

```
z3_solution_model = {
    # Operator Parameters
    'outch': 128,    # (out_channels / filters)
    'k_0': 5,        # (kernel_size)
    's_0': 1,        # (stride)
    'p_0': 0,        # (padding)
    'd_0': 1,        # (dilation)
    'g': 1,          # (groups, default is 1)
    # Input Tensor Shape
    'n': 1,          # (batch_size)
    'inch': 64,      # (in_channels)
    'h_in': 128,     # (input height)
    'w_in': 128      # (input width) }
```

```
import torch
conv_layer = torch.nn.Conv2d(
    in_channels = 64,    # from model['inch']
    out_channels = 128,  # from model['outch']
    kernel_size = 5,     # from model['k_0']
    stride = 1,          # from model['s_0']
    padding = 0,          # from model['p_0']
    dilation = 1         # from model['d_0']
).cuda()
x = torch.randn(1, 64, 128, 128).cuda()
output = conv_layer(x)
```

```
import tensorflow as tf
from keras.layers import Conv2D
conv_layer = Conv2D(
    filters = 128,      # from model['outch']
    kernel_size = 5,    # from model['k_0']
    strides = 1,         # from model['s_0']
    padding = 'valid',  # from model['p_0']
    data_format = 'channels_first'
)
x = tf.random.normal([1, 64, 128, 128])
output = conv_layer(x)
```

```
import paddle as pdl
conv_layer = pdl.nn.Conv2D(
    in_channels = 64,    # from model['inch']
    out_channels = 128,  # from model['outch']
    kernel_size = 5,     # from model['k_0']
    stride = 1,          # from model['s_0']
    padding = 0,          # from model['p_0']
    dilation = 1         # from model['d_0']
)
x = pdl.randn([1, 64, 128, 128])
output = conv_layer(x)
```