Sol)

|  |
| --- |
| def separable\_conv(input, c\_o, k\_s, stride, scope):  with slim.arg\_scope([slim.batch\_norm],  decay=0.999,  fused=True,  is\_training=\_trainable,  activation\_fn=tf.nn.relu6):  output = slim.separable\_convolution2d(input,  num\_outputs=None,  stride=stride,  trainable=\_trainable,  depth\_multiplier=1.0,  kernel\_size= **[k\_s, k\_s],**  weights\_initializer=\_init\_xavier,  weights\_regularizer=\_l2\_regularizer\_00004,  biases\_initializer=None,  scope=scope + '\_depthwise')  output = slim.convolution2d(output,  c\_o,  stride=1,  kernel\_size= **[1, 1],**  weights\_initializer=\_init\_xavier,  biases\_initializer=\_init\_zero,  normalizer\_fn=slim.batch\_norm,  trainable=\_trainable,  weights\_regularizer=None,  scope=scope + '\_pointwise')  return output |