativos.**

Por essas razões, devo enfatizar mais uma vez que este tutorial é *apenas* para fins educacionais - não é para ser um detector COVID-19 robusto.

Se você acredita que você ou um ente querido possui o COVID-19, siga os protocolos descritos pelo [Centro de Controle de Doenças \(CDC\)](#) , [Organização Mundial da Saúde \(OMS\)](#) ou país, estado ou jurisdição local.

Espero que você tenha gostado deste tutorial e o tenha achado educacional. Também espero que este tutorial sirva como ponto de partida para qualquer pessoa interessada em aplicar visão computacional e aprendizado profundo à detecção automática de COVID-19.

Qual é o próximo?

Normalmente, termino minhas postagens recomendando um dos meus livros / cursos, para que você possa aprender mais sobre como aplicar a Visão Computacional e o Aprendizado Profundo em seus próprios projetos. Por respeito à gravidade do coronavírus, não vou fazer isso - não é a hora nem o lugar.

Em vez disso, o que *vou* dizer é que estamos em uma estação muito assustadora da vida agora.

Como todas as estações do ano, isso *vai* passar, mas precisamos nos agachar e nos preparar para um inverno frio - é provável que o pior ainda esteja por vir.

Para ser franco, me sinto *incrivelmente* deprimido e isolado. Eu vejo:

- Tanques nos mercados de ações
- Países fechando suas fronteiras.
- Eventos esportivos maciços sendo cancelados.
- Algumas das bandas mais populares do mundo adiam suas turnês.
- E localmente, meus restaurantes e cafeterias favoritos fechando suas portas.

Isso é tudo no nível macro - **mas e o nível micro?**

E quanto a nós como *indivíduos*?

É muito fácil ser pego nas estatísticas globais.

Vemos números como 6.000 mortos e 160.000 casos confirmados (com potencialmente várias ordens de magnitude a mais devido à falta de kits de teste COVID-19 e que algumas pessoas estão optando pela auto-quarentena).

Quando pensamos nesses termos, perdemos de vista a nós mesmos e a nossos entes queridos. Precisamos levar as coisas dia a dia. Precisamos pensar no nível individual para nossa própria saúde mental e sanidade. **Precisamos de espaços seguros onde possamos nos retirar.**

Quando iniciei o PylmageSearch há mais de cinco anos, sabia que seria um espaço seguro. Eu dei o exemplo do que o PylmageSearch deveria se tornar e ainda faço até hoje. Por esse motivo, não permito assédio de *nenhuma* forma ou forma, incluindo, entre outros, racismo, sexismo, xenofobia, elitismo, bullying, etc.

A comunidade PylmageSearch é especial. As pessoas aqui *respeitam os outros* - e se não o fizerem, eu as removo.

Talvez uma das minhas exposições favoritas de caráter humano [aceitável](#) e altruísta tenha acontecido quando eu [dirigi o PylmageConf 2018](#) - os participantes ficaram impressionados com o quão amigável e acolhedora a conferência foi.

Dave Snowden, engenheiro de software e participante do PylmageConf, disse:

PylmageConf foi sem dúvida a conferência mais amigável e acolhedora que eu já estive. O conteúdo técnico também foi ótimo também! Foi um privilégio conhecer e aprender com algumas das pessoas que contribuíram com seu tempo para criar as ferramentas nas quais confiamos para o nosso trabalho (e diversão).

David Stone, doutor em engenharia e professor da Virginia Commonwealth University, compartilhou o seguinte:

Obrigado por montar o PylmageConf. Também concordo que foi a conferência mais amigável da qual participei.

Por que digo tudo isso?

Porque eu sei que você pode estar com medo agora.

Eu sei que você pode estar no seu limite (confie em mim, eu também estou).

E o mais importante, porque eu quero que o PylmageSearch seja o seu espaço seguro.

- Você pode ser um estudante em casa após o término prematuro do semestre, desapontado por sua educação ter sido suspensa.
- Você pode ser um desenvolvedor, totalmente perdido depois que seu local de trabalho trancou suas portas no futuro próximo.

- Você pode ser um pesquisador frustrado por não poder continuar seus experimentos e criar esse novo artigo.
- Você pode ser pai ou mãe, tentando, sem sucesso, conciliar dois filhos e um requisito obrigatório de "trabalhar em casa".

Ou você pode ser como eu - apenas tentando passar o dia aprendendo uma nova habilidade, algoritmo ou técnica.

Recebi vários e-mails dos leitores do PyImageSearch que desejam usar esse tempo de inatividade para estudar Visão de Computador e Aprendizado Profundo, em vez de enlouquecer em suas casas.

Eu respeito isso e quero ajudar e, até certo ponto, ***acredito que é minha obrigação moral ajudar como posso:***

- Para começar, **[você pode aprender](#)** mais de **[350 tutoriais gratuitos no blog do PyImageSearch](#)**. Eu publico um novo tutorial toda segunda-feira às 10h EST.
- Eu categorizei, fiz referência cruzada e compilei esses tutoriais na minha página **["Introdução"](#)**.
- Os tópicos mais populares na página ***"Introdução"*** incluem **["Deep Learning"](#)** e **["Face Applications"](#)**.

Todos esses guias são 100% gratuitos. Use-os para estudar e aprender.

Dito isto, muitos leitores também solicitam que eu faça uma venda em meus livros e cursos. No começo, fiquei um pouco hesitante sobre isso - a *última* coisa que quero é que as pessoas pensem que estou de alguma forma usando o coronavírus como um esquema para "ganhar dinheiro".

Mas a verdade é que ser um pequeno empresário que não é apenas responsável por mim e pela minha família, mas também a vida e a família dos meus colegas de equipe podem ser assustadoras e avassaladoras às vezes - a vida das pessoas, incluindo pequenas empresas, será *destruída* por isso. vírus.

Para esse fim, assim como:

- Bandas e artistas estão oferecendo shows com descontos "apenas online"
- Os restaurantes estão oferecendo entrega em domicílio
- Os treinadores de fitness estão oferecendo sessões de treinamento on-line

... Eu vou estar seguindo o exemplo.

A partir de amanhã, estarei executando uma venda nos livros do PyImageSearch. Esta venda *não tem* fins lucrativos e *certamente* não é planejada (passei o fim de semana inteiro doente, tentando juntar tudo isso).

Em vez disso, é uma venda para ajudar pessoas, como eu (e talvez como você),

que estão lutando para encontrar seu espaço seguro durante essa bagunça. Deixe-me e PyImageSearch se tornar seu refúgio.

Normalmente, faço apenas uma grande venda por ano (Black Friday), mas, considerando quantas pessoas estão solicitando, acredito que é algo que *preciso* fazer para quem deseja usar esse tempo de inatividade para estudar e / ou como uma distração do resto do mundo.

Sinta-se livre para participar ou não. Está tudo bem. Todos nós processamos esses tempos difíceis à nossa maneira.

Mas se você precisar descansar, se precisar de um refúgio, se precisar de um retiro pela educação - eu estarei aqui.

Obrigado e fique seguro.

Sumário

Neste tutorial, você aprendeu como usar o Keras, o TensorFlow e o Deep Learning para treinar um detector COVID-19 automático em um conjunto de dados de imagens de raios-X.

Ainda não existem conjuntos de dados de imagem revisados por pares de alta qualidade para COVID-19, então tivemos que trabalhar com o que tínhamos, a saber, [o repositório GitHub de Joseph Cohen de imagens de raios-X de código aberto](#) :

- **Foram amostradas 25 imagens do conjunto de dados de Cohen**, considerando apenas a visão posterior-posterior (AP) dos casos positivos de COVID-19.
- **Em seguida, amostramos 25 imagens de pacientes saudáveis** usando [o conjunto de dados de imagens de raios-X do tórax de Kaggle \(Pneumonia\)](#).

A partir daí, usamos o Keras e o TensorFlow para treinar um detector COVID-19 capaz de obter **precisão de 90 a 92 %** em nosso conjunto de testes com 100% de sensibilidade e 80% de especificidade (dado nosso conjunto de dados limitado).

Lembre-se de que o detector COVID-19 abordado neste tutorial é apenas para fins educacionais (consulte o meu “Isenção de responsabilidade” na parte superior deste tutorial). Meu objetivo é inspirar os profissionais de aprendizagem profunda, como você, e abrir os olhos para saber como a aprendizagem profunda e a visão computacional podem causar um *grande impacto* no mundo.

Espero que tenham gostado deste post do blog.

Para baixar o código fonte desta publicação (incluindo o modelo de diagnóstico COVID-19 pré-treinado), basta inserir seu endereço de e-mail no formulário abaixo!

