Model:

class **SwinTranformerIR**(**nn**.**Module**):

def **\_\_init\_\_**(self, backbone, input\_channels):

**super**(**SwinTranformerIR**, self).**\_\_init\_\_**()

self.initial\_conv = **nn**.**Conv2d**(input\_channels, 3, kernel\_size=3, padding=1)

self.backbone = backbone

self.decoder = **nn**.**Sequential**(

**UNetDecoderBlock**(1024, 512, 1), *# Input: [16, 1024, 7, 7] Output: [16, 512, 14, 14]*

**UNetDecoderBlock**(512, 256, 2), *# Input: [16, 512, 14, 14] Output: [16, 256, 28, 28]*

**UNetDecoderBlock**(256, 128, 3), *# Input: [16, 256, 28, 28] Output: [16, 128, 56, 56]*

**UNetDecoderBlock**(128, 64, 4), *# Input: [16, 128, 56, 56] Output: [16, 64, 112, 112]*

**UNetDecoderBlock**(64, 32, 5), *# Input: [16, 64, 112, 112] Output: [16, 32, 224, 224]*

**nn**.**Conv2d**(32, 1, kernel\_size=1), *# Input: [16, 32, 224, 224]Output: [16, 1, 224, 224]*

**nn**.**Sigmoid**()

)

self.**\_initialize\_weights**()

class **UNetDecoderBlock**(**nn**.**Module**):

def **\_\_init\_\_**(self, in\_channels, out\_channels, block\_index):

**super**(**UNetDecoderBlock**, self).**\_\_init\_\_**()

self.block\_index = block\_index

self.conv1 = **nn**.**Conv2d**(in\_channels, out\_channels, kernel\_size=3, padding=1)

self.relu = **nn**.**ReLU**(inplace=True)

self.conv2 = **nn**.**Conv2d**(out\_channels, out\_channels, kernel\_size=3, padding=1)

self.transconv = **nn**.**ConvTranspose2d**(out\_channels, out\_channels, kernel\_size=2, stride=2)