# AIL 862

Lecture 6

# Challenges for Semantic Segmentation

- What and where
- Classes with similar spectral signature
- Inconspicuous classes

# Solution for the mentioned problems

- Context

# Solution for the mentioned problems

- Context
- Context at multiple scales

# Solution for the mentioned problems

- Context
- Context at multiple scales
- Global context

# Contraction Phase

Reduces spatial size

Increases understanding of content (“what”)

Though loses some object information due to contraction (loses “where” to some extent)

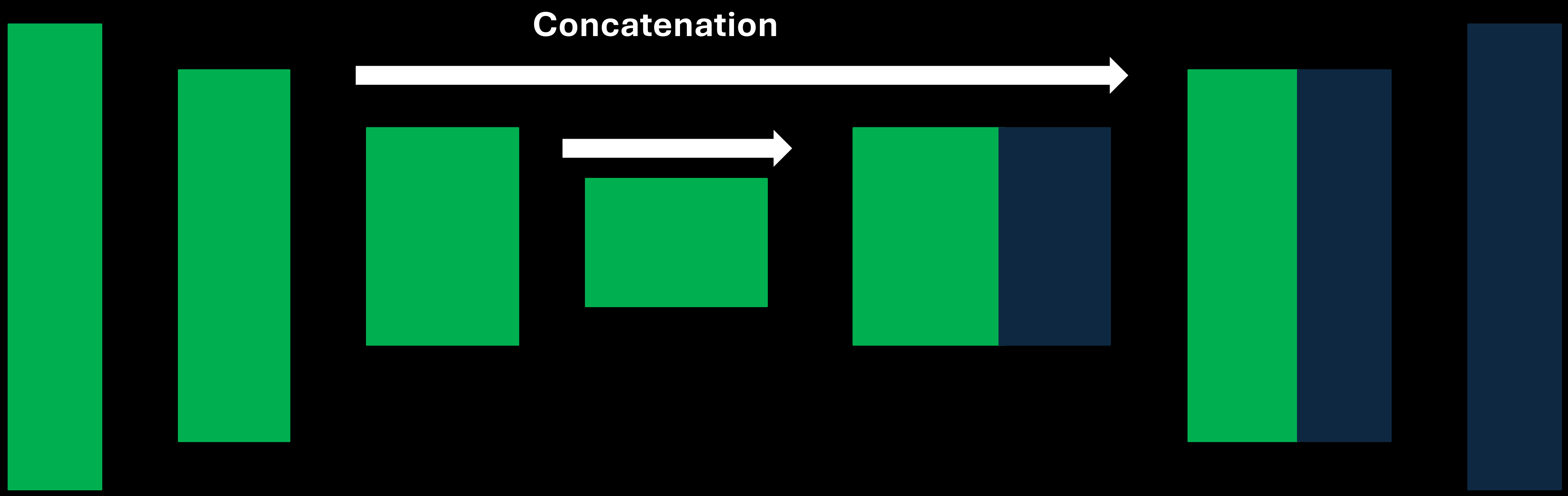
# Recover “where” in expansion phase

Concatenate feature maps from contraction phase during expansion phase during recovery

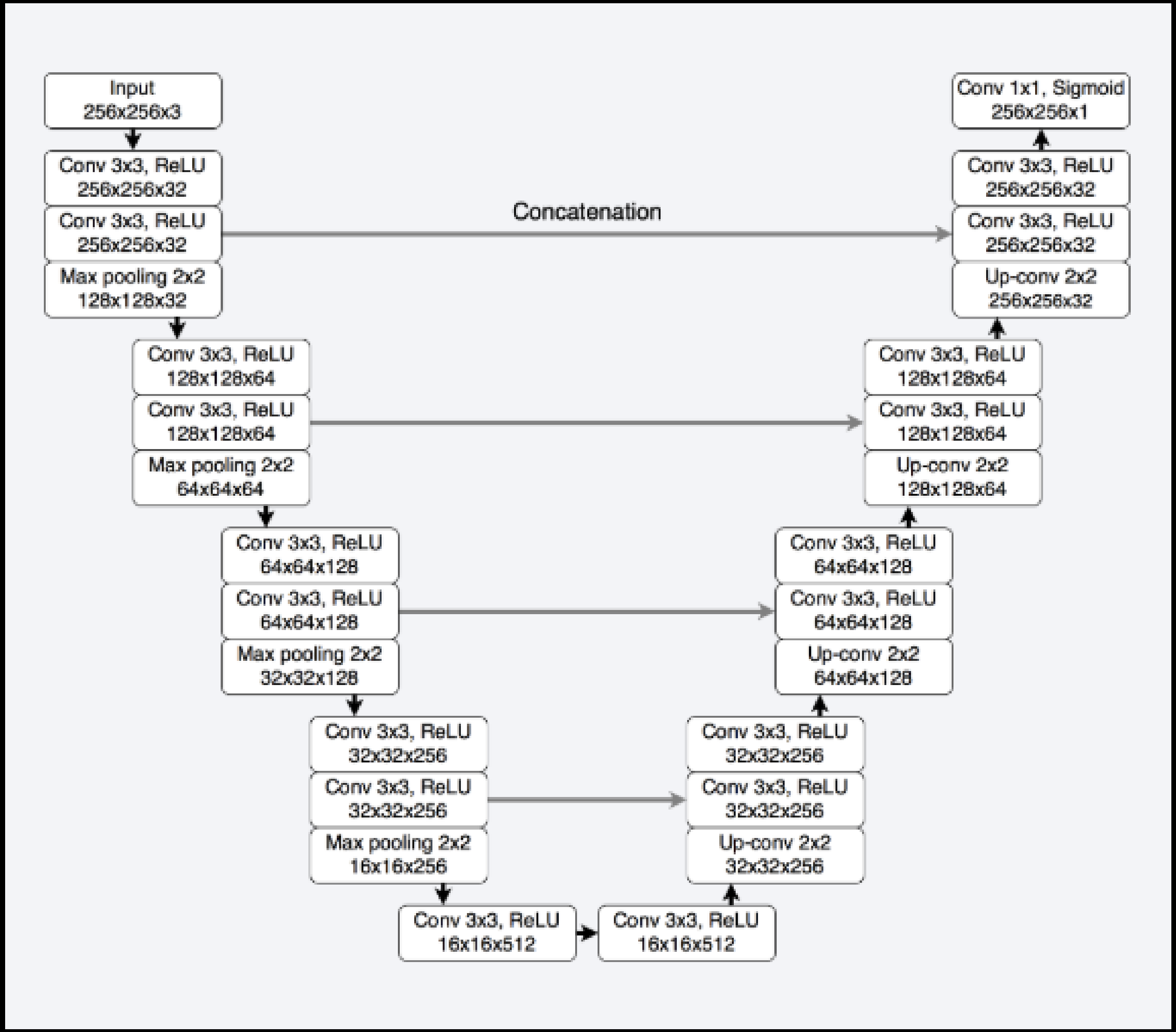
Helps in recovery of “where” information

Principle behind U-Net

# U-Net




Ronneberger et al., 2015




https://pytorch.org/hub/mateuszbuda_brain-segmentation-pytorch_unet/

```python
def forward(self, x):
    enc1 = self.encoder1(x)
    enc2 = self.encoder2(self.pool1(enc1))
    enc3 = self.encoder3(self.pool2(enc2))
    enc4 = self.encoder4(self.pool3(enc3))

    bottleneck = self.bottleneck(self.pool4(enc4))

    dec4 = self.upconv4(bottleneck)
    dec4 = torch.cat((dec4, enc4), dim=1)
    dec4 = self.decoder4(dec4)
    dec3 = self.upconv3(dec4)
    dec3 = torch.cat((dec3, enc3), dim=1)
    dec3 = self.decoder3(dec3)
    dec2 = self.upconv2(dec3)
    dec2 = torch.cat((dec2, enc2), dim=1)
    dec2 = self.decoder2(dec2)
    dec1 = self.upconv1(dec2)
    dec1 = torch.cat((dec1, enc1), dim=1)
    dec1 = self.decoder1(dec1)
    return torch.sigmoid(self.conv(dec1))
```

https://github.com/mateuszbuda/brain-segmentation-pytorch/blob/master/unet.py

```
self.decoder4 = UNet._block((features * 8) * 2, features * 8, name="dec4")
self.upconv3 = nn.ConvTranspose2d(
    features * 8, features * 4, kernel_size=2, stride=2
)
```

https://github.com/mateuszbuda/brain-segmentation-pytorch/blob/master/unet.py

# Spatial Attention + UNet

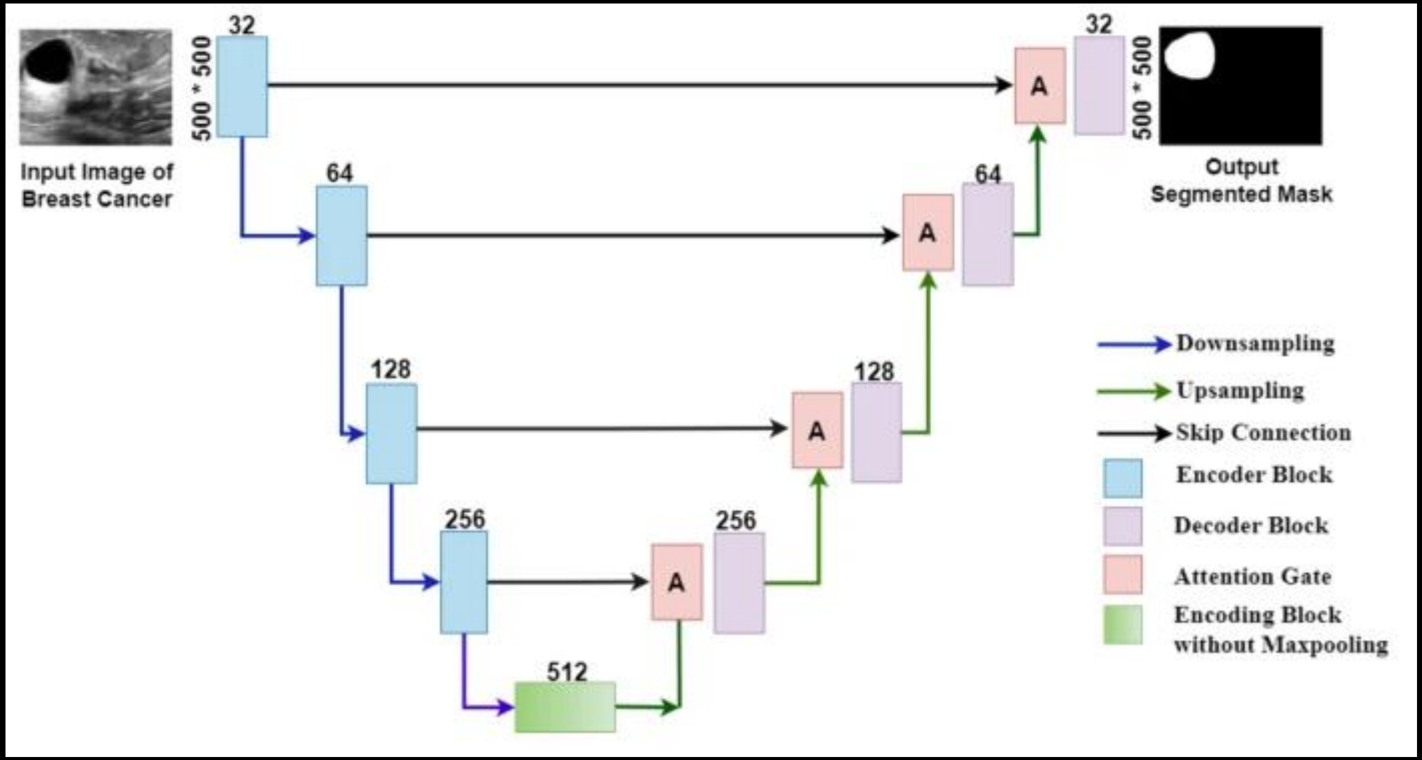



**Figure from: Attention based UNet model for breast cancer segmentation using BUSI dataset, 2024**

**However, idea of attention based UNet sis much older**

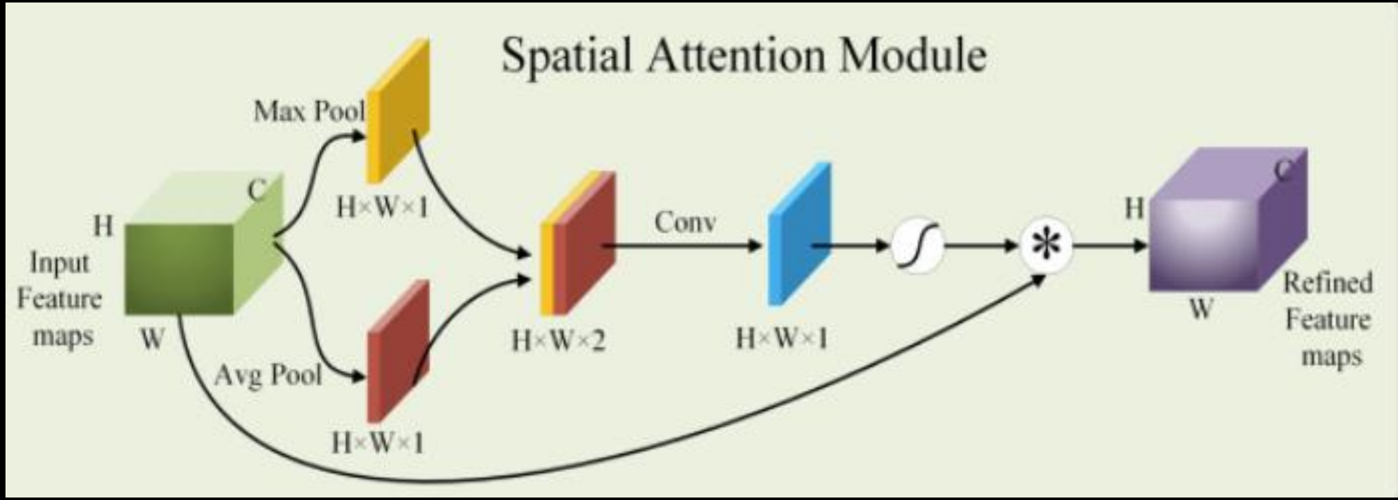
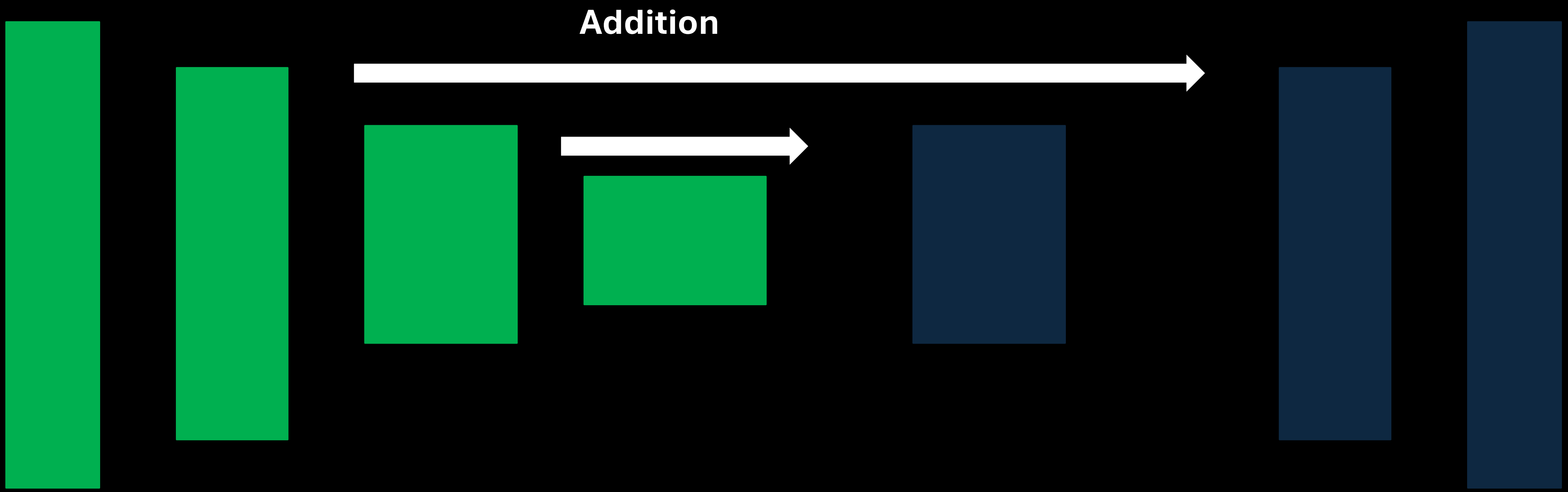
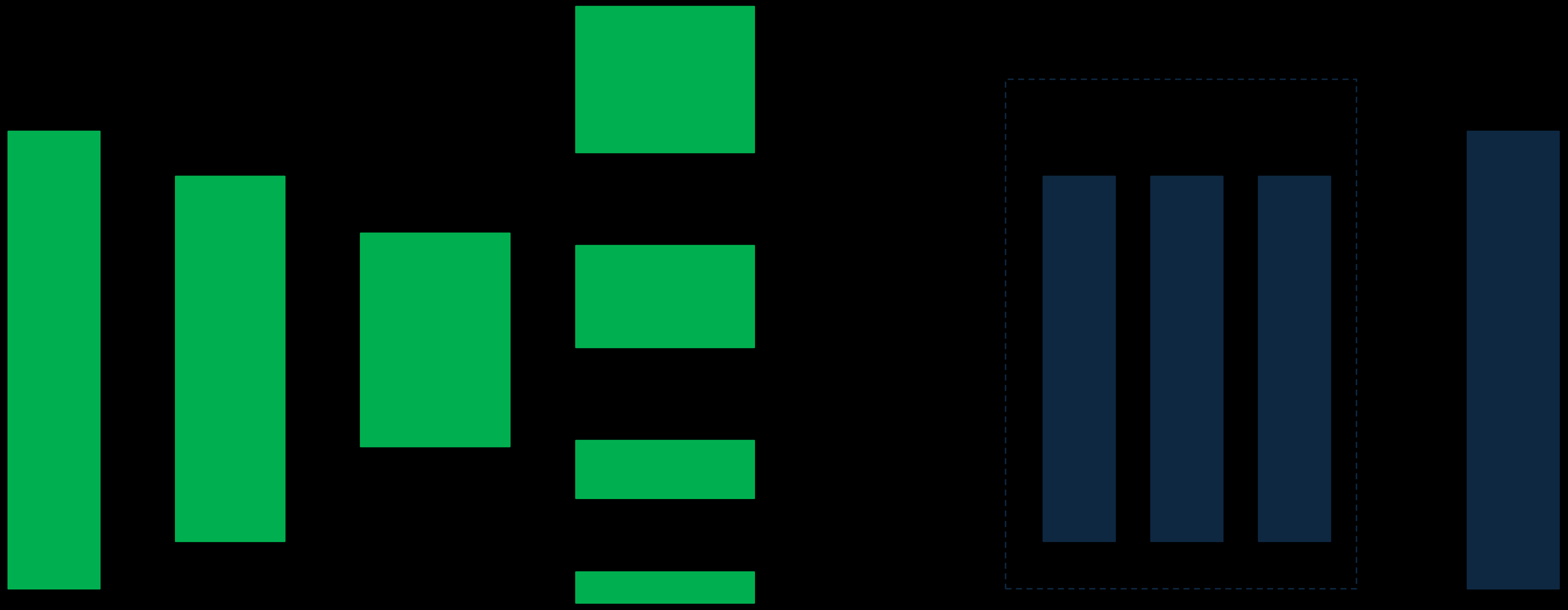



Figure from: An improved medical image segmentation framework with Channel-Height-Width-Spatial attention module

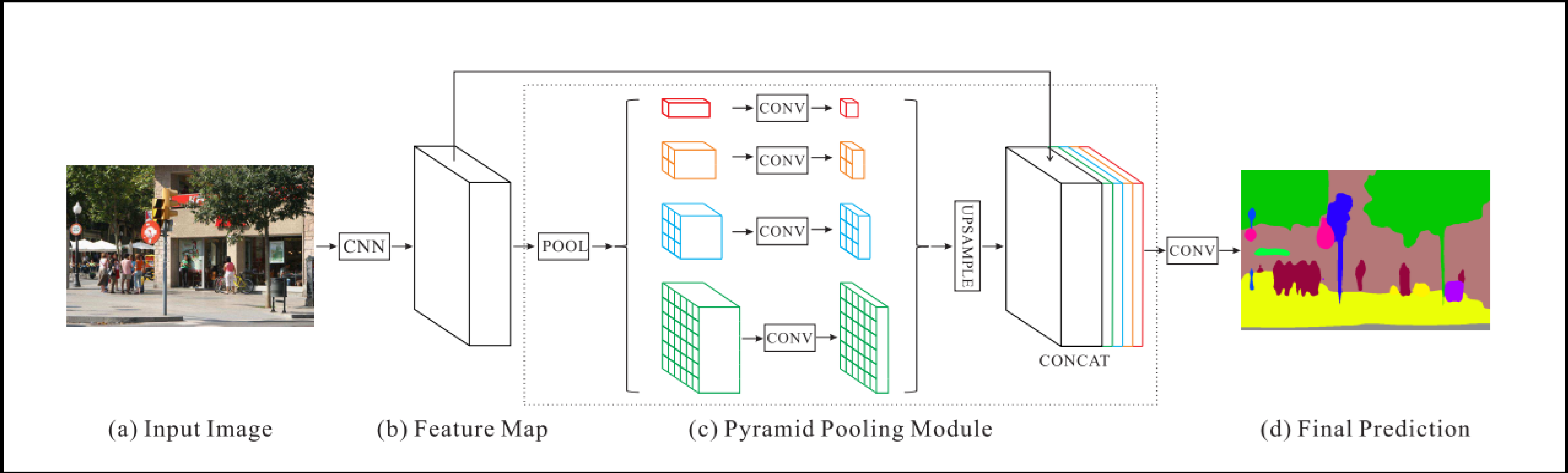
# LinkNet



# PSPNet



# PSPNet


CNN
POOL
CONV
CONV
CONV
CONV
UPSAMPLE
CONCAT
CONV
(a) Input Image
(b) Feature Map
(c) Pyramid Pooling Module
(d) Final Prediction

```python
def forward(self, x):
    f, class_f = self.feats(x)
    p = self.psp(f)
    p = self.drop_1(p)

    p = self.up_1(p)
    p = self.drop_2(p)

    p = self.up_2(p)
    p = self.drop_2(p)

    p = self.up_3(p)
    p = self.drop_2(p)

    auxiliary = F.adaptive_max_pool2d(input=class_f, output_size=(1, 1)).view(-1, class_f.size(1))

    return self.final(p), self.classifier(auxiliary)
```

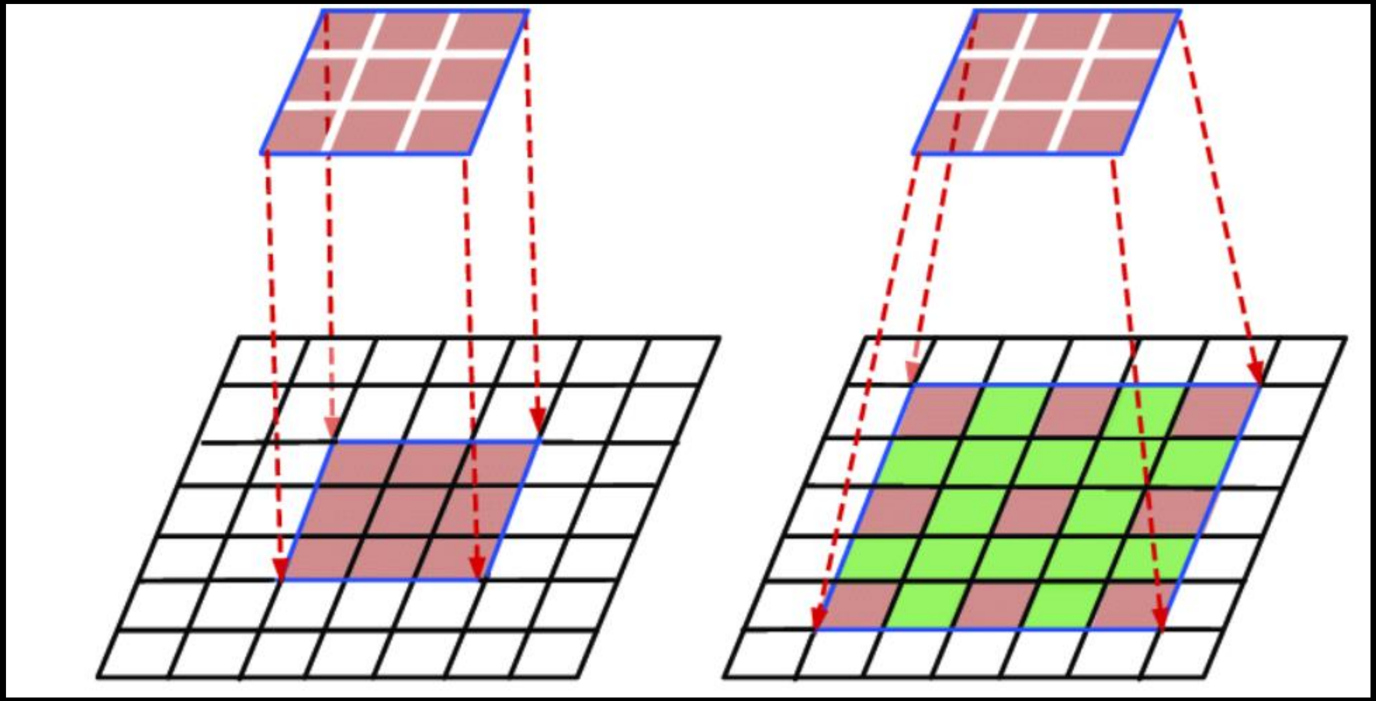
https://github.com/Lextal/pspnet-pytorch/blob/master/pspnet.py

```python
class PSPModule(nn.Module):
    def __init__(self, features, out_features=1024, sizes=(1, 2, 3, 6)):
        super().__init__()
        self.stages = []
        self.stages = nn.ModuleList([self._make_stage(features, size) for size in sizes])
        self.bottleneck = nn.Conv2d(features * (len(sizes) + 1), out_features, kernel_size=1)
        self.relu = nn.ReLU()

    def _make_stage(self, features, size):
        prior = nn.AdaptiveAvgPool2d(output_size=(size, size))
        conv = nn.Conv2d(features, features, kernel_size=1, bias=False)
        return nn.Sequential(prior, conv)

    def forward(self, feats):
        h, w = feats.size(2), feats.size(3)
        priors = [F.upsample(input=stage(feats), size=(h, w), mode='bilinear') for stage in self.stages] + [feats]
        bottle = self.bottleneck(torch.cat(priors, 1))
        return self.relu(bottle)
```

https://github.com/Lextal/pspnet-pytorch/blob/master/pspnet.py

# DeepLabV3

Rethinking Atrous Convolution for Semantic Image Segmentation, Chen et al., 2017



# Atrous Convolution

- ✓ Traditional convolution: fixed size filters do not capture features at different scales effectively.
- ✓ Strides and pooling layers are used to control the receptive field, though they cause information loss.

**Atrous convolution**

- ✓ Increased receptive field.
- ✓ Allows capturing features at multiple scales.
- ✓ Allows multi-scale feature extraction.

# Edge Pixels

UNet Original loss function: Gives weightage to edge pixels (“small separation borders that we introduce between touching cells”)

# SAR Segmentation

- ✓ Architecture wise same architectures are applicable for SAR segmentation as well.
- ✓ However, semantic segmentation performance is often sub-optimal in SAR images.
- ✓ Several works try to improve performance by providing additional training-time data source or using by identifying more reliable labels.

# SAR Segmentation (with optical side-information)

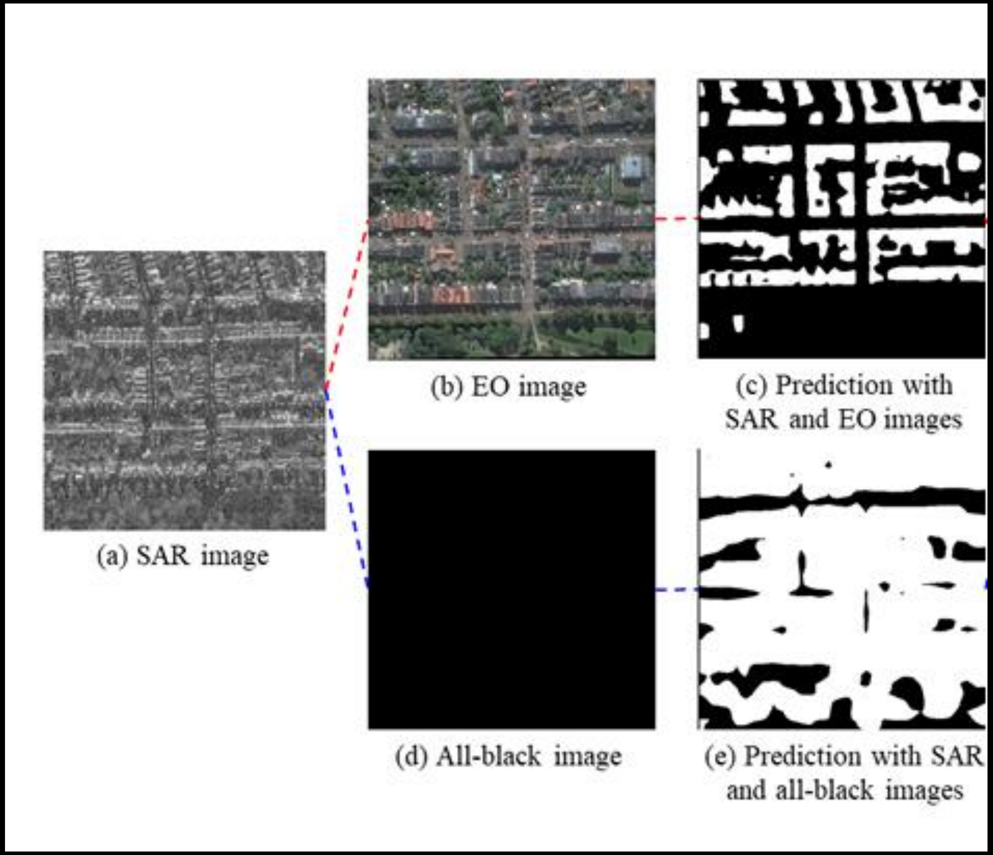
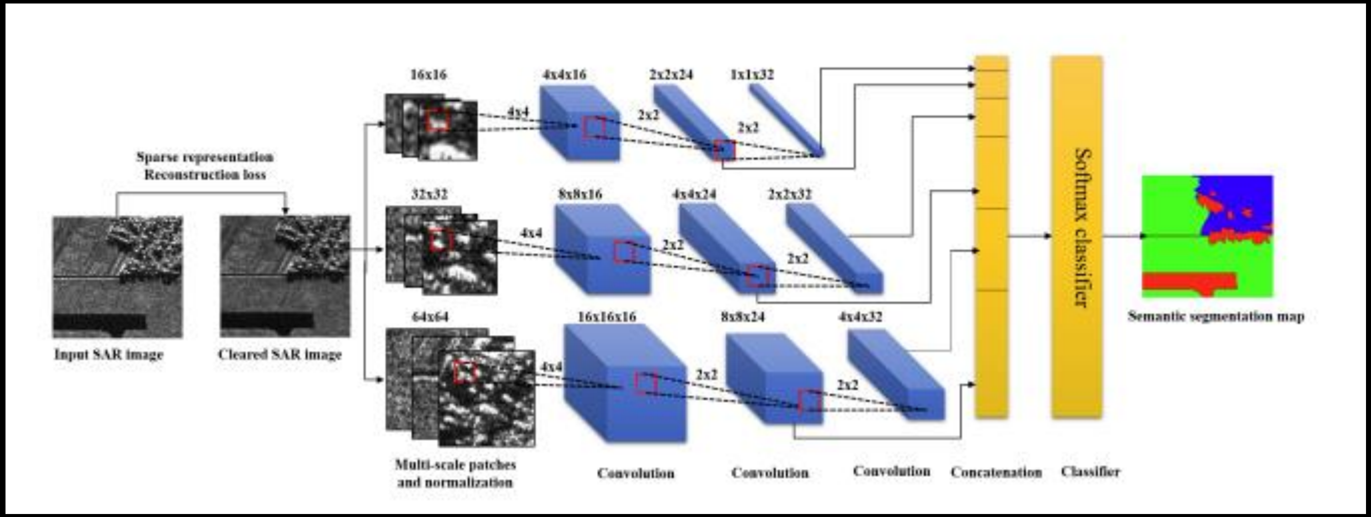


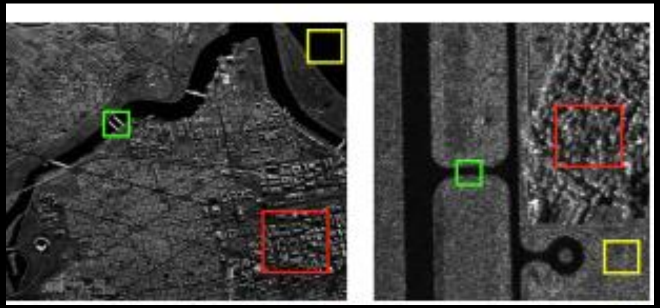

Figure from - Heterogeneous Feature Distillation Network for SAR Image Semantic Segmentation

EO means electro-optical in this slide.

Otherwise in this course, we abbreviate Earth observation as EO.

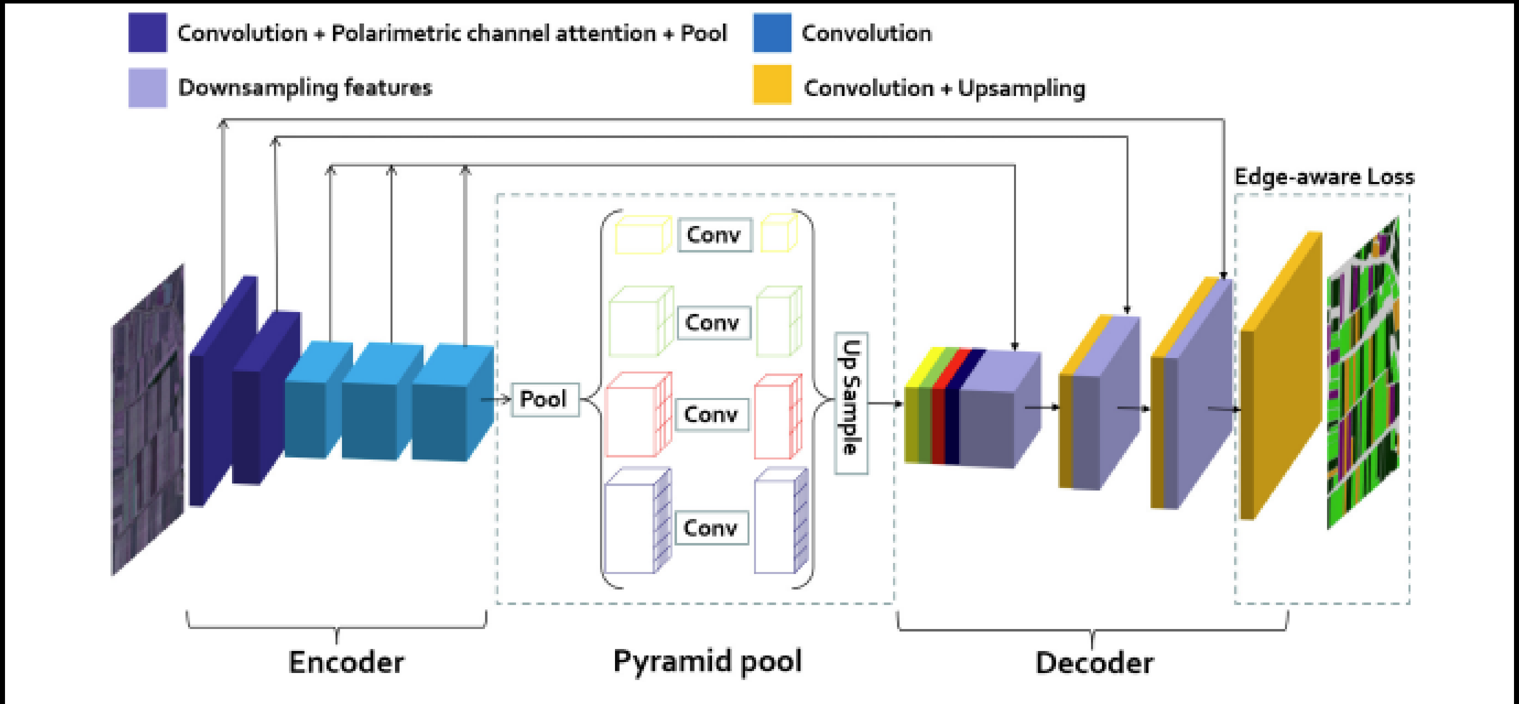
# SAR Segmentation



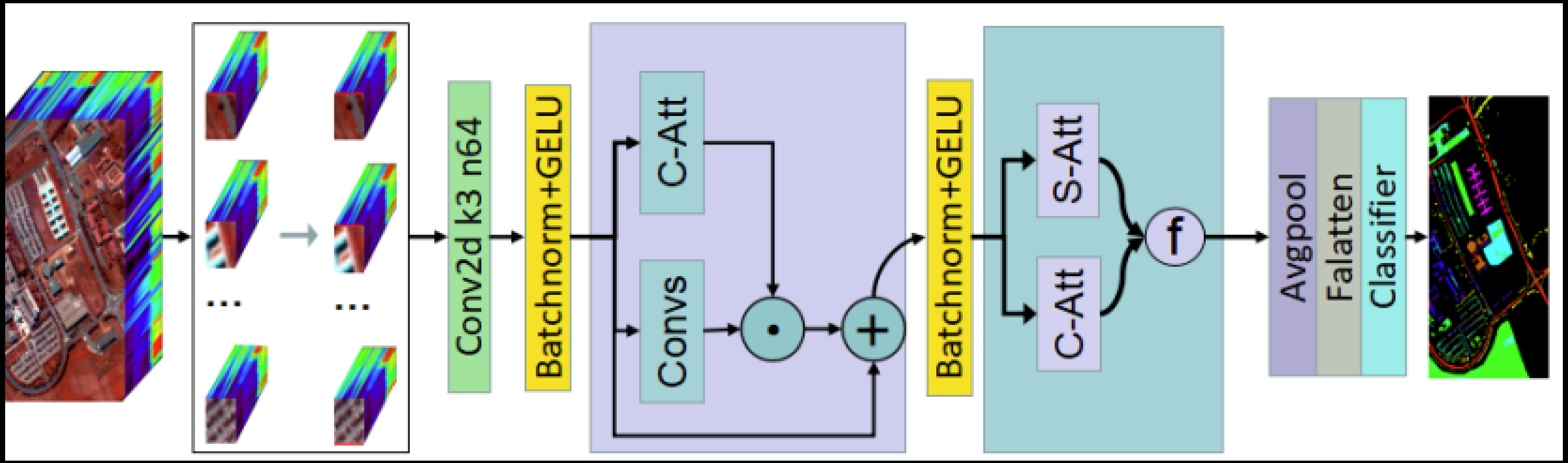
Multi-scale Convolutional Neural Network for SAR Image Semantic Segmentation

# SAR Segmentation




A Refined Pyramid Scene Parsing Network for Polarimetric SAR Image Semantic Segmentation in Agricultural Areas

# Channel Attention for Hyperspectral Images




ReSC-net: Hyperspectral Image Classification Based on Attention-Enhanced Residual Module and Spatial-Channel Attention

Paper reading allotment

Early Preparation Pays Off: New Classifier Pre-tuning for Class Incremental Semantic Segmentation

Ritik

CLIP-Guided Generative Networks for Transferable Targeted Adversarial Attacks

Shubhojit

ProS: Prompting-to-simulate Generalized knowledge for Universal Cross-Domain Retrieval

Kashish

FocusMAE: Gallbladder Cancer Detection from Ultrasound Videos with Focused Masked Autoencoders

Akshay