# Covid-19-identification-using-AI

Using AI to predict whether the patient has Covid-19

\*\*\*\*First we'll import all the packages required for setting up our Neural Network\*\*\*\*

# Importing the Keras libraries and packages

from keras.models import Sequential

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

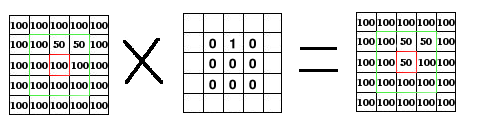
from keras.layers import Dense

# Initialising the CNN

classifier = Sequential()

\*\*\*\*Here our image matrix is multiplied with the Convolution matrix of dimension 3x3.

Relu is the rectifier activation function.\*\*\*\*

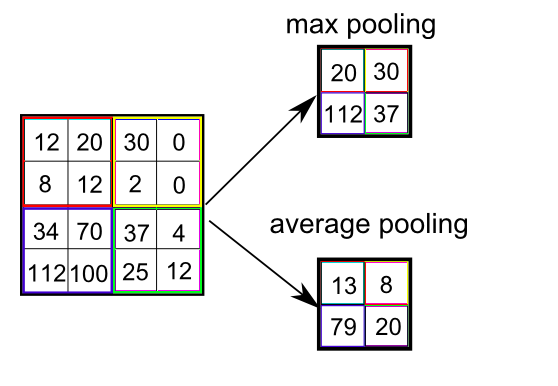


# Step 1 - Convolution

classifier.add(Conv2D(32, (3, 3), input\_shape = (128, 128, 3), activation = 'relu'))

\*\*\*\*Pooling the the process of further extracting important components from the Convolved image matrix to reduce the matrix into more smaller dimension.

Now our convolved image is Pooled using the Max Pooling\*\*\*\*\*



# Step 2 - Pooling

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

\*\*\*NOW WE'LL ADD MULTIPLE CONVOLUTIONAL LAYERS\*\*\*

# Adding a second convolutional layer

classifier.add(Conv2D(32, (3, 3), activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

#Adding a third convolution layer

classifier.add(Conv2D(32, (3, 3), activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

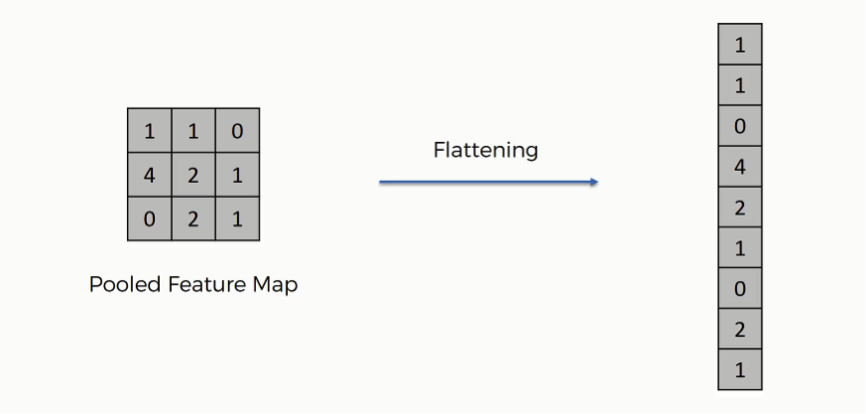
#Adding a fourth convolution layer

classifier.add(Conv2D(32, (3, 3), activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

\*\*\*\*Flattening is the process of reducing the dimension of matrix to single row.

Hence the term "FLATTENING".This is the kind of image or we can say matrix that the machine can understand\*\*\*\*



# Step 3 - Flattening

classifier.add(Flatten())

\*\*\*\*The flattened image is further fed to our Neural Network as input.Now several hidden layers will be present inside the network between input and output which will adjust the weights for image classification\*\*\*\*

# Step 4 - Full connection

classifier.add(Dense(units = 128, activation = 'relu'))

classifier.add(Dense(units = 1, activation = 'sigmoid')) \*\*\*\*Sigmoid activation function is used at the output

\*\*\*\*Compile the CNN here different parameters can be tweaked and tested\*\*\*\*

# Compiling the CNN

classifier.compile(optimizer = 'adam', loss = 'binary\_crossentropy', metrics = ['accuracy'])

\*\*\*\* The following code snippet was taken from keras image preprocessing section.There are different algorithms for different situation or we can say different type of dataset.The following is perfect for out dataset\*\*\*\*



# Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale = 1./255,

shear\_range = 0.2,

zoom\_range = 0.2,

horizontal\_flip = True)

test\_datagen = ImageDataGenerator(rescale =1./255)

training\_set = train\_datagen.flow\_from\_directory('Training Dataset path',

target\_size = (128, 128),

batch\_size = 16

,class\_mode = 'binary')

test\_set = test\_datagen.flow\_from\_directory('Test Dataset path',

target\_size = (128, 128)

,batch\_size = 32,

class\_mode = 'binary')

\*\*\*\*Training set images = 5200,Test set images = 620, Epochs is the number of times the training will repead i.e. in following situation if the epochs is 90,thn the training set is fed 90 times (5200x620 images)\*\*\*\*

classifier.fit\_generator(training\_set,

steps\_per\_epoch = 5200,

epochs = 90,

validation\_data = test\_set,

validation\_steps =620)

\*\*\*\*Now we'll make predictions on new X-Ray scans of patients whether they've pneumonia or not.\*\*\*\*

# Part 3 - Making new predictions

import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('chest\_xray/val/NORMAL/normal.jpeg', target\_size = (64,64))

\*\*\*\* Our test image is loaded into an array and Neural Network predicts the result\*\*\*\*

test\_image = image.img\_to\_array(test\_image)

test\_image = np.expand\_dims(test\_image, axis = 0)

result = classifier.predict(test\_image)

training\_set.class\_indices

if result[0][0] == 1:

prediction = 'Normal'

else:

prediction = 'Pneumonia'