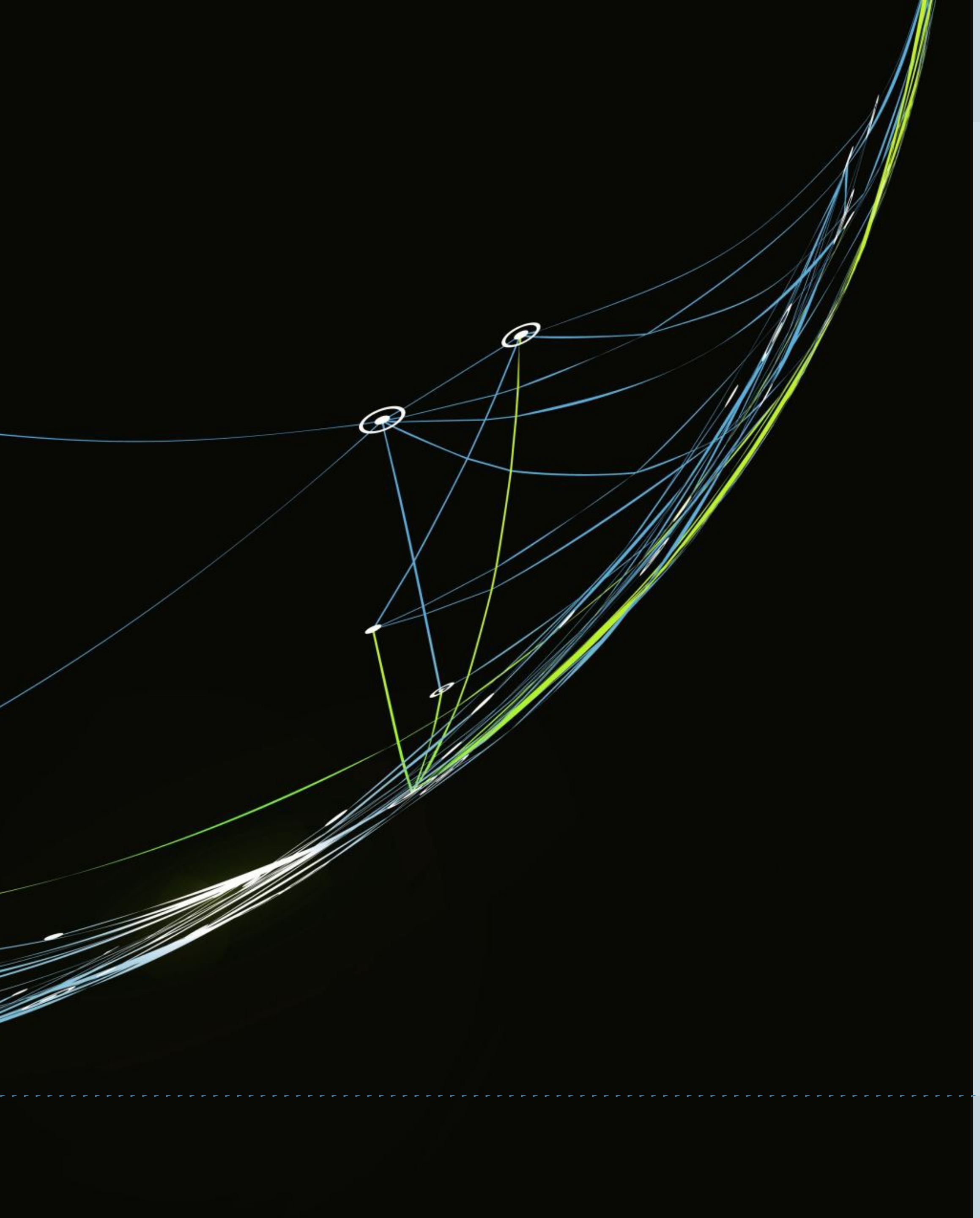


Building Change Detection using FCN Model on Aerial Satellite Images: A Comparative Analysis of 2012 and 2016 Datasets

Jingheng Huan Luyao Wang Fanbin Xu

Outline

- Motivation
- Dataset
- Conventional Method
- Our Approach
- Our model
- Future Work
- Other Applications
- Conclusion

1. Motivation

Motivation

- Efficient urban planning
- Data-driven public policy making
- Visualize urban transformation

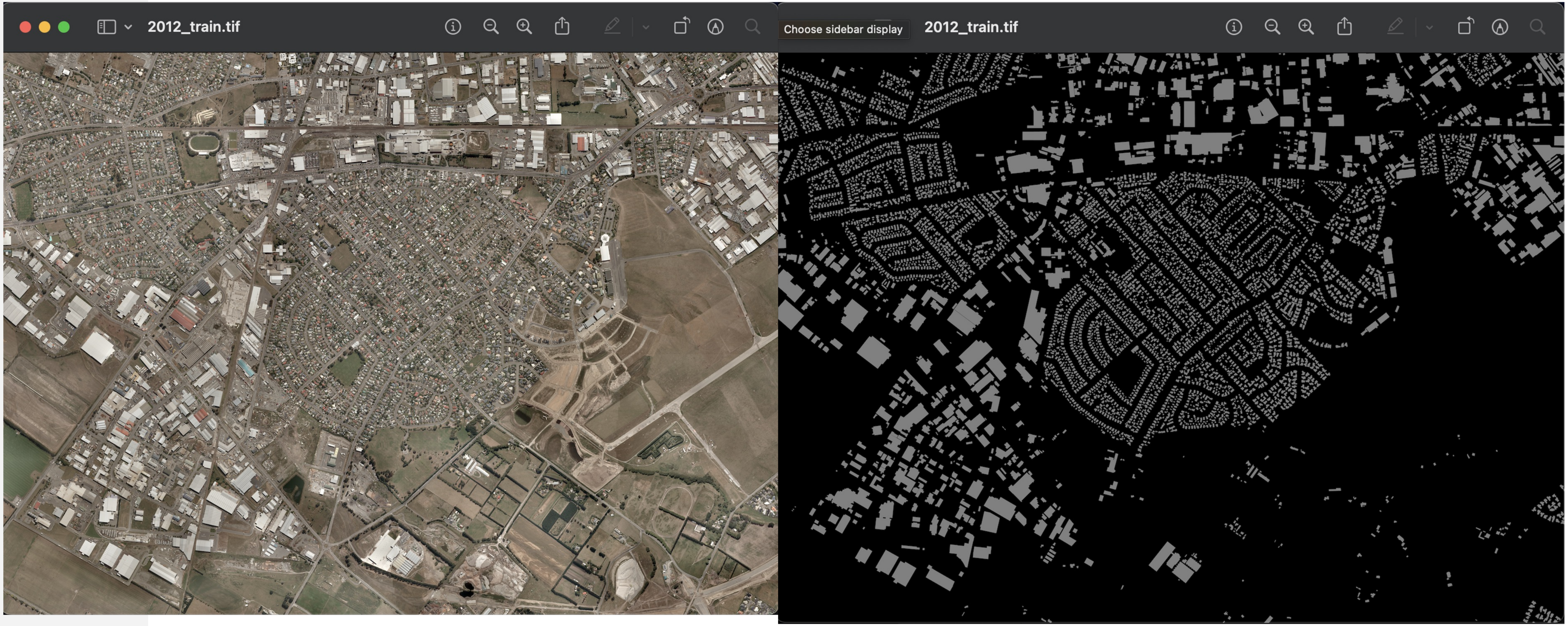
2. Dataset

April in 2012 and 2016

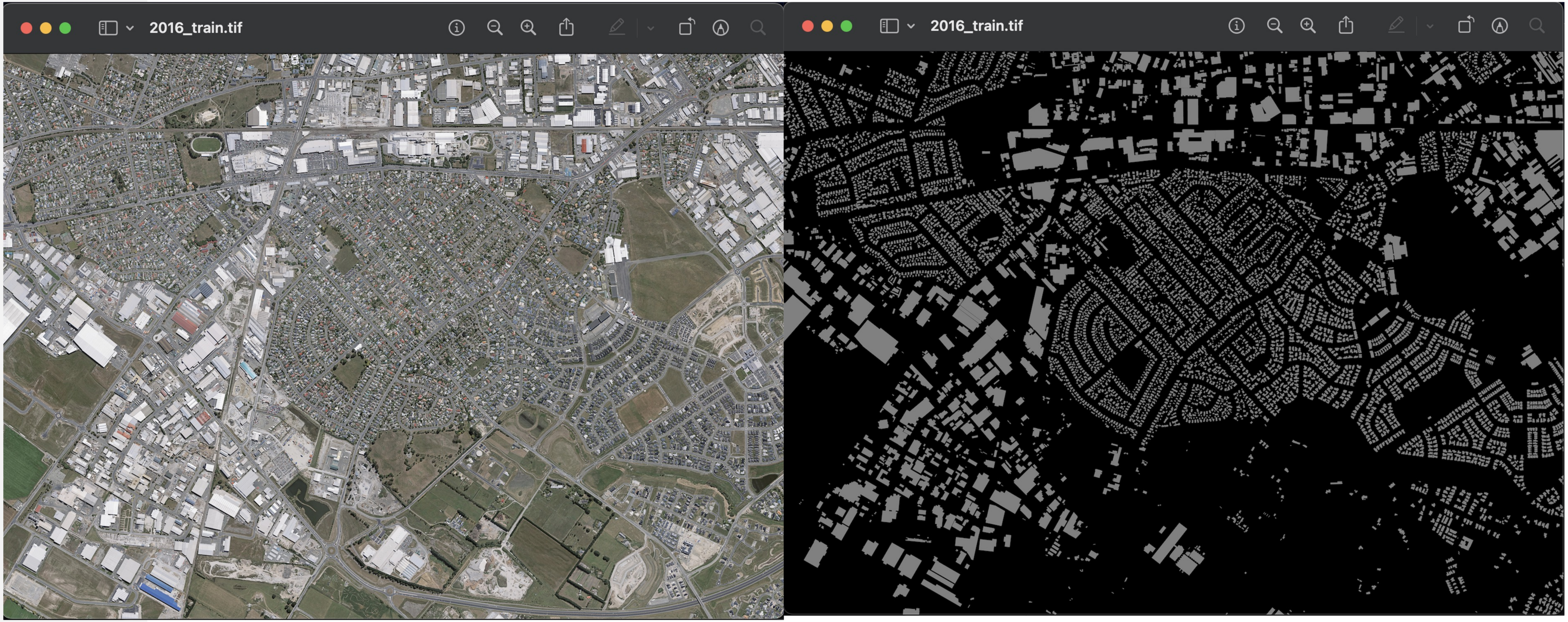
20.5 km^2

12,796 (2012) and 16,077 buildings (2016)

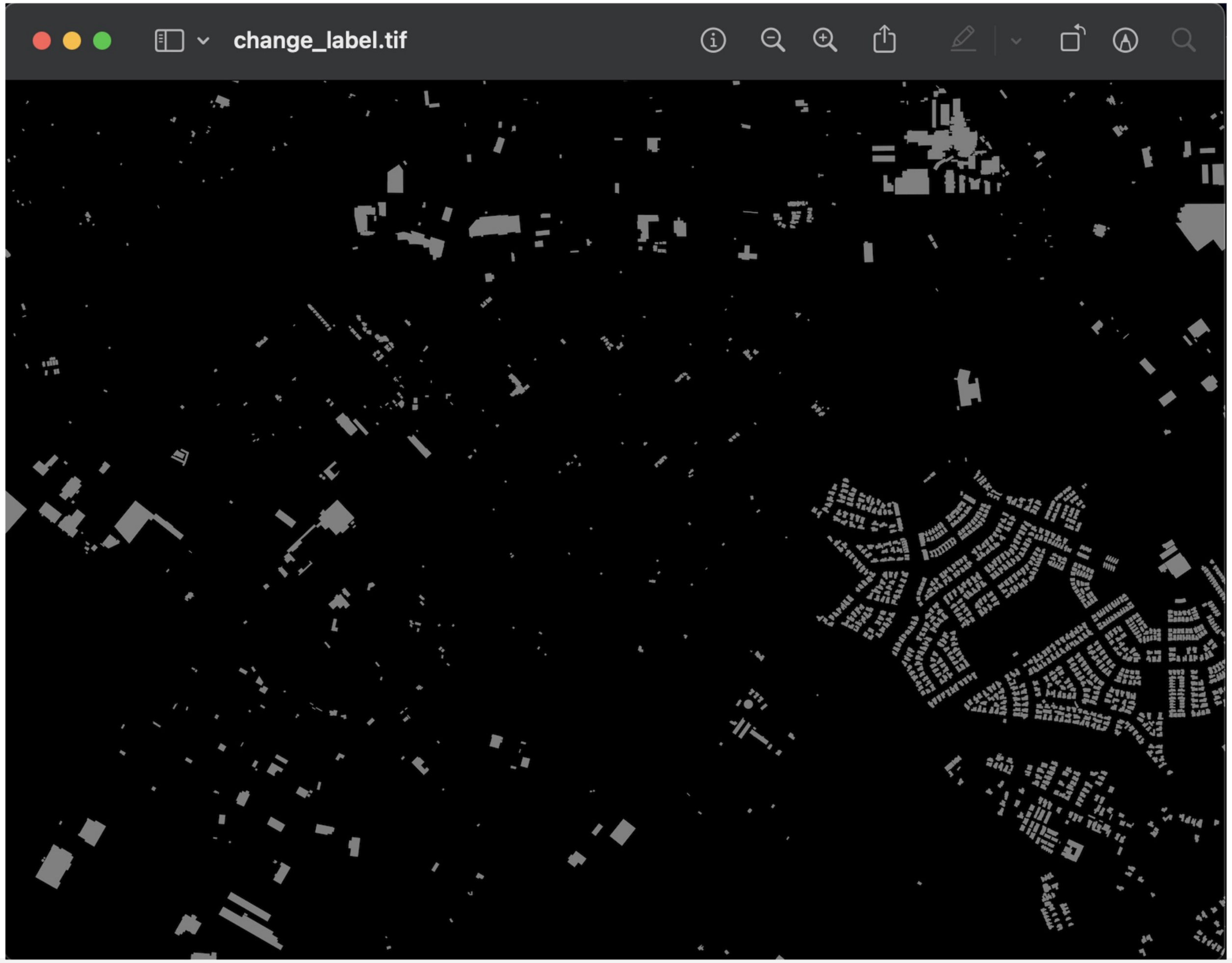
2.1 Preview of Dataset (2012)



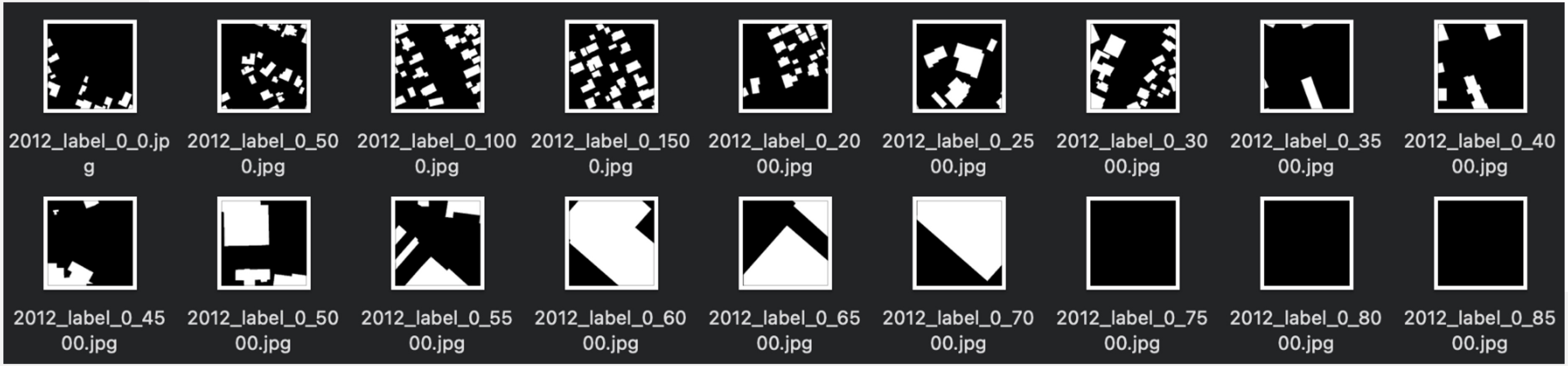
2.1 Preview of Dataset (2016)



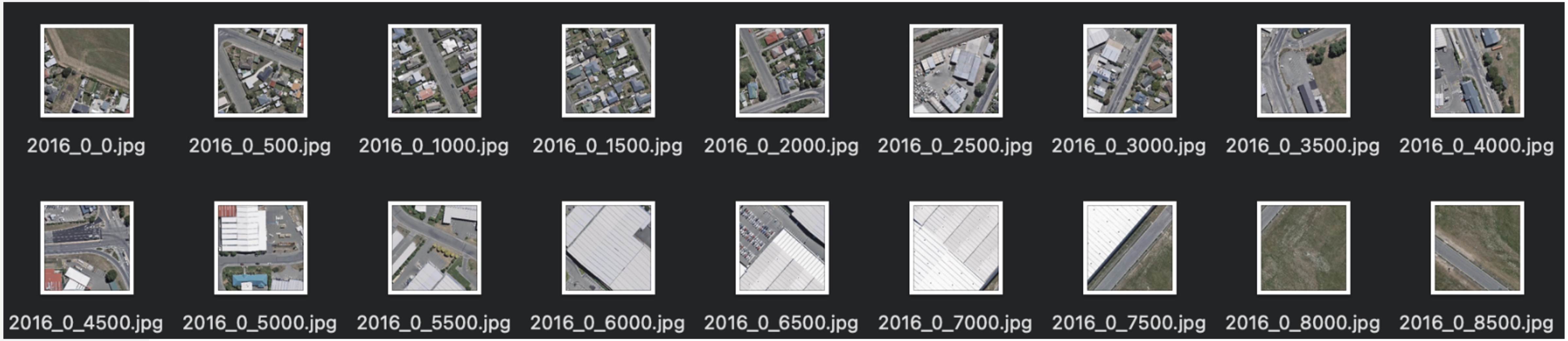
2.2 Preview of Change Detection

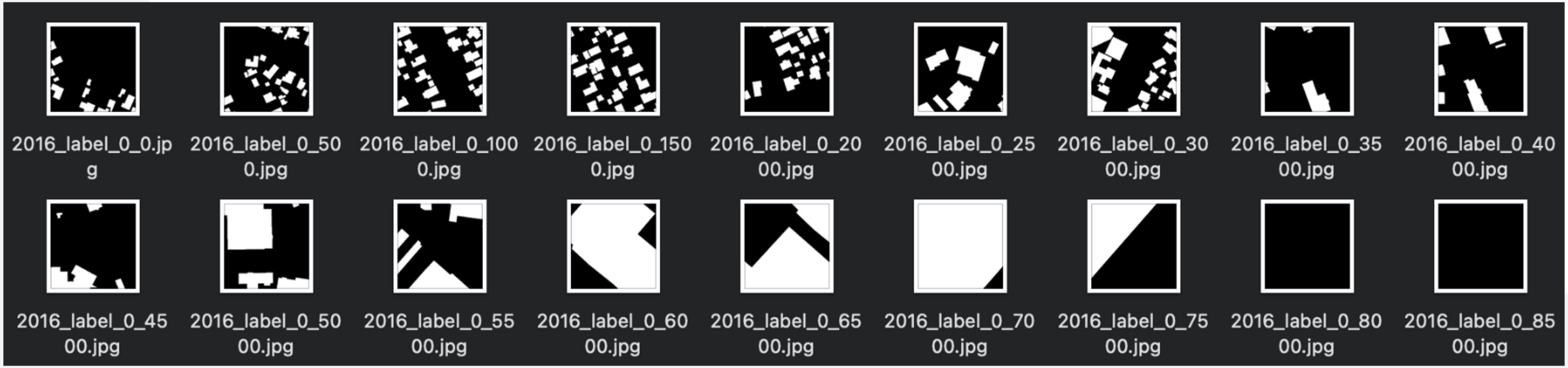


2.3 Preprocessing of Dataset (Image Cropping) 2012



2.3 Preprocessing of Dataset (Image Cropping) 2016





3. Conventional Method: Pixel-wise Comparison

https://stackoverflow.com/questions/56183201/detect-and-visualize-differences-between-two-images-with-opencv-pythonow

3. Conventional Method: Pixel-wise Comparison

https://stackoverflow.com/questions/56183201/detect-and-visualize-differences-between-two-images-with-opencv-pythonow

3.1 Advantages of Pixel-wise Comparison

- Simplicity
- Fast for small datasets
- Objective

3.2 Disadvantages of Pixel-wise Comparison

Sensitivity to noise

Slow for large datasets

Requires perfect Alignment

4. Our Approach

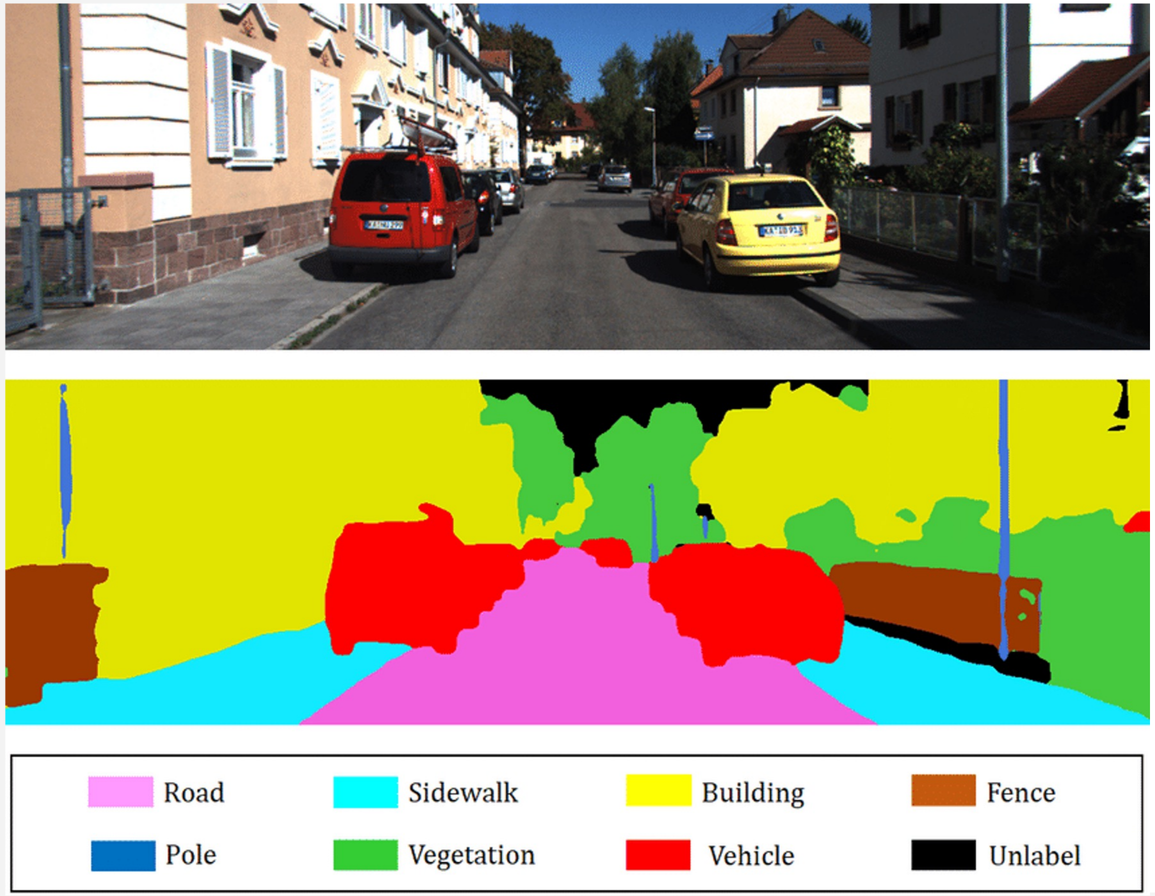
- Semantic segmentation — how we define our task

- FCN – basic network

- Skip connection – improve network performance

- Siamese network – change detection

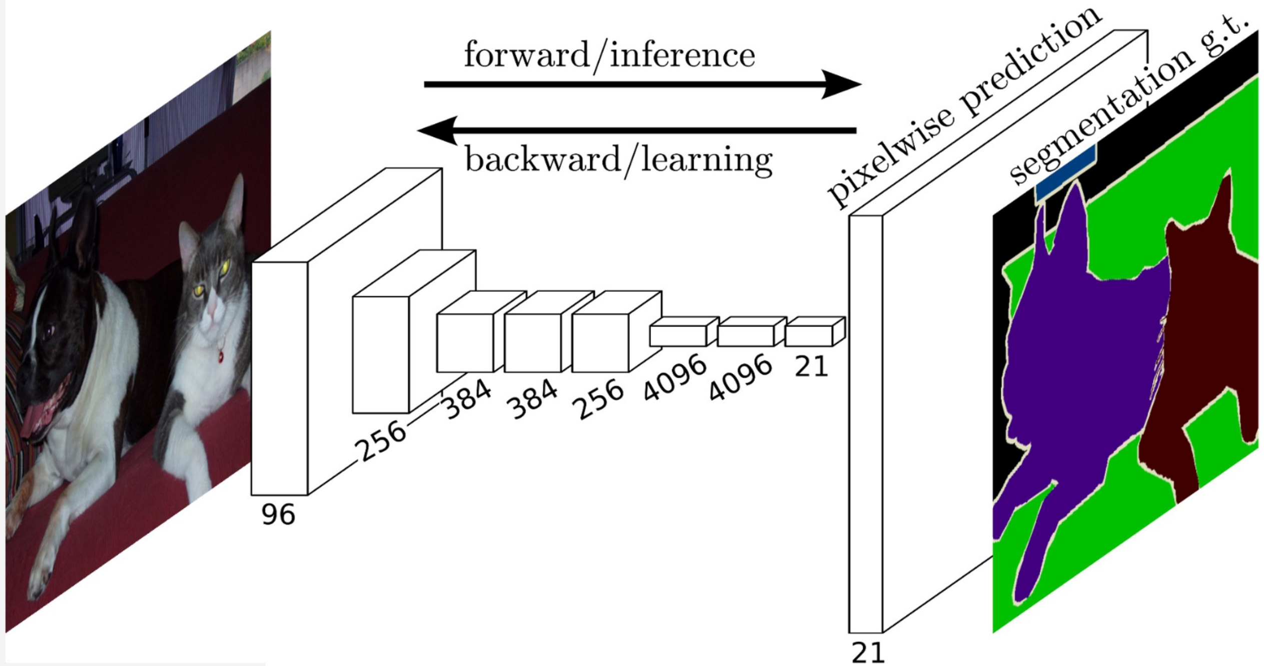
4.1 Semantic Segmentation



Identify the boundaries of each object or region and labeling every pixel accordingly

https://towardsai.net/p/l/machine-learning-7

4.2 FCN (Fully Convolutional Network)



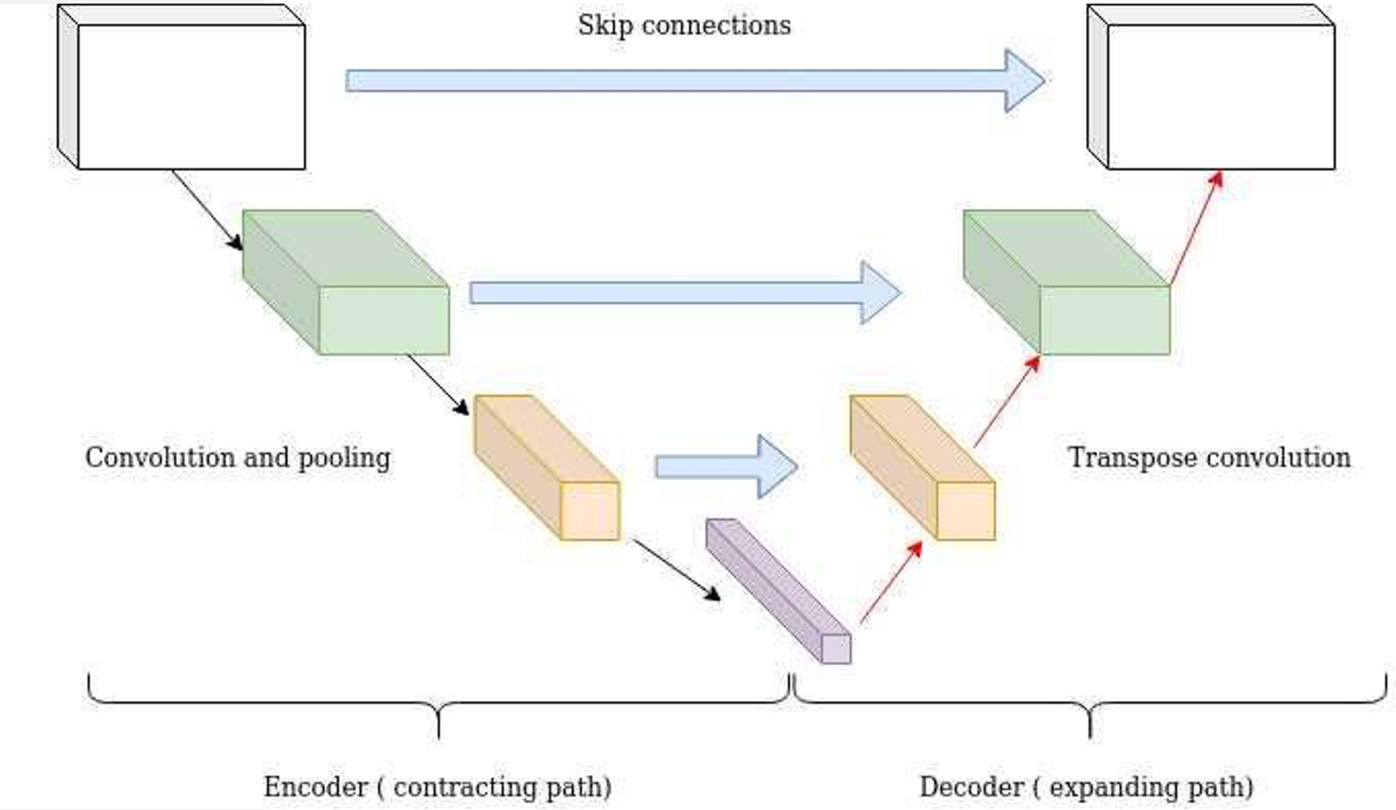
Aim at classifying each pixel in an image into a set of predefined classes or categories.

FCNs are composed entirely of convolutional layers.

FCNs allow to take input images of any size and produce output feature maps of the same spatial dimensions.

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. Retrieved May 18, 2023, from arXiv.org website: https://arxiv.org/abs/1411.4038

4.3 Skip Connection

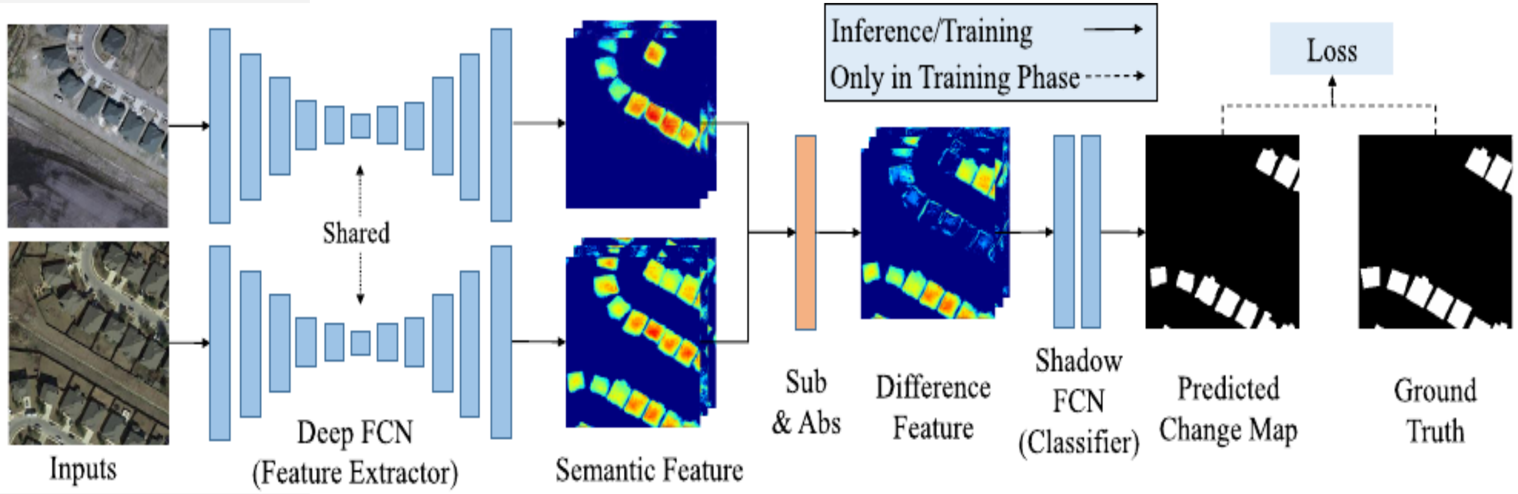


Skip Connection allows the flow of information to bypass one or more layers.

In other words, the output of a layer is "skipped" over one or more layers and added to the input of a later layer.

https://theaisummer.com/skip-connections/

4.4 Siamese Network

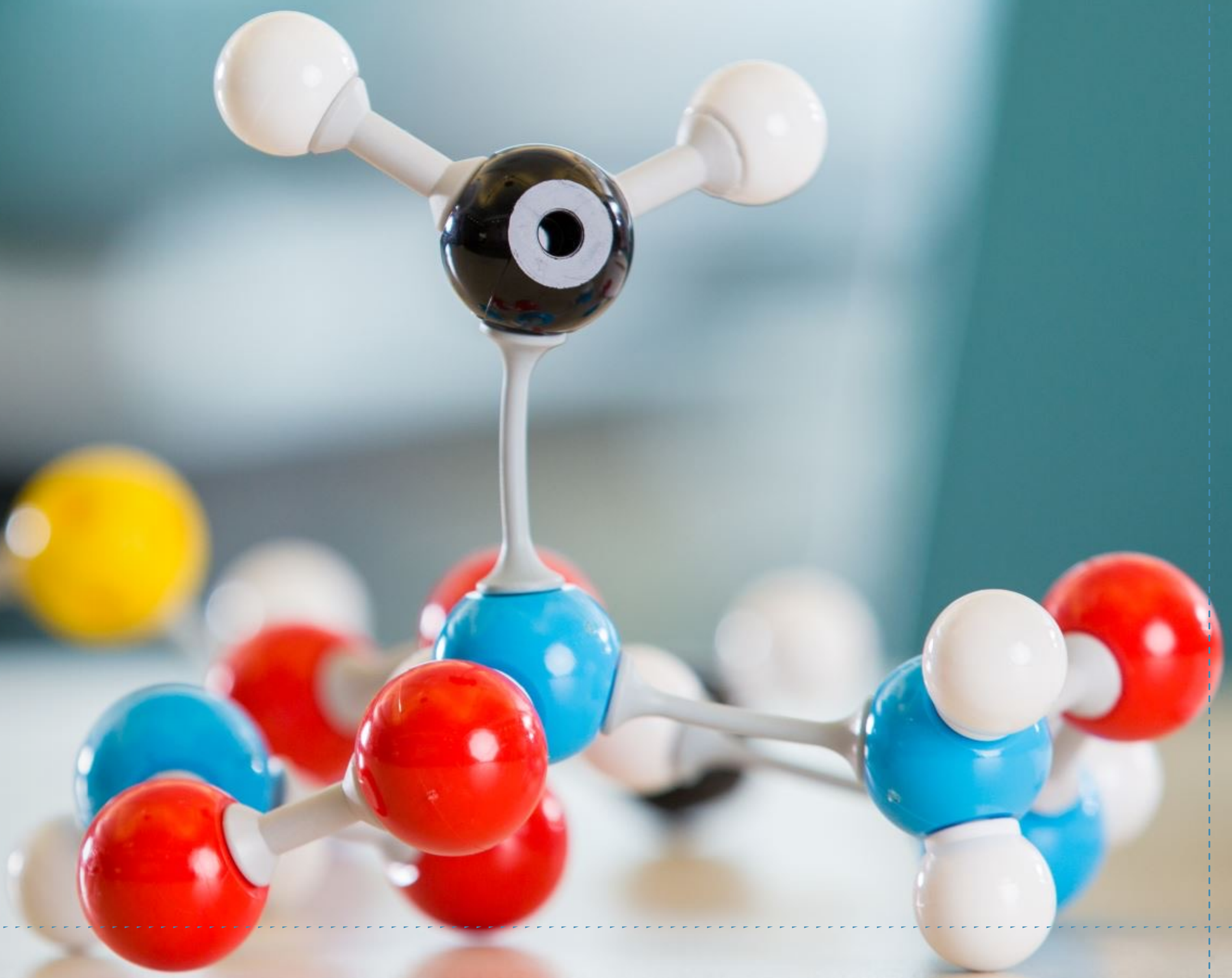


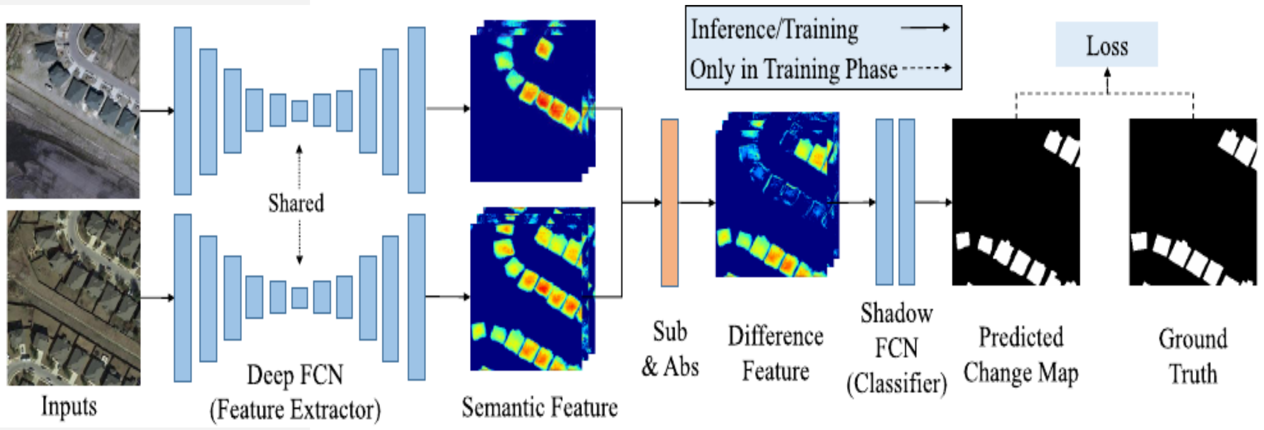
Consists of subnetworks which share the same architecture and weights.

.

[IEEE Xplore Full-Text Phttps://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9386248DF](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9386248)

5. Our model






```
-Sequential: 1-3                      [1, 1, 256, 256]      --
  └Conv2d: 2-7                        [1, 12, 256, 256]     648
  └BatchNorm2d: 2-8                   [1, 12, 256, 256]     24
  └ReLU: 2-9                          [1, 12, 256, 256]     --
  └Conv2d: 2-10                       [1, 24, 256, 256]     288
  └BatchNorm2d: 2-11                  [1, 24, 256, 256]     48
  └ReLU: 2-12                         [1, 24, 256, 256]     --
```

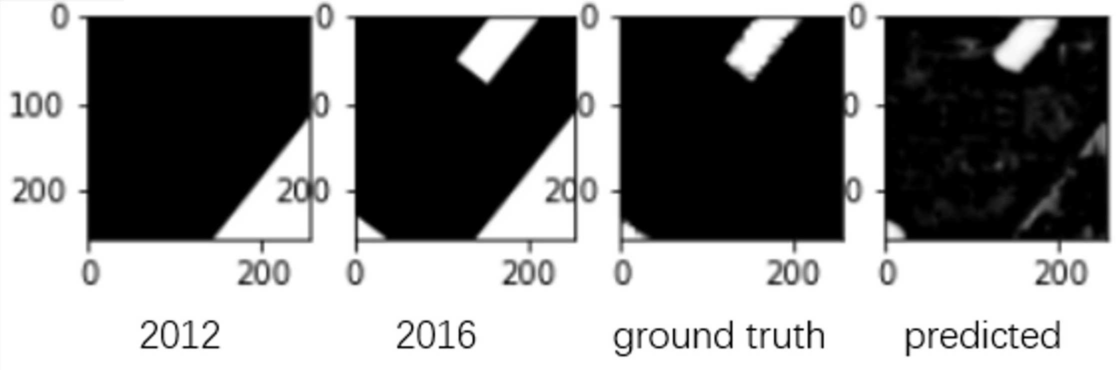
```
├─FCN: 1-1                                   [1, 3, 256, 256]      --
│    └─IntermediateLayerGetter: 2-1          [1, 2048, 32, 32]     --
│    │    └─Conv2d: 3-1                      [1, 64, 128, 128]     9,408
│    │    └─BatchNorm2d: 3-2                 [1, 64, 128, 128]     128
│    │    └─ReLU: 3-3                        [1, 64, 128, 128]     --
│    │    └─MaxPool2d: 3-4                   [1, 64, 64, 64]       --
│    │    └─Sequential: 3-5                  [1, 256, 64, 64]      215,808
│    │    └─Sequential: 3-6                  [1, 512, 32, 32]      1,219,584
│    │    └─Sequential: 3-7                  [1, 1024, 32, 32]     7,098,368
│    │    └─Sequential: 3-8                  [1, 2048, 32, 32]     14,964,736
│    └─FCNHead: 2-2                          [1, 3, 32, 32]        --
│    │    └─Conv2d: 3-9                      [1, 512, 32, 32]      9,437,184
│    │    └─BatchNorm2d: 3-10                [1, 512, 32, 32]      1,024
│    │    └─ReLU: 3-11                       [1, 512, 32, 32]      --
│    │    └─Dropout: 3-12                    [1, 512, 32, 32]      --
│    │    └─Conv2d: 3-13                     [1, 3, 32, 32]        1,539
│    └─FCNHead: 2-3                          [1, 3, 32, 32]        --
│    │    └─Conv2d: 3-14                     [1, 256, 32, 32]      2,359,296
│    │    └─BatchNorm2d: 3-15                [1, 256, 32, 32]      512
│    │    └─ReLU: 3-16                       [1, 256, 32, 32]      --
│    │    └─Dropout: 3-17                    [1, 256, 32, 32]      --
│    │    └─Conv2d: 3-18                     [1, 3, 32, 32]        771
```

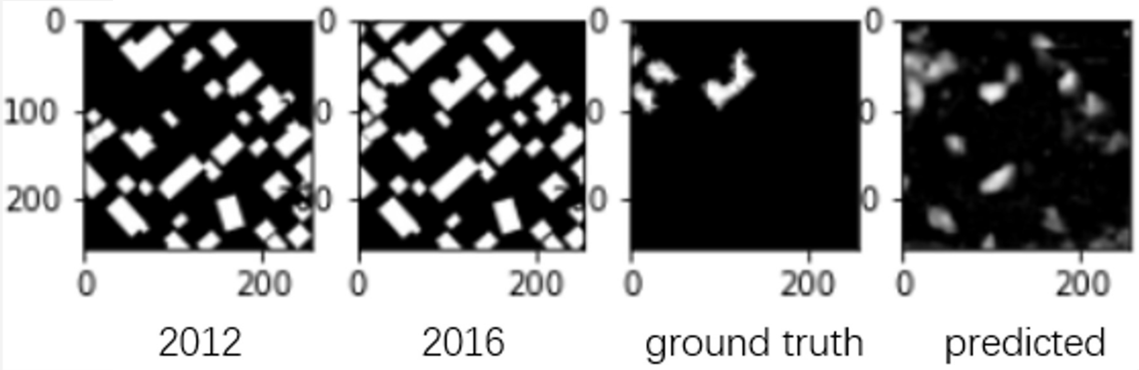
IEEE Xplore Full-Text Phttps://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9386248DF

A starting point

Using two satellite images as input,
this model gives very poor result,
binary cross entropy about 0.66

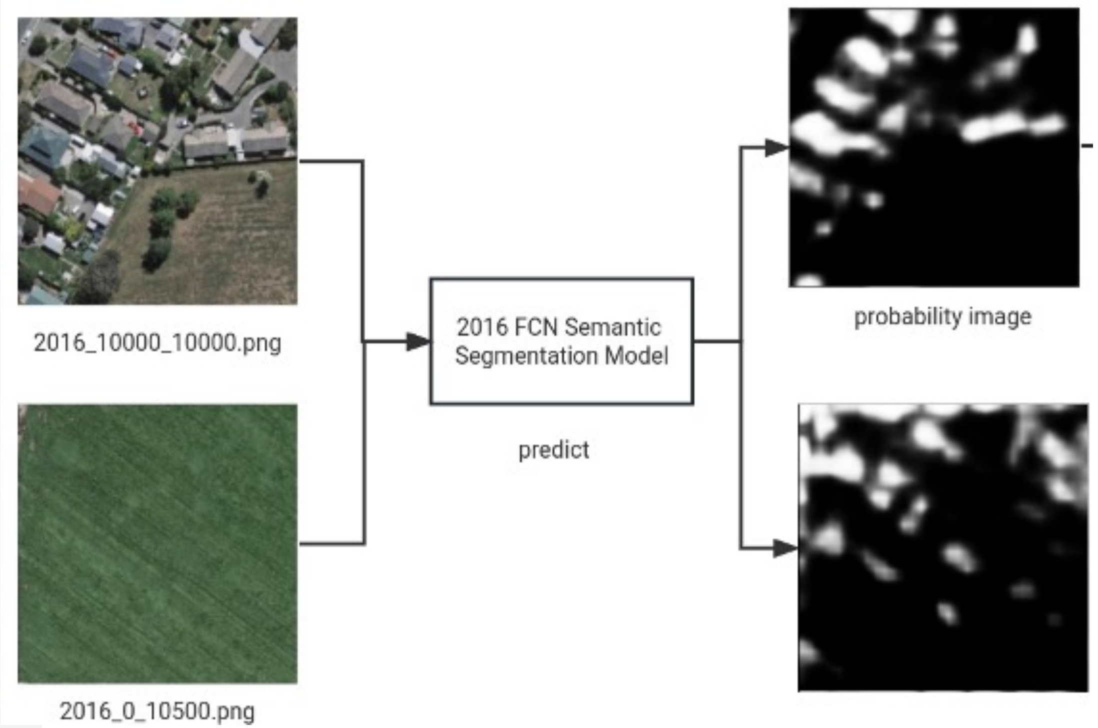
A simpler task: change between labels

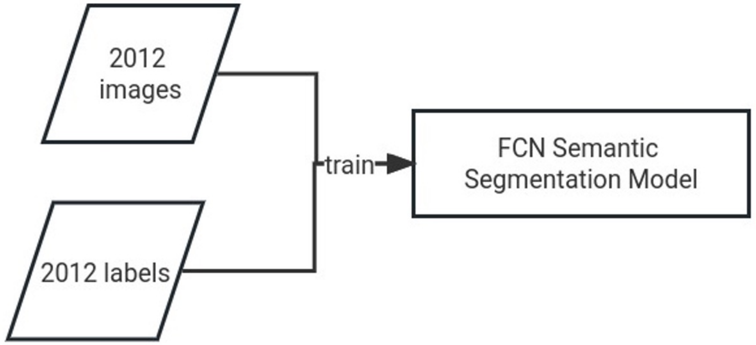




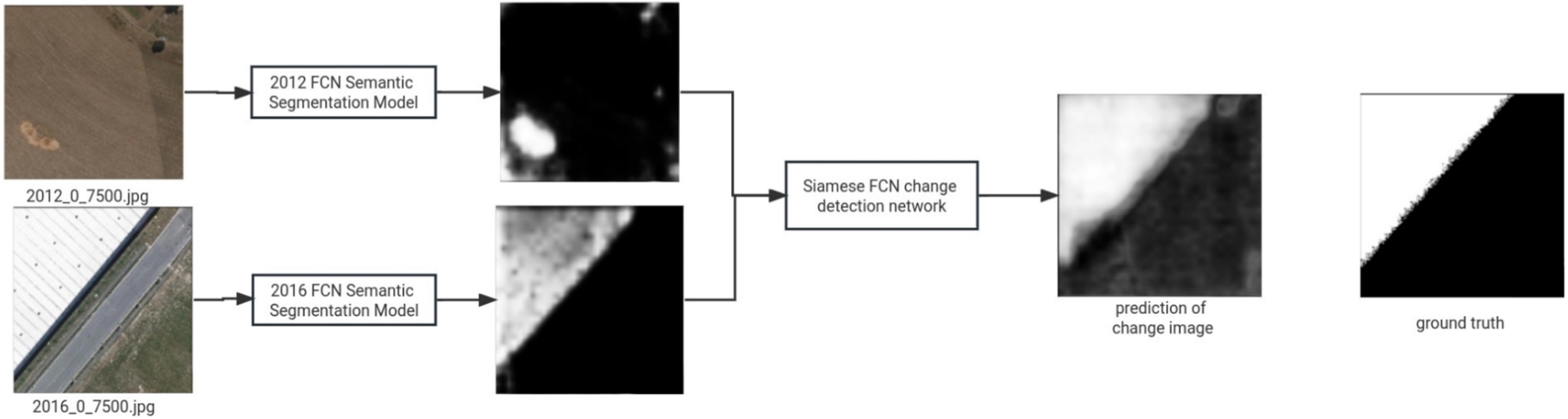
Semantic Segmentation

Train two separate model for semantic segmentation

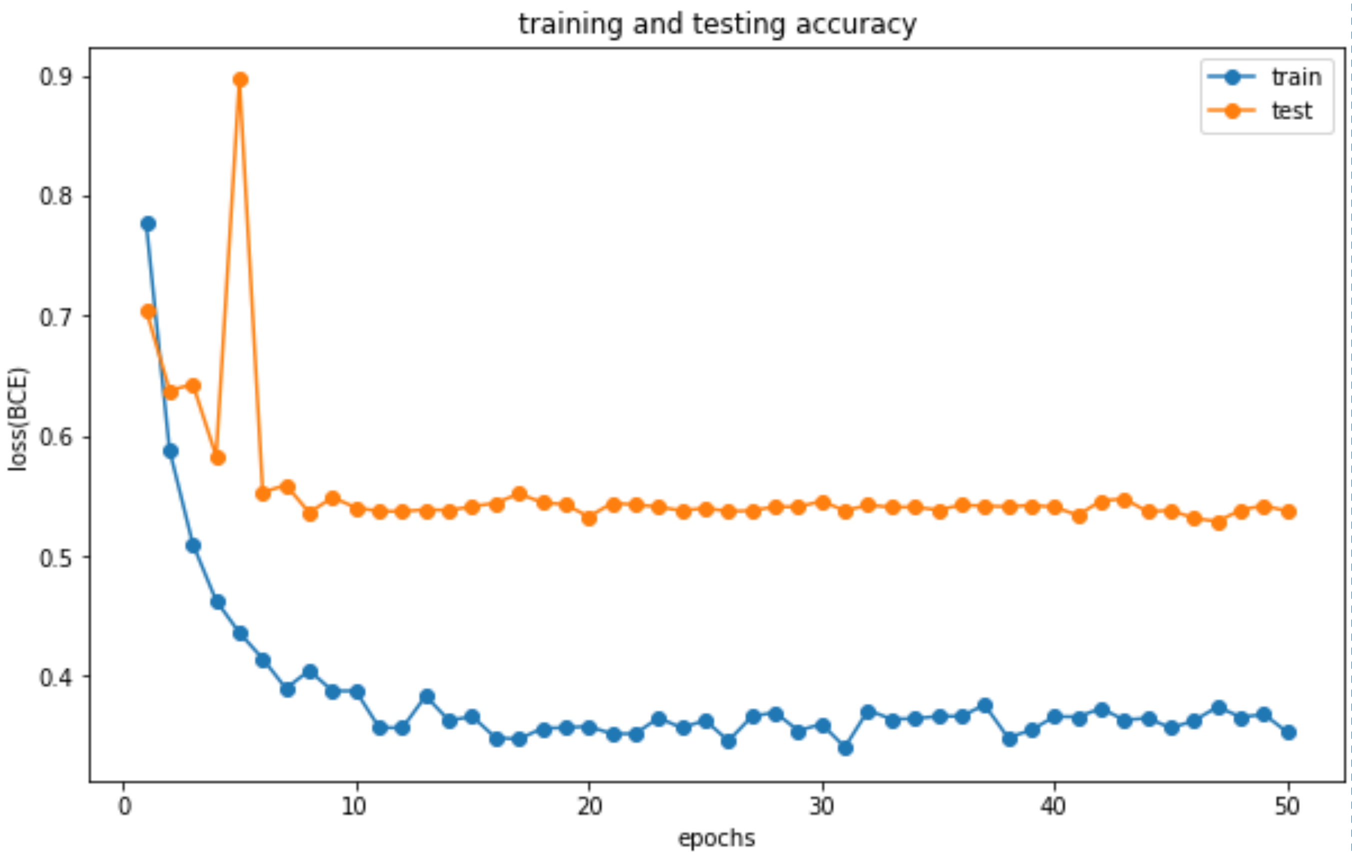




Siamese FCN model for change detection



- Loss function here:
- binary cross entropy



Upsample layer

- Implementation of upsampling layer in “torchvision.models.segmentation.fcn_resnet50()”
- The built in model uses interpolation
- For better accuracy, we try
- **transposed convolution**

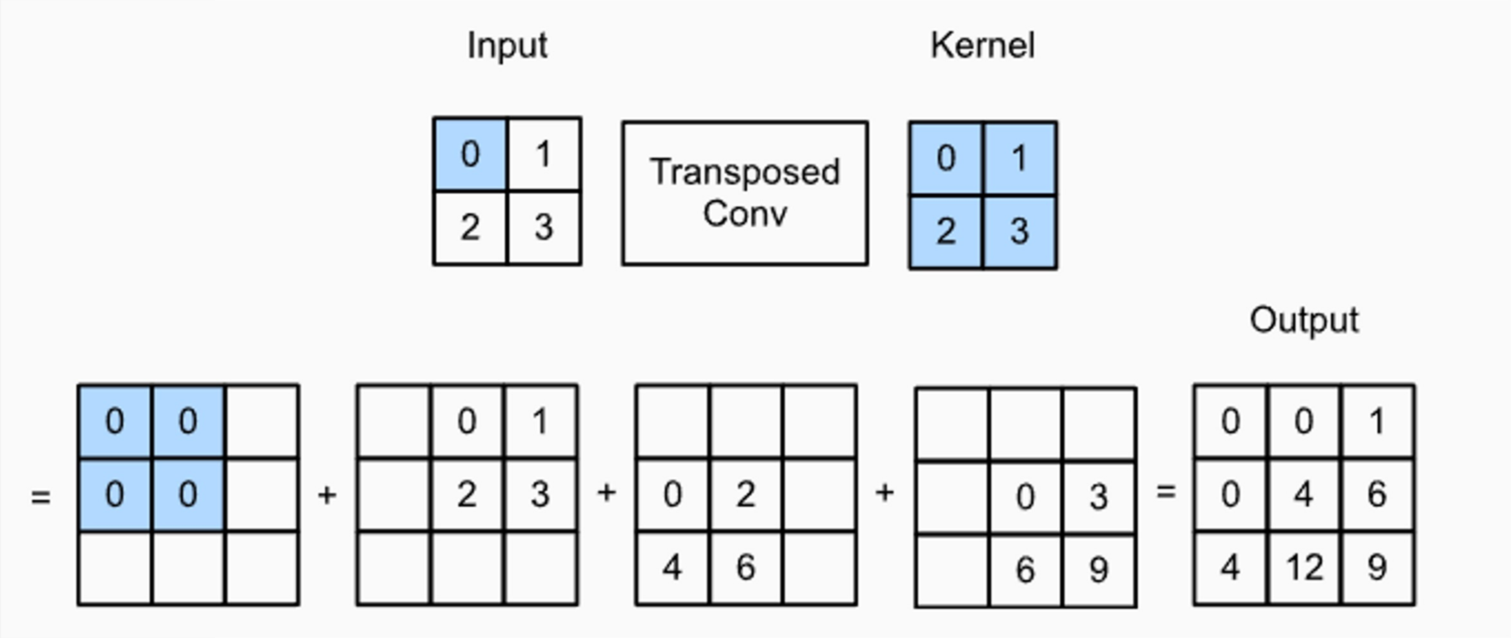
```python
def forward(self, x: Tensor) -> Dict[str, Tensor]:
    input_shape = x.shape[-2:]
    # contract: features is a dict of tensors
    features = self.backbone(x)
    result = OrderedDict()
    x = features["out"]
    x = self.classifier(x)
    x = F.interpolate(x, size=input_shape, mode="bilinear", align_corners=False)
    result["out"] = x

    if self.aux_classifier is not None:
        x = features["aux"]
        x = self.aux_classifier(x)
        x = F.interpolate(x, size=input_shape, mode="bilinear", align_corners=False)
        result["aux"] = x


    return result
```

Transposed convolution

basic transposed convolution operation with stride of 1 and no padding, ignoring channels



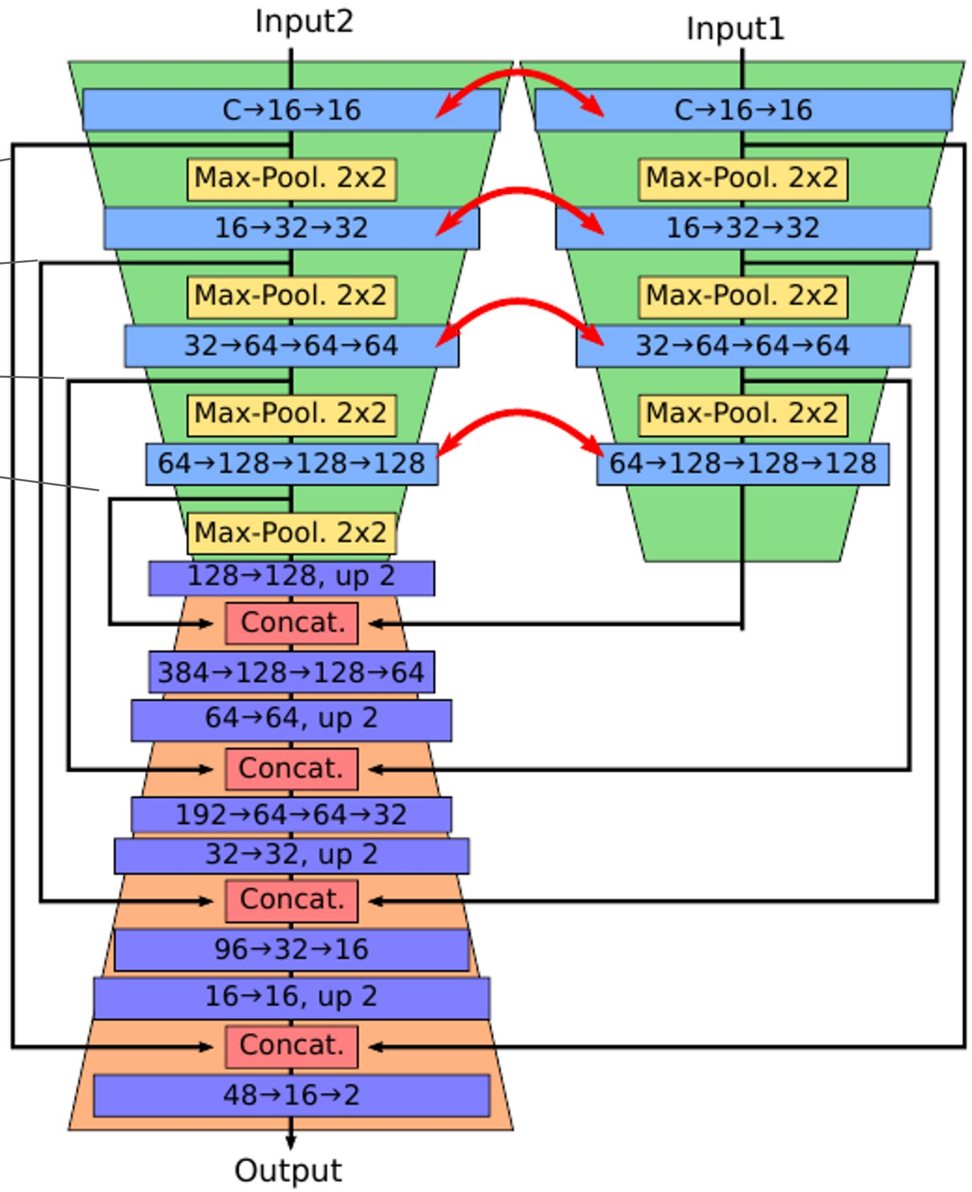
https://d2l.ai/chapter_computer-vision/transposed-conv.html

Refined Model

Skip connection

transposed convolution: in purple

In the paper, the dataset contains 13 color channels

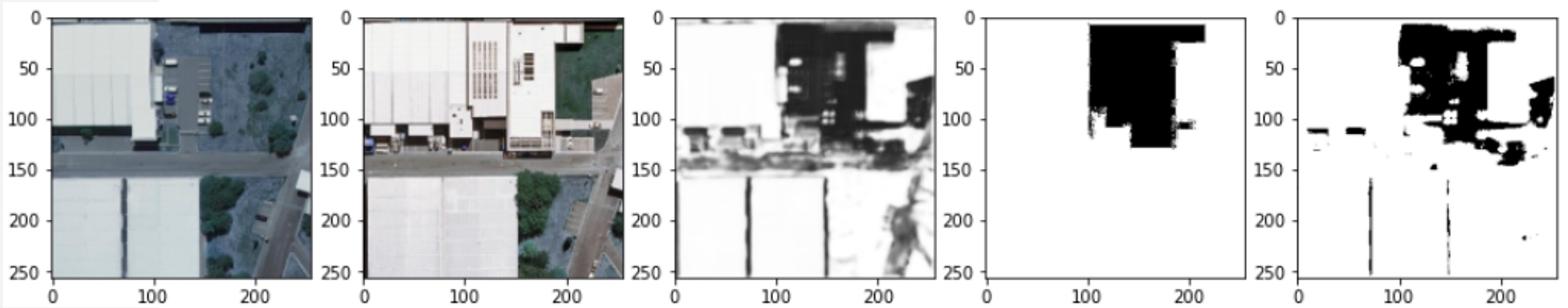


(b)FC-Siam-conc.

Rodrigo Caye Daudt, Bertrand Le Saux, and Alexandre Boulch, “Fully Convolutional Siamese Networks for Change Detection,” Oct. 2018, doi: https://doi.org/10.1109/icip.2018.8451652.

- Loss function here:
- log likelihood loss

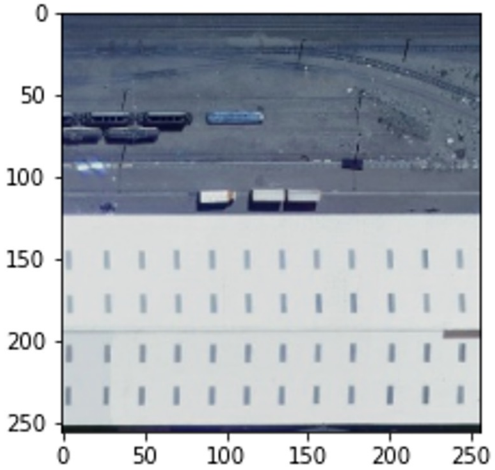
Good result



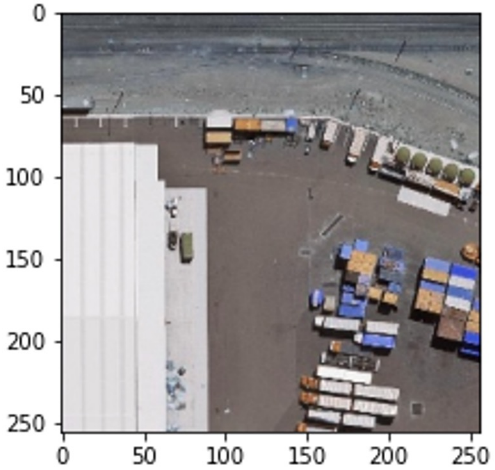
2012 image 2016 image probability image ground truth binary image

Test dataset

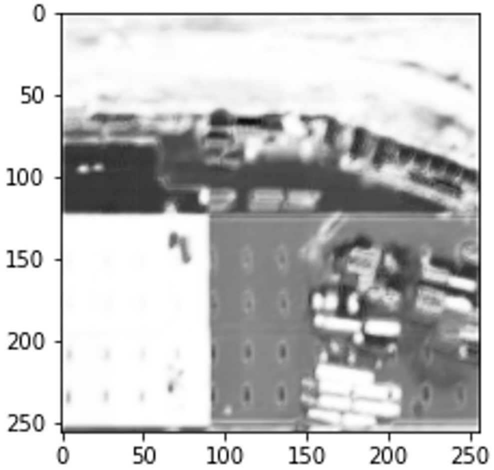
What the network learns is “change”



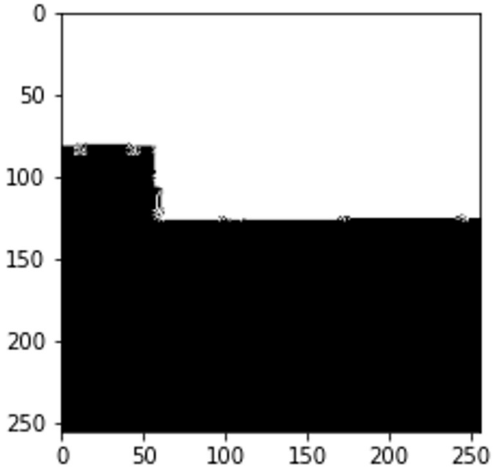
2012 image



2016 image



probability image



ground truth

6. Future work

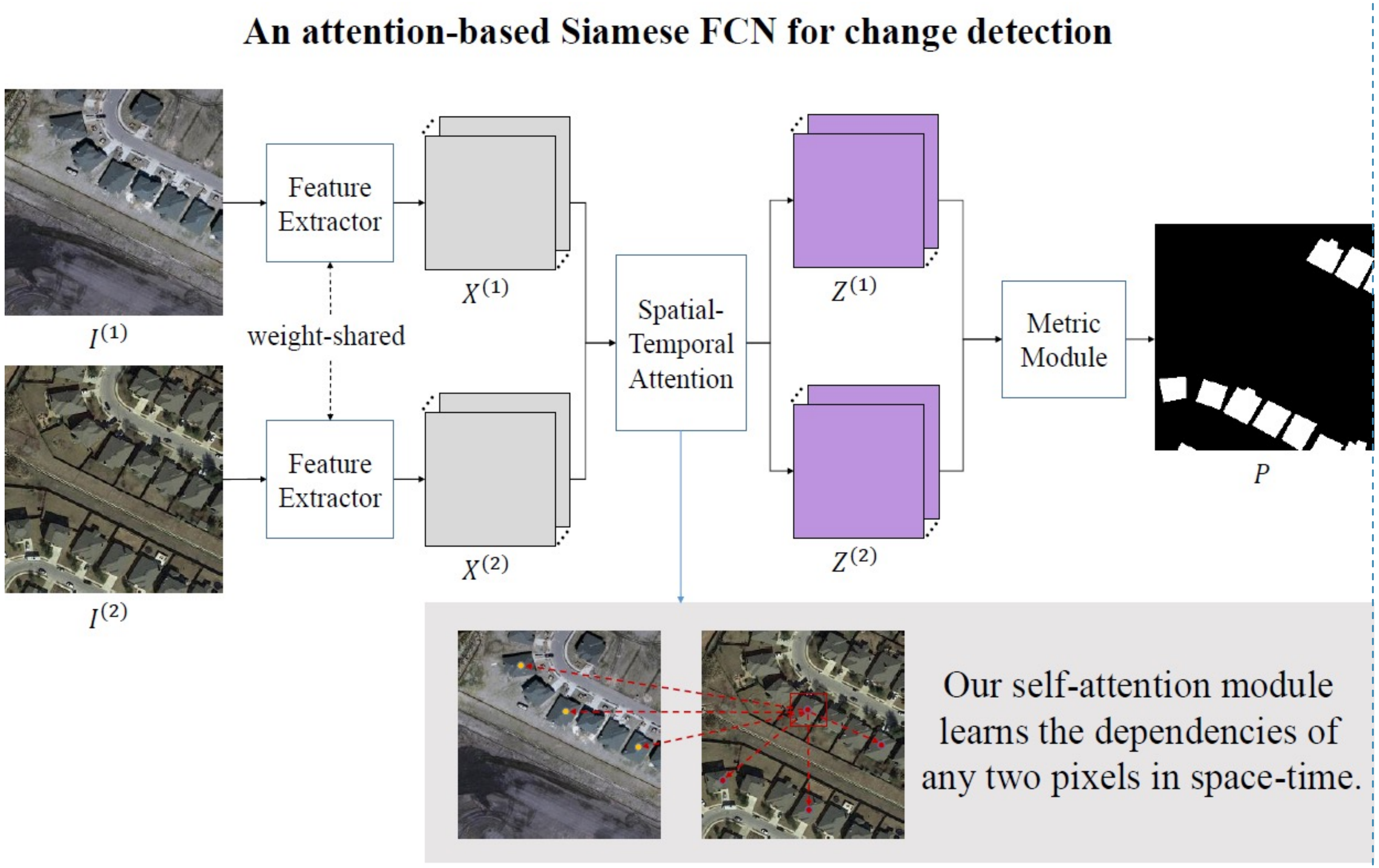
Loss function

Cross entropy may not be best choice for change detection

Try contrastive loss

6. Future work

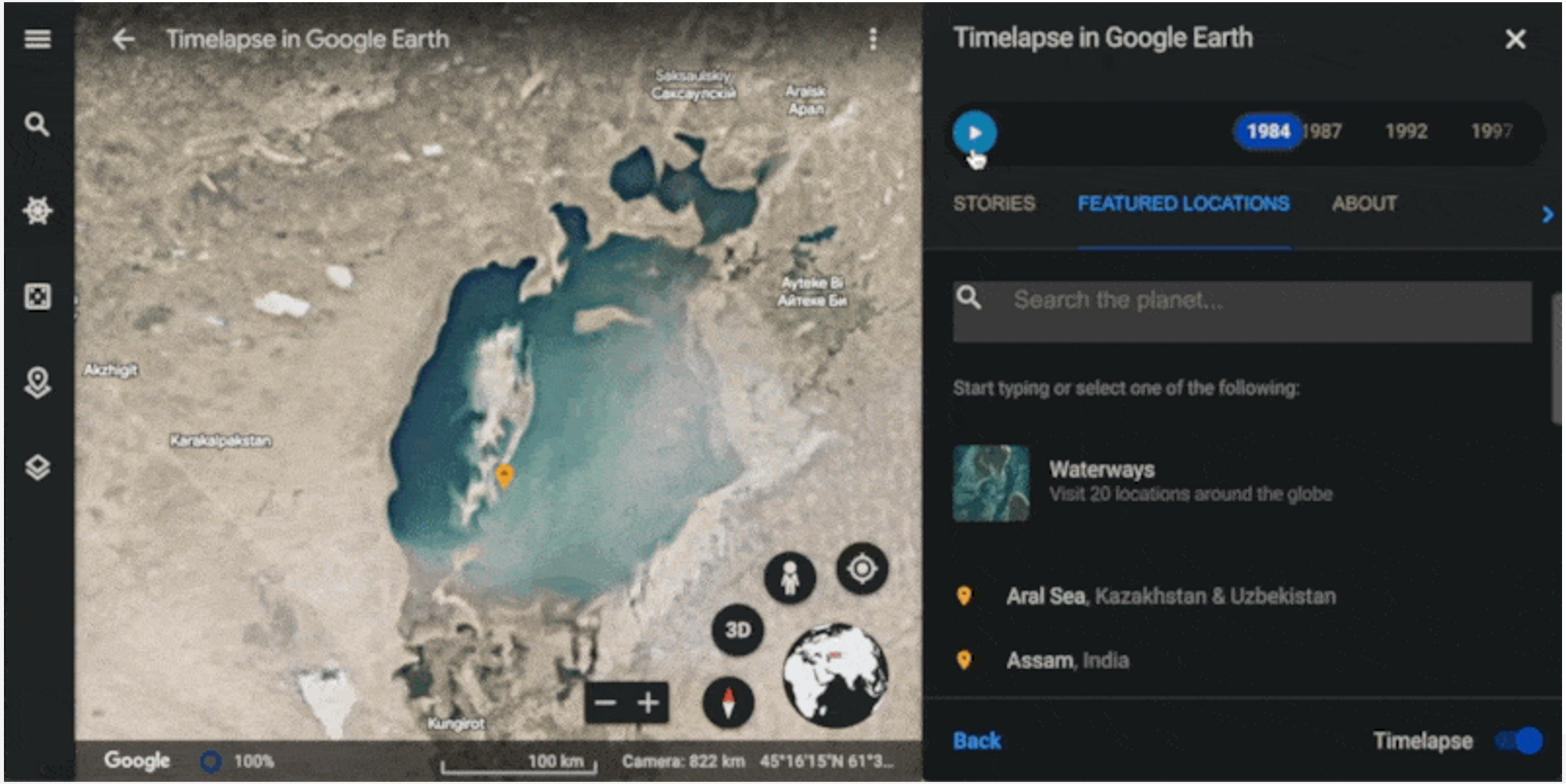
- Attention mechanism

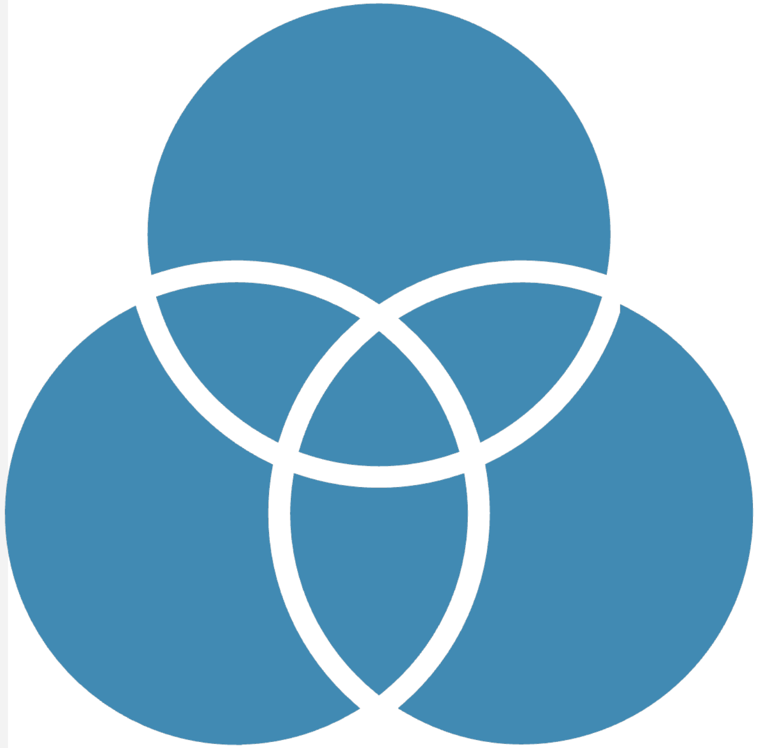


7. Other Applications: Deforestation

7. Other Applications: Glacier Retreat

7. Other Applications: Dried Lake





8. Conclusion

- We realize detecting building change by Siamese FCN model.
- Improve the performance of model by adding skip connection and transposed convolution
- We will further focus on attention mechanism to implement other applications.

References

https://opendatalab.com/Building_change_detection_dataset

http://earth.google.com

https://stackoverflow.com/questions/56183201/detect-and-visualize-differences-between-two-images-with-opencv-pythonow

https://towardsai.net/p/l/machine-learning-7

https://paperswithcode.com/method/fcn

https://theaisummer.com/skip-connections/

https://www.baeldung.com/cs/siamese-networks

https://doi.org/10.1109/icip.2018.8451652

IEEE Xplore Full-Text Phttps://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9386248DF

Thank you!