# SOURCE CODE

# CODE:

from PIL import Image

import pandas as pd

import numpy as np

import os

import tensorflow as tf

import keras

import warnings

warnings.filterwarnings("ignore")

def cnn\_model(path\_loc):

    import tensorflow as tf

    import keras

    from keras.models import Sequential

    from keras.layers import Convolution2D

    from keras.layers import MaxPool2D

    from keras.layers import Flatten

    from keras.layers import Dense

    # Initializing CNN

    classifier = Sequential()

    # Step 1 : convolution

    classifier.add(Convolution2D(32,3,3,input\_shape=(128,128,3),activation='relu'))

    # Step 2 : Pooling

    classifier.add(MaxPool2D(pool\_size=(2,2)))

    # Adding second convolution layer.

    classifier.add(Convolution2D(32,3,3,activation='relu'))

    classifier.add(MaxPool2D(pool\_size=(2,2)))

    # step 3 : flattening

    classifier.add(Flatten())

    # step 4 : Full connection

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

    classifier.add(Dense(units=2,activation='softmax'))

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

    # fitting CNN to the images

    from keras.preprocessing.image import ImageDataGenerator

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

                                   shear\_range = 0.2,

                                   zoom\_range = 0.2,

                                   horizontal\_flip = True)

    test\_datagen = ImageDataGenerator(rescale=1./255) # test data don't need much image agumentation.

    import os

    os.chdir(path\_loc) # train and test path location

    training\_set = train\_datagen.flow\_from\_directory('/content/drive/MyDrive/medicine1/DATASET',

                                                 target\_size = (128, 128),

                                                 batch\_size = 12,

                                                 class\_mode = 'categorical')

    test\_set = test\_datagen.flow\_from\_directory('/content/drive/MyDrive/medicine1/DATASET',

                                            target\_size = (128, 128),

                                            batch\_size = 12,

                                           class\_mode = 'categorical')

    # buliding model and performing model validation simultaneously

    history=classifier.fit\_generator(training\_set,

                         steps\_per\_epoch = 12, # number of images to be taken from training data set.

                         epochs  = 15,

                         validation\_data = test\_set,

                         validation\_steps = 10) # number of images to be taken from test data set.

    return (training\_set ,classifier,history)

training\_set ,classifier,history= cnn\_model('/content/drive/MyDrive/medicine1/DATASET')

import numpy as np

from keras.preprocessing import image

%matplotlib inline

test\_image = image.load\_img('/content/drive/MyDrive/COLOR\_POP (2).jpg', target\_size = (128, 128))

test\_image.show()

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

test\_image

np.shape(test\_image)

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

test\_image

np.shape(test\_image)

test = training\_set.class\_indices

predict\_x=classifier.predict(test\_image)

classes\_x=np.argmax(predict\_x,axis=1)

classes\_x

#predictions = (classifier.predict(test\_image) > 0.5).astype("int32")

for key, value in test.items():

        if value == classes\_x:

            print("The Predected image is", key)