

COVIDclassification

Robert Tedesco

10/3/2020

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(reticulate)  
library(tensorflow)  
library(keras)
```

```
library(imager)
```

```
## Loading required package: magrittr
```

```
##  
## Attaching package: 'imager'
```

```
## The following object is masked from 'package:magrittr':  
##  
##      add
```

```
## The following objects are masked from 'package:stats':  
##  
##      convolve, spectrum
```

```
## The following object is masked from 'package:graphics':  
##  
##      frame
```

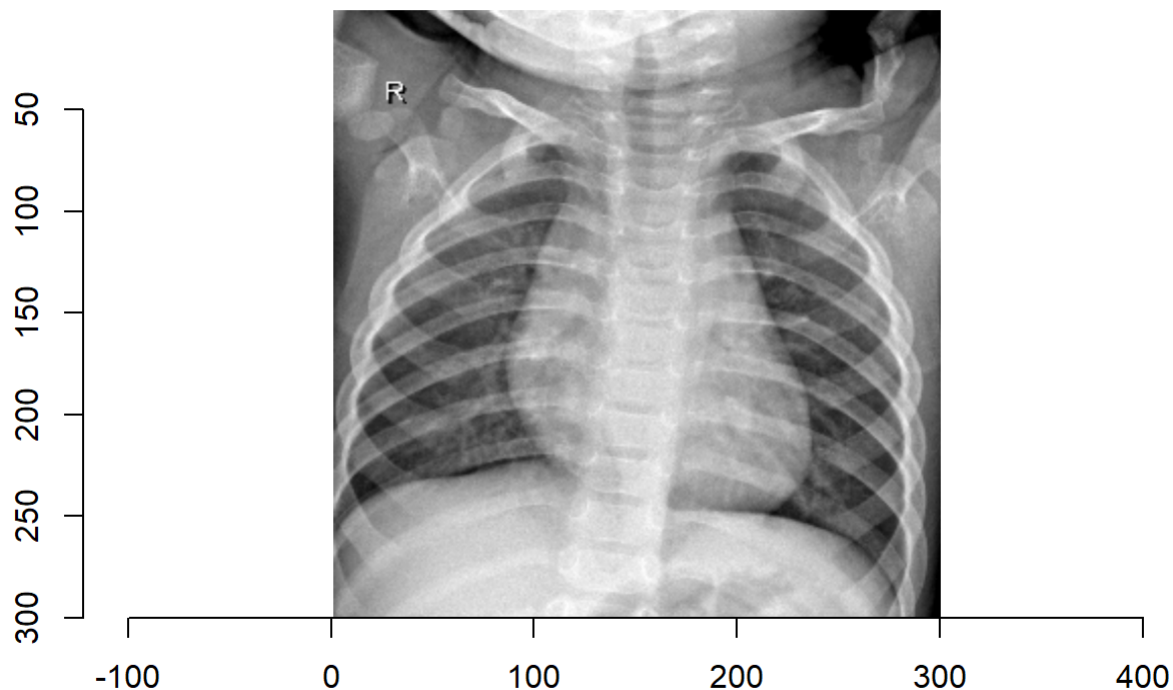
```
## The following object is masked from 'package:base':  
##  
##      save.image
```

```
library(EBImage)
```

```
##  
## Attaching package: 'EBImage'
```

```
## The following objects are masked from 'package:imager':
##
##   channel, dilate, display, erode, resize, watershed
```

```
Normaltrain<-list.files(path="C:\\Users\\rober\\Pictures\\COVIDimages\\Train\\Normal",pattern='j
peg',all.files=T,full.names=T)
Normaltrainpics<-list()
for (i in 1:169) {Normaltrainpics[[i]]<-load.image(Normaltrain[i])}
for (i in 1:169) {Normaltrainpics[[i]]<-resize(Normaltrainpics[[i]],300,300)}
plot(Normaltrainpics[[1]])
```



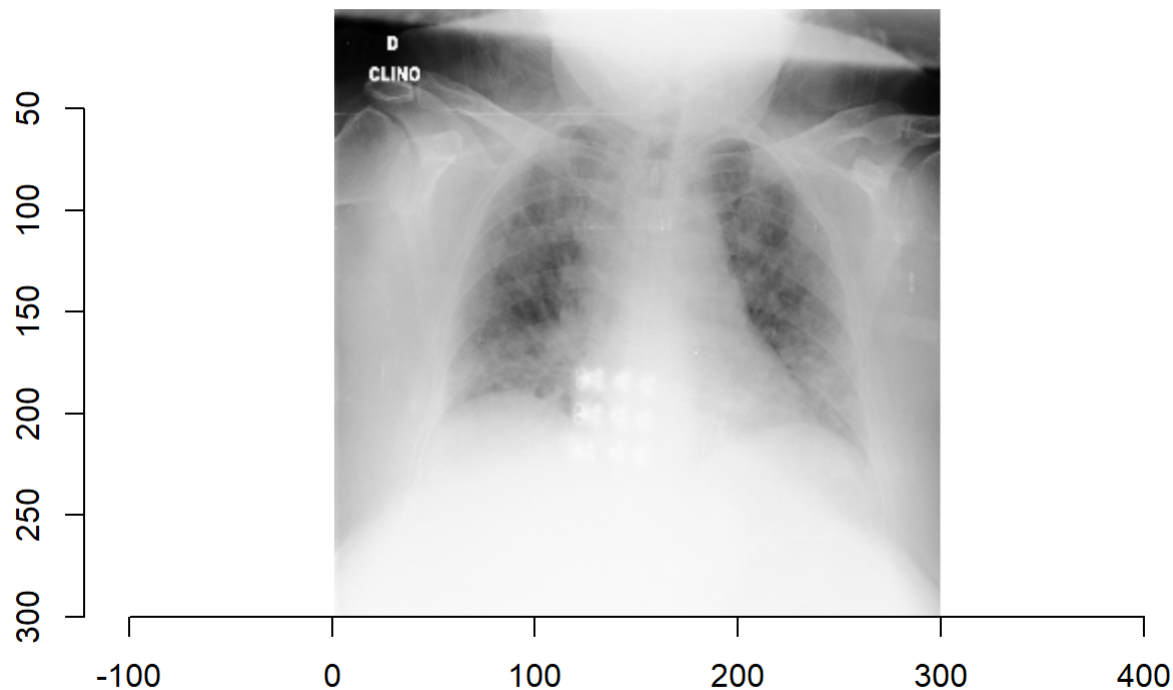
```
for (i in 1:169){
Normaltrainpics[[i]]<-add.color(Normaltrainpics[[i]])}
#All of our Normaltrainpics are grayscale, we have to make them rgb to have dimensions [x,y,1,3]
instead of [x,y,1,1] with add.color.

for (i in 1:169){
Normaltrainpics[[i]]<-array_reshape(Normaltrainpics[[i]],dim=c(300,300,1,3))}
```

```

Covidtrain<-list.files(path="C:\\Users\\rober\\Pictures\\COVIDimages\\Train\\Covid",pattern='jpg',all.files=T,full.names=T,no.. = T)
Covidtrainpics<-list()
for (i in 1:57) {Covidtrainpics[[i]]<-load.image(Covidtrain[i])}
for (i in 1:57) {Covidtrainpics[[i]]<-resize(Covidtrainpics[[i]],300,300)}
plot(Covidtrainpics[[1]])

```

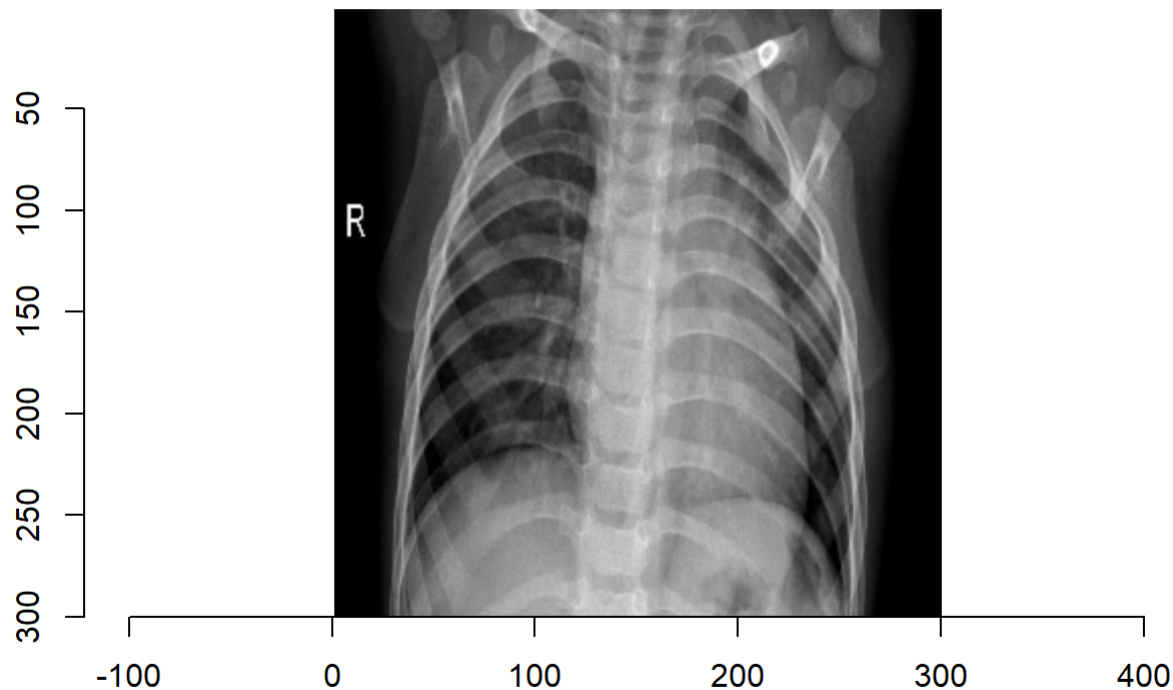


```

for (i in 1:57){ if (spectrum(Covidtrainpics[[i]])==1){
Covidtrainpics[[i]]<-add.color(Covidtrainpics[[i]])}}
#We also make our covid training xray images rgb if they are greyscale.

#Uploading Testing Normal & Covid
Testing<-list.files(path="C:\\Users\\rober\\Pictures\\COVIDimages\\Prediction",pattern='jpeg',all.files=T,full.names=T,no.. = T)
Testpics<-list()
for(i in 1:23) {Testpics[[i]]<-load.image(Testing[i])}
for (i in 1:23) {Testpics[[i]]<-resize(Testpics[[i]],300,300)}
plot(Testpics[[1]])

```



```
for (i in 1:23){ if (spectrum(Testpics[[i]])==1){
Testpics[[i]]<-add.color(Testpics[[i]])}
for (i in 1:23){ if (spectrum(Testpics[[i]])==4){
Testpics[[i]]<-rm.alpha(Testpics[[i]])}
#Our testing pictures include an alpha channel, giving them dimensions [x,y,1,4], rm.alpha turns
them into our desired array dimensions [x,y,1,3]
```

```
#Making arrays to put bind into final train and test set.
for (i in 1:57) {Covidtrainpics[[i]]<-array_reshape(Covidtrainpics[[i]],c(300,300,1,3))}
for (i in 1:23) {Testpics[[i]]<-array_reshape(Testpics[[i]],c(300,300,1,3))}
Trainingx<-NULL
for (i in 1:169){Trainingx<-rbind(Trainingx,Normaltrainpics[[i]])}
for (i in 1:57){Trainingx<-rbind(Trainingx,Covidtrainpics[[i]])}
Testingx<-NULL
for (i in 1:23){Testingx<-rbind(Testingx,Testpics[[i]])}
```

```
#Generating labels and combining accordingly.
normal<-rep(0,169)
covid<-rep(1,57)
Testingy<-c(rep(0,12),rep(1,11))
Trainingy<-c(normal,covid)
#Hot-coding our labels.
trainLabels<-to_categorical(Trainingy)
testLabels<-to_categorical(Testingy)
```

```
#Building model
model<-keras_model_sequential()
model %>%
  layer_dense(units=512,activation='relu',input_shape=c(270000)) %>%
  layer_dense(units=256,activation='relu',)%>%
  layer_dense(units=128,activation='relu',)%>%
  layer_dense(units=2,activation='softmax')
#compile
model %>%
  compile(loss='binary_crossentropy',
          optimizer=optimizer_rmsprop(),
          metrics=c('accuracy'))
summary(model)
```

```
## Model: "sequential"
## _____
## Layer (type)                Output Shape                Param #
## =====
## dense (Dense)                (None, 512)                 138240512
## _____
## dense_1 (Dense)              (None, 256)                 131328
## _____
## dense_2 (Dense)              (None, 128)                 32896
## _____
## dense_3 (Dense)              (None, 2)                   258
## =====
## Total params: 138,404,994
## Trainable params: 138,404,994
## Non-trainable params: 0
## _____
```

```
#fit
history<-model %>%
  fit(Trainingx,
      trainLabels,
      epochs=30,
      batch_size=64,
      validation_split=.2)

#Predict
model %>% evaluate(Testingx,testLabels)
```

```
##      loss  accuracy
## 7.3351321 0.5217391
```

```
pred<-model %>% predict_classes(Testingx)
table(Predicted=pred,Actual=Testingy)
```

| ## | | Actual | |
|--------------|--|--------|----|
| ## Predicted | | 0 | 1 |
| ## | | 0 12 | 11 |