

Efficient Models for Real-time Person Segmentation on Mobile Phones

European Signal Processing Conference 2021

Supplementary Material

Julian Strohmayer

Jakob Knapp

Martin Kampel

Computer Vision Lab, TU Wien, Vienna, Austria

23 August 2021

1 MobSegL Network Architecture

Decoder Architecture

Stage	Input Shape	Out Channels	Kernel	SE-Block	Non-Linearity
1	$112 \times 112 \times 32$	16	3×3	✓	ReLU
2	$56 \times 56 \times 48$	16	3×3	-	ReLU
3	$28 \times 28 \times 80$	24	3×3	-	ReLU
4	$14 \times 14 \times 224$	40	5×5	✓	h-swish
5	$7 \times 7 \times 960$	112	5×5	✓	h-swish

Table 1: Bottleneck block parameters of the MobSegL decoder.

PyTorch print

```
MobSegL(  
  (encoder): MobileNetv3_Large(  
    (initial_conv): StandardConv(  
      (conv): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (bn): BatchNorm2d(16, eps=1e-05, momentum=0.01, affine=True, track_running_stats=True)  
      (se_block): Identity()  
    )  
    (bneck1): Residual_Bottleneck(  
      (bneck): Mobile_Inverted_Bottleneck(  
        (activation): ReLU6()  
        (expansion): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1))  
        (bn1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (dwise): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=16)  
        (bn2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (se_block): Identity()  
        (project): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1))  
        (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      )  
    )  
    (bneck2): Residual_Bottleneck(  
      (bneck): Mobile_Inverted_Bottleneck(  
        (activation): ReLU6()  
        (expansion): Conv2d(16, 64, kernel_size=(1, 1), stride=(1, 1))  
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (dwise): Conv2d(64, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=64)  
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (se_block): Identity()  
        (project): Conv2d(64, 24, kernel_size=(1, 1), stride=(1, 1))  
        (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      )  
    )  
    (bneck3): Residual_Bottleneck(  
      (bneck): Mobile_Inverted_Bottleneck(  
        (activation): ReLU6()  
        (expansion): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1))  
        (bn1): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (dwise): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=72)  
        (bn2): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (se_block): Identity()  
        (project): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1))  
        (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      )  
    )  
    (bneck4): Residual_Bottleneck(  
      (bneck): Mobile_Inverted_Bottleneck(  

```

```

(activation): ReLU6()
(expansion): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1))
(bn1): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(dwise): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=72)
(bn2): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=72, out_features=18, bias=True)
  (fc2): Linear(in_features=18, out_features=72, bias=True)
)
(project): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1))
(bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(bneck5): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): ReLU6()
    (expansion): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=120)
    (bn2): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=120, out_features=30, bias=True)
      (fc2): Linear(in_features=30, out_features=120, bias=True)
    )
    (project): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(bneck6): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): ReLU6()
    (expansion): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=120)
    (bn2): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=120, out_features=30, bias=True)
      (fc2): Linear(in_features=30, out_features=120, bias=True)
    )
    (project): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(bneck7): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=240)
    (bn2): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(bneck8): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(80, 200, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(200, 200, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=200)
    (bn2): BatchNorm2d(200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(200, 80, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)

```

```

)
)
(bneck9): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(184, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=184)
    (bn2): BatchNorm2d(184, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck10): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(184, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=184)
    (bn2): BatchNorm2d(184, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck11): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480)
    (bn2): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=480, out_features=120, bias=True)
      (fc2): Linear(in_features=120, out_features=480, bias=True)
    )
    (project): Conv2d(480, 112, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck12): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(672, 672, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=672)
    (bn2): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=672, out_features=168, bias=True)
      (fc2): Linear(in_features=168, out_features=672, bias=True)
    )
    (project): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck13): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(672, 672, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=672)
    (bn2): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=672, out_features=168, bias=True)

```

```

        (fc2): Linear(in_features=168, out_features=672, bias=True)
    )
    (project): Conv2d(672, 160, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(bneck14): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=960)
    (bn2): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=960, out_features=240, bias=True)
      (fc2): Linear(in_features=240, out_features=960, bias=True)
    )
    (project): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck15): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=960)
    (bn2): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=960, out_features=240, bias=True)
      (fc2): Linear(in_features=240, out_features=960, bias=True)
    )
    (project): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(final_conv): StandardConv(
  (conv): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1))
  (bn): BatchNorm2d(960, eps=1e-05, momentum=0.01, affine=True, track_running_stats=True)
  (se_block): Identity()
)
)
(decoder): Bottleneck_Decoder(
  (bnecks): ListModule(
    (0): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): Hardswish()
        (expansion): Conv2d(960, 1920, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(1920, 1920, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=1920)
        (bn2): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): SEBlock(
          (global_pool): AdaptiveAvgPool2d(output_size=1)
          (fc1): Linear(in_features=1920, out_features=480, bias=True)
          (fc2): Linear(in_features=480, out_features=1920, bias=True)
        )
        (project): Conv2d(1920, 112, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): Hardswish()
        (expansion): Conv2d(224, 448, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(448, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(448, 448, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=448)
      )
    )
  )
)

```

```

    (bn2): BatchNorm2d(448, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=448, out_features=112, bias=True)
      (fc2): Linear(in_features=112, out_features=448, bias=True)
    )
    (project): Conv2d(448, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(2): Compression_Bottleneck_Single(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): ReLU6()
    (expansion): Conv2d(80, 160, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwconv): Conv2d(160, 160, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=160)
    (bn2): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(160, 24, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(3): Compression_Bottleneck_Single(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): ReLU6()
    (expansion): Conv2d(48, 96, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwconv): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (bn2): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): Identity()
    (project): Conv2d(96, 16, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(4): Compression_Bottleneck_Single(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): ReLU6()
    (expansion): Conv2d(32, 64, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwconv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=64)
    (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=64, out_features=16, bias=True)
      (fc2): Linear(in_features=16, out_features=64, bias=True)
    )
    (project): Conv2d(64, 16, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
)
(act): ReLU6(inplace=True)
(label): Conv2d(16, 1, kernel_size=(1, 1), stride=(1, 1))
)
)

```

2 MobSegS Network Architecture

Decoder Architecture

Stage	Input Shape	Out Channels	Kernel	SE-Block	Non-Linearity
1	$112 \times 112 \times 32$	16	3×3	✓	ReLU
2	$56 \times 56 \times 32$	16	3×3	-	ReLU
3	$28 \times 28 \times 48$	16	3×3	-	ReLU
4	$14 \times 14 \times 96$	24	5×5	✓	h-swish
5	$7 \times 7 \times 576$	48	5×5	✓	h-swish

Table 2: Bottleneck block parameters of the MobSegS decoder.

PyTorch print

```
MobSegS(
  (encoder): MobileNetv3_Small(
    (initial_conv): StandardConv(
      (conv): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
      (bn): BatchNorm2d(16, eps=1e-05, momentum=0.01, affine=True, track_running_stats=True)
      (se_block): Identity()
    )
    (bneck1): Residual_Bottleneck(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(16, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=16)
        (bn2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): SEBlock(
          (global_pool): AdaptiveAvgPool2d(output_size=1)
          (fc1): Linear(in_features=16, out_features=4, bias=True)
          (fc2): Linear(in_features=4, out_features=16, bias=True)
        )
        (project): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (bneck2): Residual_Bottleneck(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(16, 72, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(72, 72, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=72)
        (bn2): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): Identity()
        (project): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (bneck3): Residual_Bottleneck(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(24, 88, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(88, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(88, 88, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=88)
        (bn2): BatchNorm2d(88, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): Identity()
        (project): Conv2d(88, 24, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
)
```

```

)
)
(bneck4): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(24, 96, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(96, 96, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=96)
    (bn2): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=96, out_features=24, bias=True)
      (fc2): Linear(in_features=24, out_features=96, bias=True)
    )
    (project): Conv2d(96, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck5): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=240)
    (bn2): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=240, out_features=60, bias=True)
      (fc2): Linear(in_features=60, out_features=240, bias=True)
    )
    (project): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck6): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=240)
    (bn2): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=240, out_features=60, bias=True)
      (fc2): Linear(in_features=60, out_features=240, bias=True)
    )
    (project): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck7): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=120)
    (bn2): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=120, out_features=30, bias=True)
      (fc2): Linear(in_features=30, out_features=120, bias=True)
    )
    (project): Conv2d(120, 48, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(bneck8): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(

```



```

(activation): Hardswish()
(expansion): Conv2d(48, 144, kernel_size=(1, 1), stride=(1, 1))
(bn1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(dwise): Conv2d(144, 144, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=144)
(bn2): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=144, out_features=36, bias=True)
  (fc2): Linear(in_features=36, out_features=144, bias=True)
)
(project): Conv2d(144, 48, kernel_size=(1, 1), stride=(1, 1))
(bn3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(bneck9): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(48, 288, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(288, 288, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=288)
    (bn2): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=288, out_features=72, bias=True)
      (fc2): Linear(in_features=72, out_features=288, bias=True)
    )
    (project): Conv2d(288, 96, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(bneck10): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=576)
    (bn2): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=576, out_features=144, bias=True)
      (fc2): Linear(in_features=144, out_features=576, bias=True)
    )
    (project): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(bneck11): Residual_Bottleneck(
  (bneck): Mobile_Inverted_Bottleneck(
    (activation): Hardswish()
    (expansion): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1))
    (bn1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (dwise): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=576)
    (bn2): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=576, out_features=144, bias=True)
      (fc2): Linear(in_features=144, out_features=576, bias=True)
    )
    (project): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(final_conv): StandardConv(
  (conv): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1))
  (bn): BatchNorm2d(576, eps=1e-05, momentum=0.01, affine=True, track_running_stats=True)
  (se_block): Identity()
)
)

```

```

(decoder): Bottleneck_Decoder(
  (bnecks): ListModule(
    (0): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): Hardswish()
        (expansion): Conv2d(576, 1152, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=1152)
        (bn2): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): SEBlock(
          (global_pool): AdaptiveAvgPool2d(output_size=1)
          (fc1): Linear(in_features=1152, out_features=288, bias=True)
          (fc2): Linear(in_features=288, out_features=1152, bias=True)
        )
        (project): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): Hardswish()
        (expansion): Conv2d(96, 192, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(192, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2), groups=192)
        (bn2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): SEBlock(
          (global_pool): AdaptiveAvgPool2d(output_size=1)
          (fc1): Linear(in_features=192, out_features=48, bias=True)
          (fc2): Linear(in_features=48, out_features=192, bias=True)
        )
        (project): Conv2d(192, 24, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (2): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(48, 96, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
        (bn2): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): Identity()
        (project): Conv2d(96, 16, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (3): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(32, 64, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=64)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): Identity()
        (project): Conv2d(64, 16, kernel_size=(1, 1), stride=(1, 1))
        (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (4): Compression_Bottleneck_Single(
      (bneck): Mobile_Inverted_Bottleneck(
        (activation): ReLU6()
        (expansion): Conv2d(32, 64, kernel_size=(1, 1), stride=(1, 1))
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (dwise): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=64)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (se_block): SEBlock(
          (global_pool): AdaptiveAvgPool2d(output_size=1)
          (fc1): Linear(in_features=64, out_features=16, bias=True)
        )
      )
    )
  )
)

```

```

        (fc2): Linear(in_features=16, out_features=64, bias=True)
    )
    (project): Conv2d(64, 16, kernel_size=(1, 1), stride=(1, 1))
    (bn3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
)
(act): ReLU6(inplace=True)
(label): Conv2d(16, 1, kernel_size=(1, 1), stride=(1, 1))
)
)
)
)

```

3 ShuSegL Network Architecture

Decoder Architecture

Stage	Input Shape	Out Channels	Repetitions
1	$112 \times 112 \times 48$	24	1
2	$56 \times 56 \times 48$	24	2
3	$28 \times 28 \times 232$	24	4
4	$14 \times 14 \times 464$	116	8
5	$7 \times 7 \times 1024$	232	5

Table 3: Bottleneck block parameters of the ShuSegL decoder.

PyTorch print

```

ShuSegL(
  (encoder): ShuffleNetv2_1_0(
    (conv1): Sequential(
      (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
      (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (max_pool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (stage2_block1): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=24)
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=58)
        (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (stage2_block2): ShuffleNetBlock(
    (left_branch): Sequential()
    (right_branch): Sequential()
  )
)

```

```

(0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
(1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU()
(3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58)
(4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
(6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(7): ReLU()
)
)
(stage2_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage2_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block1): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)

```

```

(stage3_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block5): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block6): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block7): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block8): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(

```

```

(0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
(1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU()
(3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
(4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
(6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(7): ReLU()
)
)
(stage4_block1): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232)
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
(stage4_block2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
(stage4_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
(stage4_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)

```

```

(conv5): Sequential(
  (0): Conv2d(464, 1024, kernel_size=(1, 1), stride=(1, 1))
  (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
)
(decoder): ShuffleBlock_Decoder(
  (shuffle_blocks): ListModule(
    (0): Compression_Shuffle_Block(
      (shuffle_blocks): ListModule(
        (0): ShuffleNetBlock(
          (left_branch): Sequential(
            (0): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=1024)
            (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): Conv2d(1024, 232, kernel_size=(1, 1), stride=(1, 1))
            (3): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (4): ReLU(inplace=True)
          )
          (right_branch): Sequential(
            (0): Conv2d(1024, 232, kernel_size=(1, 1), stride=(1, 1))
            (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
            (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
            (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
            (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (7): ReLU()
          )
        )
      )
    )
  )
  (1): ShuffleNetBlock(
    (left_branch): Sequential(
      (right_branch): Sequential(
        (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
        (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (2): ShuffleNetBlock(
    (left_branch): Sequential(
      (right_branch): Sequential(
        (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
        (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (3): ShuffleNetBlock(
    (left_branch): Sequential(
      (right_branch): Sequential(
        (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
        (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
)
)

```

```

(4): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(464, 464, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=464)
    (1): BatchNorm2d(464, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(464, 116, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(464, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
)
(1): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(464, 464, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=464)
        (1): BatchNorm2d(464, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(464, 116, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(464, 116, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (1): ShuffleNetBlock(
    (left_branch): Sequential(
      (right_branch): Sequential(
        (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (2): ShuffleNetBlock(
    (left_branch): Sequential(
      (right_branch): Sequential(
        (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
)
)

```



```

(3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(5): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(6): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(7): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(232, 58, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(232, 58, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)

```

```

    (7): ReLU()
  )
)
)
)
(2): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232)
        (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(232, 24, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(232, 24, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
        (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
)
(1): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
    (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
    (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(3): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12)
    (4): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(12, 12, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)

```

```

        (7): ReLU()
    )
)
)
)
(3): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (right_branch): Sequential(
          (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
          (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU()
          (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
          (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
          (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (7): ReLU()
        )
      )
    (1): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
        (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12)
        (4): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(12, 12, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
)
)
(4): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
        (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12)
        (4): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(12, 12, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
)
)
)
(label): Conv2d(24, 1, kernel_size=(1, 1), stride=(1, 1))
)
)

```

4 ShuSegS Network Architecture

Decoder Architecture

Stage	Input Shape	Out Channels	Repetitions
1	$112 \times 112 \times 48$	24	1
2	$56 \times 56 \times 48$	24	2
3	$28 \times 28 \times 96$	24	4
4	$14 \times 14 \times 192$	48	8
5	$7 \times 7 \times 1024$	96	5

Table 4: Bottleneck block parameters of the ShuSegS decoder.

PyTorch print

```
ShuSegS(  
  (encoder): ShuffleNetv2_0_5(  
    (conv1): Sequential(  
      (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      (2): ReLU(inplace=True)  
    )  
    (max_pool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)  
    (stage2_block1): ShuffleNetBlock(  
      (left_branch): Sequential(  
        (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=24)  
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (2): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))  
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (4): ReLU(inplace=True)  
      )  
      (right_branch): Sequential(  
        (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))  
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (2): ReLU()  
        (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=24)  
        (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))  
        (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (7): ReLU()  
      )  
    )  
    (stage2_block2): ShuffleNetBlock(  
      (left_branch): Sequential()  
      (right_branch): Sequential(  
        (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))  
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (2): ReLU()  
        (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)  
        (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))  
        (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (7): ReLU()  
      )  
    )  
  )  
)
```

```

(stage2_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
    (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage2_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
    (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block1): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=48)
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)

```

```

    (7): ReLU()
  )
)
(stage3_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block5): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block6): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block7): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage3_block8): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)

```

```

(stage4_block1): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=96)
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage4_block2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage4_block3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(stage4_block4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(conv5): Sequential(
  (0): Conv2d(192, 1024, kernel_size=(1, 1), stride=(1, 1))
  (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
(decoder): ShuffleBlock_Decoder(
  (shuffle_blocks): ListModule(
    (0): Compression_Shuffle_Block(
      (shuffle_blocks): ListModule(

```

```

(0): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=1024)
    (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(1024, 96, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(1024, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(1): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(2): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(3): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(4): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192)
    (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1))

```



```

(1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU()
(3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
(4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
(6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(7): ReLU()
)
)
)
(1): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192)
        (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1))
        (3): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (right_branch): Sequential(
        (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1))
        (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
        (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
        (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU()
      )
    )
  )
  (1): ShuffleNetBlock(
    (left_branch): Sequential()
    (right_branch): Sequential(
      (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
      (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU()
    )
  )
  (2): ShuffleNetBlock(
    (left_branch): Sequential()
    (right_branch): Sequential(
      (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
      (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU()
    )
  )
  (3): ShuffleNetBlock(
    (left_branch): Sequential()
    (right_branch): Sequential(
      (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
      (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
      (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

    (7): ReLU()
  )
)
(4): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(5): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(6): ShuffleNetBlock(
  (left_branch): Sequential()
  (right_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (4): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
(7): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)
    (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
    (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
)
(2): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=96)

```

```

(1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
(3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(4): ReLU(inplace=True)
)
(right_branch): Sequential(
  (0): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
  (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU()
  (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
  (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
  (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (7): ReLU()
)
)
(1): ShuffleNetBlock(
  (left_branch): Sequential(
    (right_branch): Sequential(
      (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
      (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
      (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU()
    )
  )
)
(2): ShuffleNetBlock(
  (left_branch): Sequential(
    (right_branch): Sequential(
      (0): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24)
      (4): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1))
      (6): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU()
    )
  )
)
(3): ShuffleNetBlock(
  (left_branch): Sequential(
    (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48)
    (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
    (3): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): ReLU(inplace=True)
  )
  (right_branch): Sequential(
    (0): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12)
    (4): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(12, 12, kernel_size=(1, 1), stride=(1, 1))
    (6): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU()
  )
)
)
)
(3): Compression_Shuffle_Block(
  (shuffle_blocks): ListModule(
    (0): ShuffleNetBlock(
      (left_branch): Sequential(
        (right_branch): Sequential(

```


5 GhoSegL Network Architecture

Decoder Architecture

Stage	Input Shape	Out Channels	SE-Block
1	$112 \times 112 \times 32$	16	-
2	$56 \times 56 \times 48$	16	✓
3	$28 \times 28 \times 80$	24	✓
4	$14 \times 14 \times 224$	40	✓
5	$7 \times 7 \times 960$	112	✓

Table 5: Bottleneck block parameters of the GhoSegL decoder.

PyTorch print

```
GhoSegL(  
  (encoder): GhostNet(  
    (initial_conv): Sequential(  
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)  
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      (2): ReLU(inplace=True)  
    )  
    (stage1_gbneck1): GhostBottleneck(  
      (ghost_module1): GhostModule(  
        (primary_conv): Sequential(  
          (0): Conv2d(16, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)  
          (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
          (2): ReLU(inplace=True)  
        )  
        (cheap_operation): Sequential(  
          (0): Conv2d(8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=8, bias=False)  
          (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
          (2): ReLU(inplace=True)  
        )  
      )  
      (downsample): Identity()  
      (se_block): Identity()  
      (ghost_module2): GhostModule(  
        (primary_conv): Sequential(  
          (0): Conv2d(16, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)  
          (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
          (2): Identity()  
        )  
        (cheap_operation): Sequential(  
          (0): Conv2d(8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=8, bias=False)  
          (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
          (2): Identity()  
        )  
      )  
      (shortcut): Identity()  
    )  
    (stage2_gbneck1): GhostBottleneck(  
      (ghost_module1): GhostModule(  
        (primary_conv): Sequential(  
          (0): Conv2d(16, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)  
          (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
          (2): ReLU(inplace=True)  
        )  
        (cheap_operation): Sequential(  
          (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=24, bias=False)
```

```

        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
    )
)
(downsample): Sequential(
  (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=48, bias=False)
  (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(se_block): Identity()
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(48, 12, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12, bias=False)
    (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Sequential(
  (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=16, bias=False)
  (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(16, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(stage2_gbneck2): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(24, 36, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(36, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(36, 36, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=36, bias=False)
      (1): BatchNorm2d(36, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): Identity()
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(72, 12, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12, bias=False)
      (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
  (shortcut): Identity()
)
)
(stage3_gbneck1): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(24, 36, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(36, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(36, 36, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=36, bias=False)
      (1): BatchNorm2d(36, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )

```

```

)
)
(downsample): Sequential(
  (0): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=72, bias=False)
  (1): BatchNorm2d(72, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=72, out_features=18, bias=True)
  (fc2): Linear(in_features=18, out_features=72, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(72, 20, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(20, 20, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=20, bias=False)
    (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Sequential(
  (0): Conv2d(24, 24, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=24, bias=False)
  (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(24, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(stage3_gbneck2): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(40, 60, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(60, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(60, 60, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=60, bias=False)
      (1): BatchNorm2d(60, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)
(downsample): Identity()
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=120, out_features=30, bias=True)
  (fc2): Linear(in_features=30, out_features=120, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(120, 20, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(20, 20, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=20, bias=False)
    (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Identity()
)
(stage4_gbneck1): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

    (2): ReLU(inplace=True)
  )
  (cheap_operation): Sequential(
    (0): Conv2d(120, 120, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=120, bias=False)
    (1): BatchNorm2d(120, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
  )
)
(downsample): Sequential(
  (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=240, bias=False)
  (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(se_block): Identity()
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=40, bias=False)
    (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Sequential(
  (0): Conv2d(40, 40, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=40, bias=False)
  (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(40, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(stage4_gbneck2): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(80, 100, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(100, 100, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=100, bias=False)
      (1): BatchNorm2d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): Identity()
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(200, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=40, bias=False)
      (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
)
(shortcut): Identity()
)
(stage4_gbneck3): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(80, 92, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(92, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)

```



```

(cheap_operation): Sequential(
  (0): Conv2d(92, 92, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=92, bias=False)
  (1): BatchNorm2d(92, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
)
(downsample): Identity()
(se_block): Identity()
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(184, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=40, bias=False)
    (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Identity()
)
(stage4_gbneck4): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(80, 92, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(92, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(92, 92, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=92, bias=False)
      (1): BatchNorm2d(92, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): Identity()
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(184, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=40, bias=False)
      (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
  (shortcut): Identity()
)
)
(stage4_gbneck5): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(80, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=240, bias=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): SEBlock(
    (global_pool): AdaptiveAvgPool2d(output_size=1)
    (fc1): Linear(in_features=480, out_features=120, bias=True)
  )
)

```

```

    (fc2): Linear(in_features=120, out_features=480, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(480, 56, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(56, 56, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=56, bias=False)
    (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Sequential(
  (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
  (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(80, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(stage4_gbneck6): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(112, 336, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(336, 336, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)
      (1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)
(downsample): Identity()
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=672, out_features=168, bias=True)
  (fc2): Linear(in_features=168, out_features=672, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(672, 56, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(56, 56, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=56, bias=False)
    (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Identity()
)
(stage5_gbneck1): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(112, 336, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(336, 336, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=336, bias=False)
      (1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)
(downsample): Sequential(

```

```

(0): Conv2d(672, 672, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=672, bias=False)
(1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=672, out_features=168, bias=True)
  (fc2): Linear(in_features=168, out_features=672, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(672, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
)
(shortcut): Sequential(
  (0): Conv2d(112, 112, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=112, bias=False)
  (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(112, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(stage5_gbneck2): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(160, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)
(downsample): Identity()
(se_block): Identity()
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(960, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
)
(shortcut): Identity()
)
(stage5_gbneck3): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(160, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
)
)

```

```

)
(downsample): Identity()
(se_block): SEBlock(
  (global_pool): AdaptiveAvgPool2d(output_size=1)
  (fc1): Linear(in_features=960, out_features=240, bias=True)
  (fc2): Linear(in_features=240, out_features=960, bias=True)
)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(960, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Identity()
)
(stage5_gbneck4): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(160, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): Identity()
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(960, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
      (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
  (shortcut): Identity()
)
(stage5_gbneck5): GhostBottleneck(
  (ghost_module1): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(160, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): SEBlock(
    (global_pool): AdaptiveAvgPool2d(output_size=1)
    (fc1): Linear(in_features=960, out_features=240, bias=True)
    (fc2): Linear(in_features=240, out_features=960, bias=True)
  )
)

```

```

)
(ghost_module2): GhostModule(
  (primary_conv): Sequential(
    (0): Conv2d(960, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
  (cheap_operation): Sequential(
    (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
    (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Identity()
  )
)
(shortcut): Identity()
)
(final_conv): Sequential(
  (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
)
(decoder): GhostBottleneck_Decoder(
  (gbnecks): ListModule(
    (0): Compression_Bottleneck_Ghost(
      (gbneck): GhostBottleneck(
        (ghost_module1): GhostModule(
          (primary_conv): Sequential(
            (0): Conv2d(960, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (cheap_operation): Sequential(
            (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=960, bias=False)
            (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU(inplace=True)
          )
        )
      )
      (downsample): Identity()
      (se_block): SEBlock(
        (global_pool): AdaptiveAvgPool2d(output_size=1)
        (fc1): Linear(in_features=1920, out_features=480, bias=True)
        (fc2): Linear(in_features=480, out_features=1920, bias=True)
      )
      (ghost_module2): GhostModule(
        (primary_conv): Sequential(
          (0): Conv2d(1920, 56, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): Identity()
        )
        (cheap_operation): Sequential(
          (0): Conv2d(56, 56, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=56, bias=False)
          (1): BatchNorm2d(56, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): Identity()
        )
      )
      (shortcut): Sequential(
        (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=960, bias=False)
        (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Conv2d(960, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
)
(1): Compression_Bottleneck_Ghost(
  (gbneck): GhostBottleneck(
    (ghost_module1): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(224, 224, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (1): BatchNorm2d(224, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
    )
    (cheap_operation): Sequential(
      (0): Conv2d(224, 224, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=224, bias=False)
      (1): BatchNorm2d(224, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
    )
  )
  (downsample): Identity()
  (se_block): SEBlock(
    (global_pool): AdaptiveAvgPool2d(output_size=1)
    (fc1): Linear(in_features=448, out_features=112, bias=True)
    (fc2): Linear(in_features=112, out_features=448, bias=True)
  )
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(448, 20, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(20, 20, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=20, bias=False)
      (1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
  (shortcut): Sequential(
    (0): Conv2d(224, 224, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=224, bias=False)
    (1): BatchNorm2d(224, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Conv2d(224, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(2): Compression_Bottleneck_Ghost(
  (gbneck): GhostBottleneck(
    (ghost_module1): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(80, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
      (cheap_operation): Sequential(
        (0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
        (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
    )
  )
  (downsample): Identity()
  (se_block): SEBlock(
    (global_pool): AdaptiveAvgPool2d(output_size=1)
    (fc1): Linear(in_features=160, out_features=40, bias=True)
    (fc2): Linear(in_features=40, out_features=160, bias=True)
  )
  (ghost_module2): GhostModule(
    (primary_conv): Sequential(
      (0): Conv2d(160, 12, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
    (cheap_operation): Sequential(
      (0): Conv2d(12, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=12, bias=False)
      (1): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Identity()
    )
  )
  (shortcut): Sequential(

```

```

(0): Conv2d(80, 80, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=80, bias=False)
(1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): Conv2d(80, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
(3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
)
(3): Compression_Bottleneck_Ghost(
  (gbneck): GhostBottleneck(
    (ghost_module1): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(48, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
      (cheap_operation): Sequential(
        (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48, bias=False)
        (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
    )
    (downsample): Identity()
    (se_block): SEBlock(
      (global_pool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Linear(in_features=96, out_features=24, bias=True)
      (fc2): Linear(in_features=24, out_features=96, bias=True)
    )
    (ghost_module2): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(96, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Identity()
      )
      (cheap_operation): Sequential(
        (0): Conv2d(8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=8, bias=False)
        (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Identity()
      )
    )
    (shortcut): Sequential(
      (0): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=48, bias=False)
      (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(48, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
)
(4): Compression_Bottleneck_Ghost(
  (gbneck): GhostBottleneck(
    (ghost_module1): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
      (cheap_operation): Sequential(
        (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32, bias=False)
        (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU(inplace=True)
      )
    )
    (downsample): Identity()
    (se_block): Identity()
    (ghost_module2): GhostModule(
      (primary_conv): Sequential(
        (0): Conv2d(64, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Identity()
      )

```

```
)
(cheap_operation): Sequential(
  (0): Conv2d(8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=8, bias=False)
  (1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Identity()
)
)
(shortcut): Sequential(
  (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32, bias=False)
  (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
)
)
(act): ReLU6(inplace=True)
(label): Conv2d(16, 1, kernel_size=(1, 1), stride=(1, 1))
)
```