from keras.models import Model

from keras.layers import Input

from keras.layers import Conv2D

from keras.layers import MaxPool2D

from keras.layers import UpSampling2D

from keras.layers import Dropout

from keras.layers import concatenate

from all\_params import IMG\_ROWS, IMG\_COLS

def get\_model(input\_shape=(IMG\_ROWS, IMG\_COLS, 1), train=True):

layers = {}

layers['inputs'] = Input(shape=input\_shape, name='inputs')

layers['conv1\_1'] = Conv2D(32, (3, 3), padding='same', activation='relu', name='conv1\_1')(layers['inputs'])

layers['conv1\_2'] = Conv2D(32, (3, 3), padding='same', activation='relu', name='conv1\_2')(layers['conv1\_1'])

layers['pool\_1'] = MaxPool2D(pool\_size=(2, 2), name='pool\_1')(layers['conv1\_2'])

if train == True:

layers['dropout\_1'] = Dropout(0.25, name='dropout\_1')(layers['pool\_1'])

layers['conv2\_1'] = Conv2D(64, (3, 3), padding='same', activation='relu', name='conv2\_1')(layers['dropout\_1'])

else:

layers['conv2\_1'] = Conv2D(64, (3, 3), padding='same', activation='relu', name='conv2\_1')(layers['pool\_1'])

layers['conv2\_2'] = Conv2D(64, (3, 3), padding='same', activation='relu', name='conv2\_2')(layers['conv2\_1'])

layers['pool\_2'] = MaxPool2D(pool\_size=(2, 2), name='pool\_2')(layers['conv2\_2'])

if train == True:

layers['dropout\_2'] = Dropout(0.25, name='dropout\_2')(layers['pool\_2'])

layers['conv3\_1'] = Conv2D(128, (3, 3), padding='same', activation='relu', name='conv3\_1')(layers['dropout\_2'])

else:

layers['conv3\_1'] = Conv2D(128, (3, 3), padding='same', activation='relu', name='conv3\_1')(layers['pool\_2'])

layers['conv3\_2'] = Conv2D(128, (3, 3), padding='same', activation='relu', name='conv3\_2')(layers['conv3\_1'])

layers['pool\_3'] = MaxPool2D(pool\_size=(2, 2), name='pool\_3')(layers['conv3\_2'])

if train == True:

layers['dropout\_3'] = Dropout(0.25, name='dropout\_3')(layers['pool\_3'])

layers['conv4\_1'] = Conv2D(256, (3, 3), padding='same', activation='relu', name='conv4\_1')(layers['dropout\_3'])

else:

layers['conv4\_1'] = Conv2D(256, (3, 3), padding='same', activation='relu', name='conv4\_1')(layers['pool\_3'])

layers['conv4\_2'] = Conv2D(256, (3, 3), padding='same', activation='relu', name='conv4\_2')(layers['conv4\_1'])

layers['pool\_4'] = MaxPool2D(pool\_size=(2, 2), name='pool\_4')(layers['conv4\_2'])

if train == True:

layers['dropout\_4'] = Dropout(0.25, name='dropout\_4')(layers['pool\_4'])

layers['conv5\_1'] = Conv2D(512, (3, 3), padding='same', activation='relu', name='conv5\_1')(layers['dropout\_4'])

else:

layers['conv5\_1'] = Conv2D(512, (3, 3), padding='same', activation='relu', name='conv5\_1')(layers['pool\_4'])

layers['conv5\_2'] = Conv2D(512, (3, 3), padding='same', activation='relu', name='conv5\_2')(layers['conv5\_1'])

layers['upsample\_1'] = UpSampling2D(size=(2, 2), name='upsample\_1')(layers['conv5\_2'])

layers['concat\_1'] = concatenate([layers['upsample\_1'], layers['conv4\_2']], name='concat\_1')

layers['conv6\_1'] = Conv2D(256, (3, 3), padding='same', activation='relu', name='conv6\_1')(layers['concat\_1'])

layers['conv6\_2'] = Conv2D(256, (3, 3), padding='same', activation='relu', name='conv6\_2')(layers['conv6\_1'])

if train == True:

layers['dropout\_6'] = Dropout(0.25, name='dropout\_6')(layers['conv6\_2'])

layers['upsample\_2'] = UpSampling2D(size=(2, 2), name='upsample\_2')(layers['dropout\_6'])

else:

layers['upsample\_2'] = UpSampling2D(size=(2, 2), name='upsample\_2')(layers['conv6\_2'])

layers['concat\_2'] = concatenate([layers['upsample\_2'], layers['conv3\_2']], name='concat\_2')

layers['conv7\_1'] = Conv2D(128, (3, 3), padding='same', activation='relu', name='conv7\_1')(layers['concat\_2'])

layers['conv7\_2'] = Conv2D(128, (3, 3), padding='same', activation='relu', name='conv7\_2')(layers['conv7\_1'])

if train == True:

layers['dropout\_7'] = Dropout(0.25, name='dropout\_7')(layers['conv7\_2'])

layers['upsample\_3'] = UpSampling2D(size=(2, 2), name='upsample\_3')(layers['dropout\_7'])

else:

layers['upsample\_3'] = UpSampling2D(size=(2, 2), name='upsample\_3')(layers['conv7\_2'])

layers['concat\_3'] = concatenate([layers['upsample\_3'], layers['conv2\_2']], name='concat\_3')

layers['conv8\_1'] = Conv2D(64, (3, 3), padding='same', activation='relu', name='conv8\_1')(layers['concat\_3'])

layers['conv8\_2'] = Conv2D(64, (3, 3), padding='same', activation='relu', name='conv8\_2')(layers['conv8\_1'])

if train == True:

layers['dropout\_8'] = Dropout(0.25, name='dropout\_8')(layers['conv8\_2'])

layers['upsample\_4'] = UpSampling2D(size=(2, 2), name='upsample\_4')(layers['dropout\_8'])

else:

layers['upsample\_4'] = UpSampling2D(size=(2, 2), name='upsample\_4')(layers['conv8\_2'])

layers['concat\_4'] = concatenate([layers['upsample\_4'], layers['conv1\_2']], name='concat\_4')

layers['conv9\_1'] = Conv2D(32, (3, 3), padding='same', activation='relu', name='conv9\_1')(layers['concat\_4'])

layers['conv9\_2'] = Conv2D(32, (3, 3), padding='same', activation='relu', name='conv9\_2')(layers['conv9\_1'])

if train == True:

layers['dropout\_9'] = Dropout(0.25, name='dropout\_9')(layers['conv9\_2'])

layers['outputs'] = Conv2D(1, (1, 1), activation='sigmoid', name='outputs')(layers['dropout\_9'])

else:

layers['outputs'] = Conv2D(1, (1, 1), activation='sigmoid', name='outputs')(layers['conv9\_2'])

model = Model(inputs=layers['inputs'], outputs=layers['outputs'])

return model