**COVID-19: IA para detectar el COVID-19 con radiografías de tórax**



**José Brayan Isla Pérez**

**30/06/2020**

**UNIVERSIDAD PRIVADA DOMINGO SAVIO**

**Informe Final**

**INTRODUCCIÓN**

El sistema está basado en **COVIDx**: una base de datos que hace una semana contaba con 5.941 imágenes de radiografías de tórax de 2.839 pacientes de fuentes públicas. Este lunes sumaban ya **16.756 radiografías de tórax** de **13.645 personas**con diferentes afecciones pulmonares, como infecciones bacterianas, virales, COVID-19 y otros virus.

**OBJETIVO GENERAL**

El objetico general en la lucha contra esta pandemia es la**detección efectiva de los pacientes infectados**. Las imágenes radiológicas obtenidas mediante radiografía de tórax son para este propósito un elemento clave.

**DESCRIPCION DE LA TOPOLOGIA DE LA RED NEURONAL**

COVID-Net es un diseño de red neural convolucional profunda (un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico) adaptado para la detección de casos de COVID-19 a partir de imágenes de radiografía de tórax. Es de código abierto y está disponible para el público en general.

Los científicos han publicado dos modelos basados en el conjunto de datos anterior: **COVIDNet-Grande y COVIDNet-Pequeño.** El primero proporciona una mayor sensibilidad para la detección de COVID-19, mientras que el segundo logra un mayor equilibrio entre la eficiencia de los cálculos y el rendimiento.

**CONCLUSIONES**

Los algoritmos de IA aprenden, como los médicos, a través del estudio y el análisis de casos. Pueden abstraer patrones, inferir los hallazgos clave entre una y otra enfermedad y, de esa manera, predecir o clasificar problemas médicos. Por ejemplo, ante una cantidad de imágenes de radiografías de tórax normales y un conjunto de radiografías de pacientes con neumonía, los médicos y la IA aprenden a distinguirlas por sus características, como la presencia de consolidación o manchas en la radiografía, entre otras

**Direccion de dataset**

<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>

**CODIGO**

#---------IMPORTANDO LIBRERIAS NECESARIAS PARA EL PROYECTO

import os

import tensorflow as tf

from tensorflow import keras

import numpy as np

import cv2

from tqdm import tqdm

from keras.utils import np\_utils

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

#----------------GUARDANDO LAS PATHS(DIRECCIONES) EN VARIABLES

dir1='../input/covid19-radiography-database/COVID-19 Radiography Database/NORMAL'

dir2='../input/covid19-radiography-database/COVID-19 Radiography Database/COVID-19'

labels=[]

for i in os.listdir('../input/covid19-radiography-database/COVID-19 Radiography Database/NORMAL'):

labels.append(0)

for i in os.listdir('../input/covid19-radiography-database/COVID-19 Radiography Database/COVID-19'):

labels.append(1)

imagen=[]

for i in tqdm(os.listdir(dir1)):

f=cv2.imread(os.path.join(dir1,i))

f=cv2.resize(f,(100,100))

imagen.append(f)

for i in tqdm(os.listdir(dir2)):

f=cv2.imread(os.path.join(dir2,i))

f=cv2.resize(f,(100,100))

imagen.append(f)

x=np.array(imagen)

y=np.array(labels)

x=(x/255.0)

y=np\_utils.to\_categorical(y)

x=x.reshape(1560,30000)

model=keras.Sequential()

x\_train,x\_test,y\_train,y\_test= train\_test\_split(x,y)

model.add(keras.layers.Dense(300,activation='relu',input\_dim=x\_train.shape[1]))

model.add(keras.layers.Dense(200,activation='relu'))

model.add(keras.layers.Dense(200,activation='relu'))

model.add(keras.layers.Dense(200,activation='relu'))

model.add(keras.layers.Dense(200,activation='relu'))

model.add(keras.layers.Dense(2,activation='softmax'))

model.compile(loss='categorical\_crossentropy',optimizer='adam')

model.fit(x\_train,y\_train,epochs=50)

index=200

plt.imshow(x\_test[index].reshape(100,100,3))

plt.show()

p=model.predict(x\_test[index].reshape(1,30000))

np.argmax(p)

np.argmax(y\_test[index])

model.save('direccion donde guardara el entrenamiento del proyecto')

model.load('direccion donde guardado el entrenamiento del proyecto para poder cargarlo nuevamente')