#Parametres

#-----------------------------------

BATCH\_SIZE = 54

EPOCHS = 200

def createCNN():

n\_classes = 2

cnn = Sequential()

cnn.add(Conv1D(128, kernel\_size=80 ,strides=8,input\_shape=(5400,1)))

cnn.add(BatchNormalization())

cnn.add(MaxPooling1D(pool\_size=4))

cnn.add(Activation('relu'))

cnn.add(Conv1D(256,kernel\_size=4))

cnn.add(BatchNormalization())

cnn.add(MaxPooling1D(pool\_size=4))

cnn.add(Activation('relu'))

cnn.add(Conv1D(256,kernel\_size=4))

cnn.add(BatchNormalization())

cnn.add(MaxPooling1D(pool\_size=4))

cnn.add(Activation('relu'))

cnn.add(Conv1D(512,kernel\_size=2))

cnn.add(BatchNormalization())

cnn.add(MaxPooling1D(pool\_size=4))

cnn.add(Activation('relu'))

cnn.add(AveragePooling1D(pool\_size=2))

cnn.add(Flatten())

cnn.add(Dropout(0.6))

cnn.add(Dense(n\_classes,kernel\_regularizer=keras.regularizers.l2(0.001),activation='softmax'))

cnn.compile(optimizer=tensorflow.keras.optimizers.Adamax(lr = 6.5e-6), loss='binary\_crossentropy', metrics=['accuracy',tensorflow.keras.metrics.Recall(name ='recall')])

return cnn

createCNN().summary()