# hwalsuklee\_tensorflow-mnist-cnn\_cnn\_model.py

# Some code was borrowed from https://github.com/petewarden/tensorflow\_makefile/blob/master/tensorflow/models/image/mnist/convolutional.py  
from \_\_future\_\_ import absolute\_import  
from \_\_future\_\_ import division  
from \_\_future\_\_ import print\_function  
  
import tensorflow as tf  
import tensorflow.contrib.slim as slim  
  
  
# Create model of CNN with slim api  
def CNN(inputs, is\_training=True):  
 batch\_norm\_params = {'is\_training': is\_training, 'decay': 0.9, 'updates\_collections': None}  
 with slim.arg\_scope([slim.conv2d, slim.fully\_connected],  
 normalizer\_fn=slim.batch\_norm,  
 normalizer\_params=batch\_norm\_params):  
 x = tf.reshape(inputs, [-1, 28, 28, 1])  
  
 # For slim.conv2d, default argument values are like  
 # normalizer\_fn = None, normalizer\_params = None, <== slim.arg\_scope changes these arguments  
 # padding='SAME', activation\_fn=nn.relu,  
 # weights\_initializer = initializers.xavier\_initializer(),  
 # biases\_initializer = init\_ops.zeros\_initializer,  
 net = slim.conv2d(x, 32, [5, 5], scope='conv1')  
 net = slim.max\_pool2d(net, [2, 2], scope='pool1')  
 net = slim.conv2d(net, 64, [5, 5], scope='conv2')  
 net = slim.max\_pool2d(net, [2, 2], scope='pool2')  
 net = slim.flatten(net, scope='flatten3')  
  
 # For slim.fully\_connected, default argument values are like  
 # activation\_fn = nn.relu,  
 # normalizer\_fn = None, normalizer\_params = None, <== slim.arg\_scope changes these arguments  
 # weights\_initializer = initializers.xavier\_initializer(),  
 # biases\_initializer = init\_ops.zeros\_initializer,  
 net = slim.fully\_connected(net, 1024, scope='fc3')  
 net = slim.dropout(net, is\_training=is\_training, scope='dropout3') # 0.5 by default  
 outputs = slim.fully\_connected(net, 10, activation\_fn=None, normalizer\_fn=None, scope='fco')  
 return outputs