



# Dissertation Defense

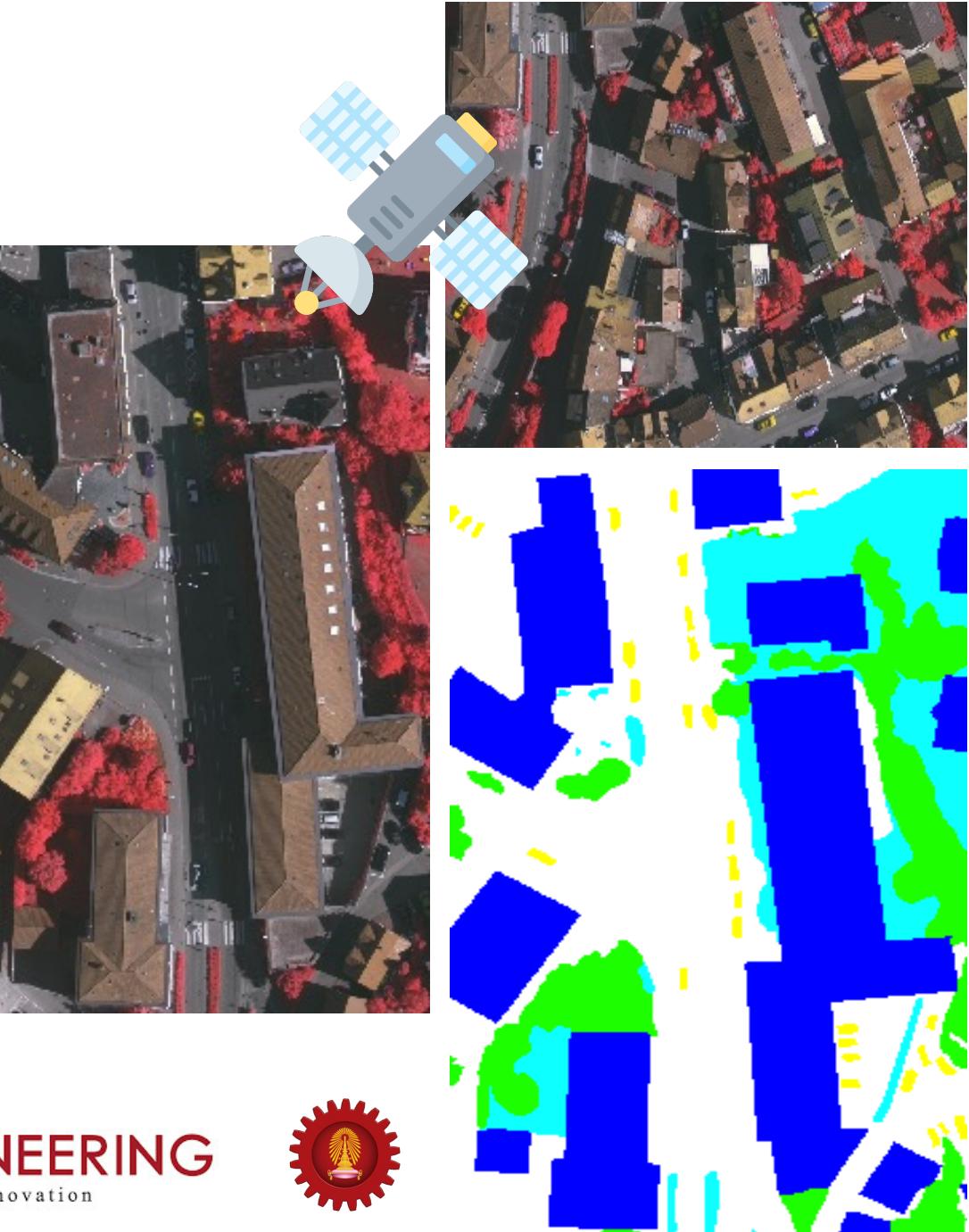
Semantic Segmentation on Remotely  
Sensed Images Using Deep Convolutional  
Encoder-Decoder Neural Network

Teerapong Panboonyuen (ID: 6071467821)

Ph.D. Candidate (Computer Engineering, Chulalongkorn University)



**GitHub** <https://kaopanboonyuen.github.io>



# Doctoral Dissertation Defense Committee

- Boonserm Kijksirikul, Ph.D. (Chairman)
  - Tokyo Institute of Technology, Japan
- Peerapon Vateekul, Ph.D. (Mentor)
  - University of Miami, USA
- Ekapol Chuangsawanich, Ph.D. (Examiner)
  - Massachusetts Institute of Technology (MIT), USA
- Thanarat Chalidabhongse, Ph.D. (Examiner)
  - University of Maryland, College Park, USA
- Kulsawasd Jitkajornwanich, Ph.D. (Examiner)
  - University of Texas at Arlington, USA
- Siam Lawawirojwong, Ph.D. (External Examiner)
  - Michigan State University, USA
- Panu Srestasathiern, Ph.D. (External Examiner)
  - The Ohio State University, USA



UNIVERSITY OF  
TEXAS  
ARLINGTON



**Massachusetts  
Institute of  
Technology**



THE OHIO STATE  
UNIVERSITY



MICHIGAN STATE  
UNIVERSITY

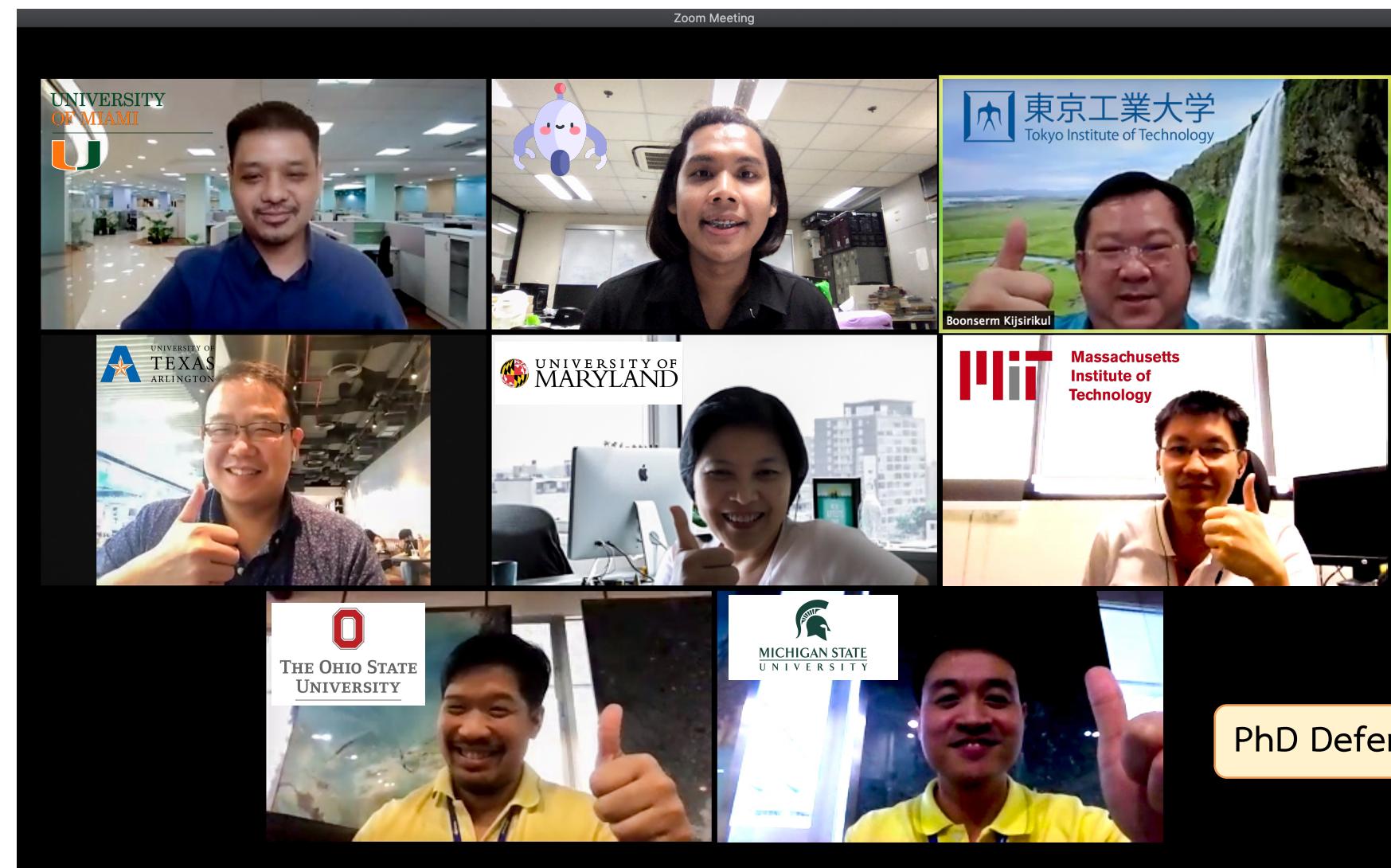
Teerapong Panboonyuen successfully defended his Ph.D. dissertation on July 9, 2020, at the Department of Computer Engineering, Chulalongkorn University, before a distinguished committee of Thai scholars with international doctoral credentials.

# Moments from my PhD Defense Day

 PhD Defense  
Accomplished!

Grateful for the support of my committee during this journey, especially navigating it over Zoom due to COVID-19.

Excited to officially be on the other side and ready for what's next in AI!



PhD Defense Date: July 9, 2020

Teerapong Panboonyuen successfully defended his Ph.D. dissertation on July 9, 2020,  
at the Department of Computer Engineering, Chulalongkorn University,  
before a distinguished committee of Thai scholars with international doctoral credentials.

# Outline

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Outline | Introduction

- **Introduction**
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Introduction

- Semantic segmentation of **remotely-sensed corpora**
  - Aerial (or **Very-High Resolution**, VHR) images
  - Satellite (or **Medium-Resolution**, MR) images
- **Convolution Neural Network (CNNs)**
  - Classification of images has becomes very efficient and smart
  - Can **create the pre-trained** deep CNNs with fixed parameters are transferred for remote scene classification
  - **Overcomes the traditional method** (K-means, Neural Nets) on Remote Sensing corpora



# Introduction

- Semantic segmentation of **remotely-sensed corpora**
  - Aerial (or **Very-High Resolution**, VHR) images
  - Satellite (or **Medium-Resolution**, MR) images
- Convolution Neural Network (CNNs)
  - Classification of images has becomes very efficient and smart
  - Can create the pre-trained deep CNNs with fixed parameters are transferred for remote scene classification
  - Overcomes the traditional method (K-means, Neural Nets) on Remote Sensing corpora



# Introduction

- Semantic segmentation of remotely-sensed corpora
  - Aerial (or Very-High Resolution, VHR) images
  - Satellite (or Medium-Resolution, MR) images
- Convolution Neural Network (CNNs)
  - Classification of images has becomes very efficient and smart
  - Can create the pre-trained deep CNNs with fixed parameters are transferred for remote scene classification
  - Overcomes the traditional method (K-means, Neural Nets) on Remote Sensing corpora



# Introduction (cont.)

- It has been implemented in many applications in various domains
  - Urban planning, map updates, route optimization, and navigation
  - Allowing us to better understand the domain's images and create important real-world applications
- It is mainly used for the agricultural purpose
  - Crop mapping, forest inventory, land cover
- The most widely used satellite for agriculture is LANDSAT 8
  - It contains operational land imager (OLI) and thermal infrared sensor (TIRS)
  - It covers the landmass, agriculture and remote areas



# Introduction (cont.)

- It has been implemented in many applications in various domains
  - Urban planning, map updates, route optimization, and navigation
  - Allowing us to better understand the domain's images and create important real-world applications
- **It is mainly used for the agricultural purpose**
  - Crop mapping, forest inventory, land cover
- **The most widely used satellite for agriculture is LANDSAT 8**
  - It contains operational land imager (OLI) and thermal infrared sensor (TIRS)
  - It covers the landmass, agriculture and remote areas

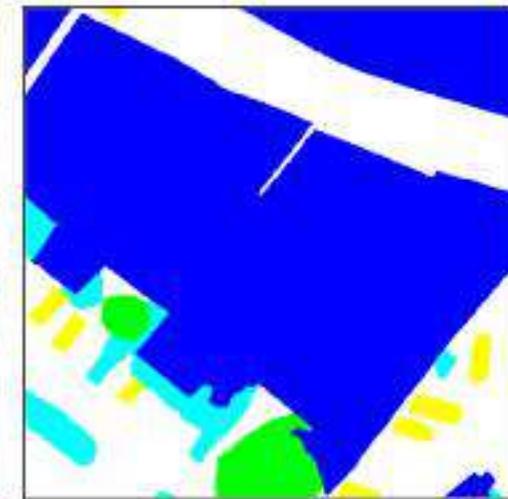


# Public and Private Corpora

## Public corpus (ISPRS Vaihingen Corpus)



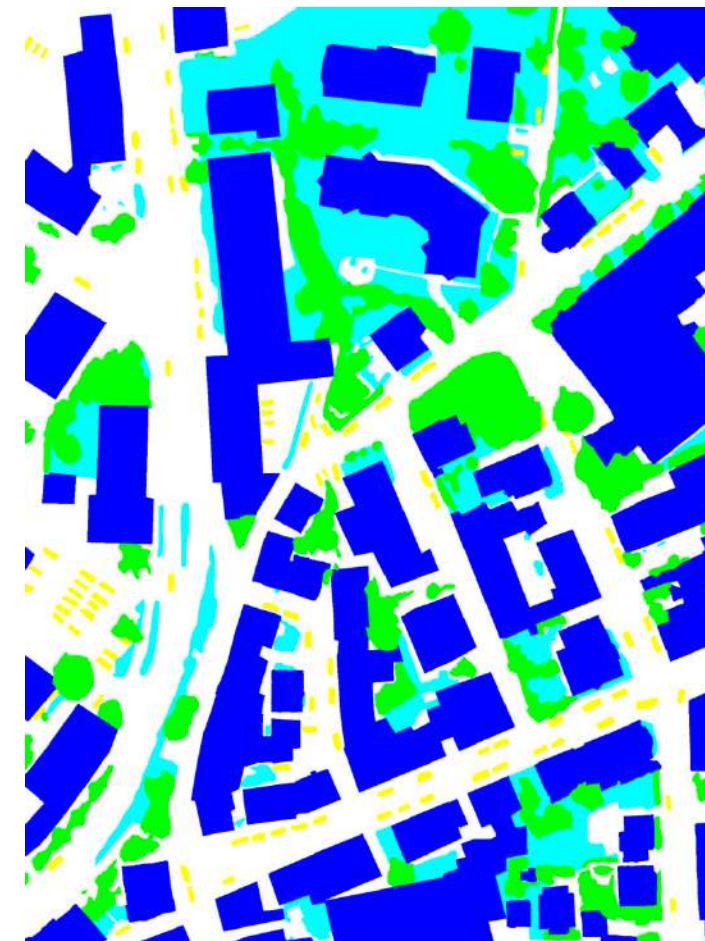
(a) image



(b) ground truth

# Public and Private Corpora

## Public corpus (ISPRS Vaihingen Corpus)



Color	Class
Yellow	Car
Blue	Building
Green	Tree
Cyan	Low Vegetation
White	Imp Surfaces
Red	Clutter

# Public and Private Corpora

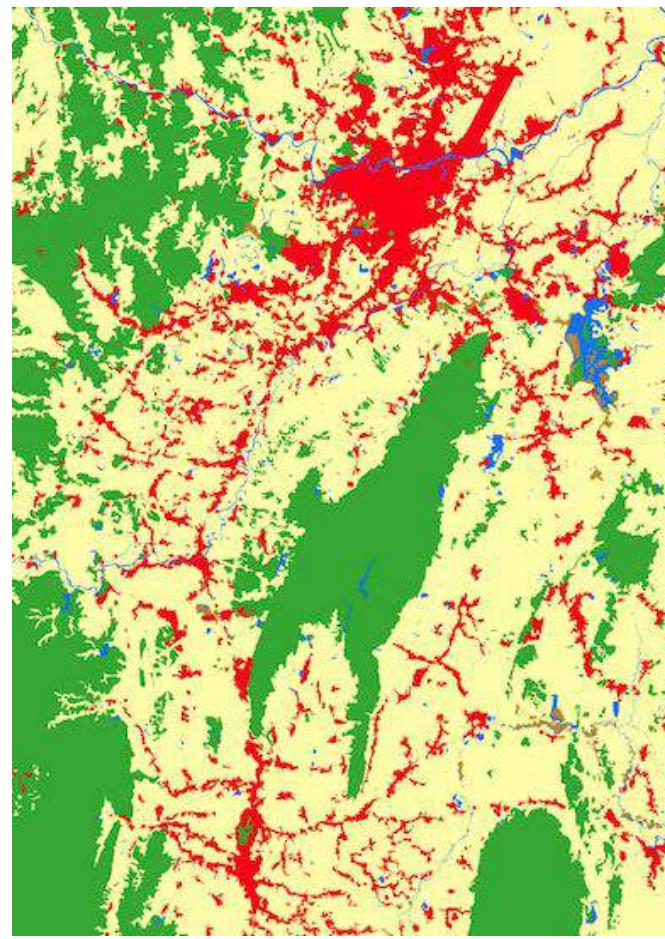
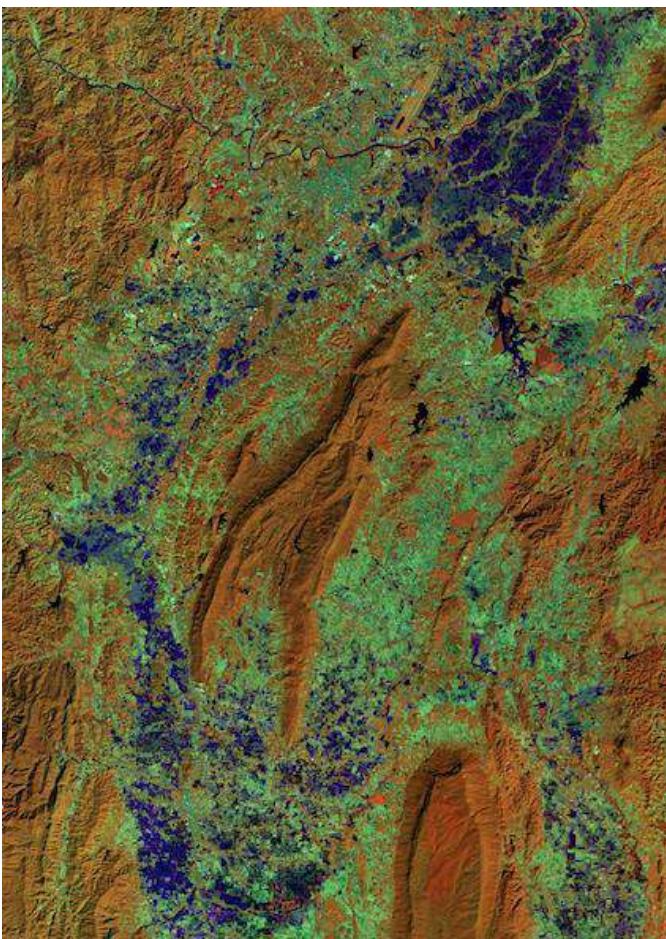
## Public corpus (ISPRS Vaihingen Corpus)

---

- There are 33 images of about  $2,500 \times 2,000$  pixels at a ground sampling distance (GSD) of about 9 cm in the image data
- We randomly split the 16 images with ground truth available
  - into a training set of 10 images and a validation set of 6 images
- 4 tiles (Image Numbers 5, 7, 23, and 30) were removed from the training set as the testing corpus

# Public and Private Corpora

## Private corpus (GISTDA Nan Province Corpus)



Color	Class
Yellow	Agriculture
Green	Forest
Brown	Miscellaneous
Red	Urban
Blue	Water

# Public and Private Corpora

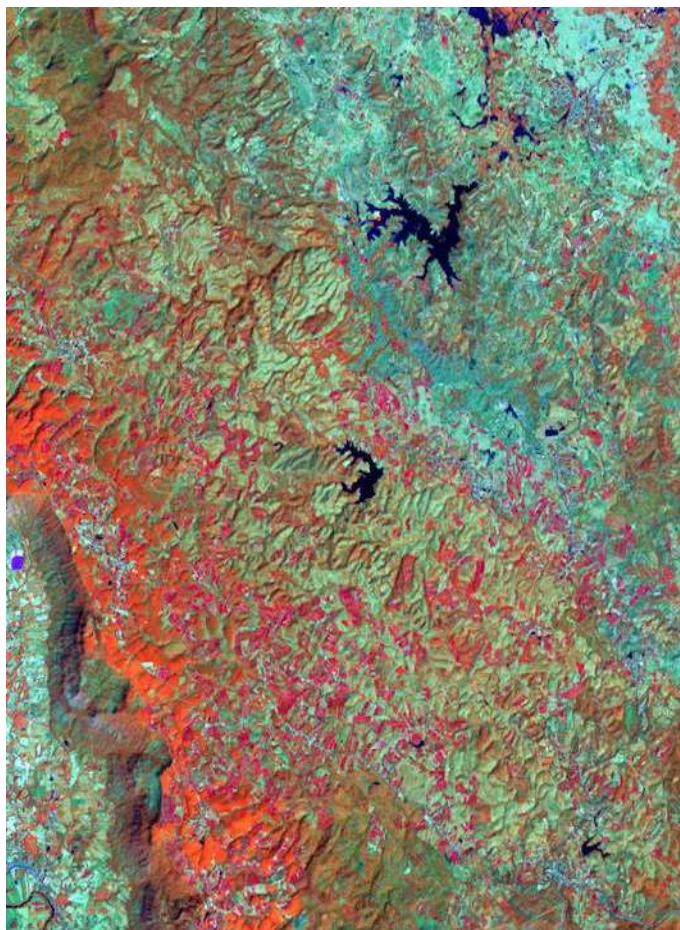


## Private corpus (GISTDA Nan Province Corpus)

- The dataset is obtained from Landsat-8 satellite consisting of 1,012 satellite images
- Bands 5, 4, and 3 are used
- Capture at Nan, a province in Thailand
- Medium resolution (16,800 × 15,800)
- The 1,012 images were split into 800 training and 112 validation images with publicly available annotation, as well as 100 testing images with annotations withheld

# Public and Private Corpora

## Private corpus (GISTDA ISAN Zone Corpus)

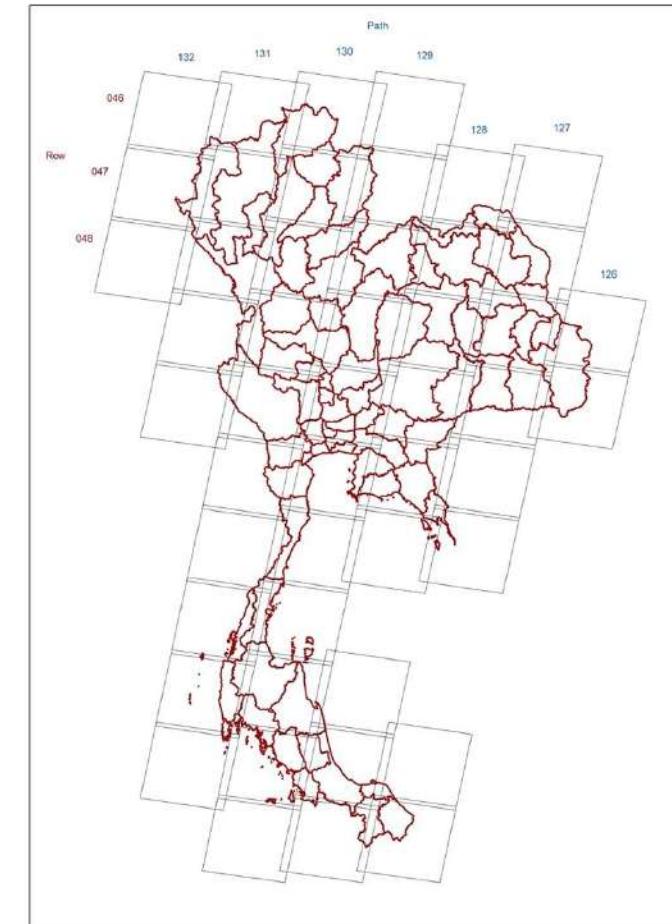
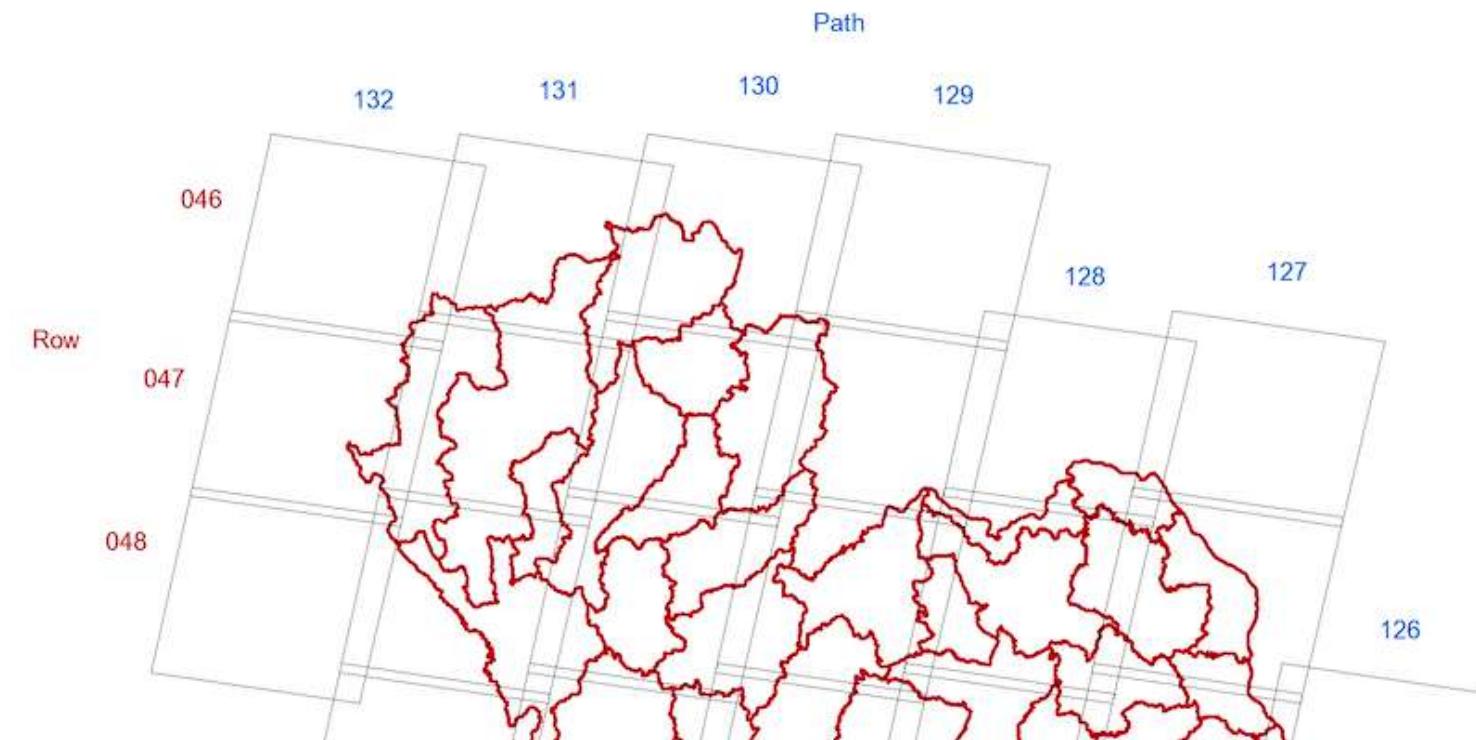


Color	Class
Yellow	Corn
Green	Pineapple
Red	Para Rubber
White	Miscellaneous

# Public and Private Corpora

## Private corpus (GISTDA ISAN Zone Corpus)

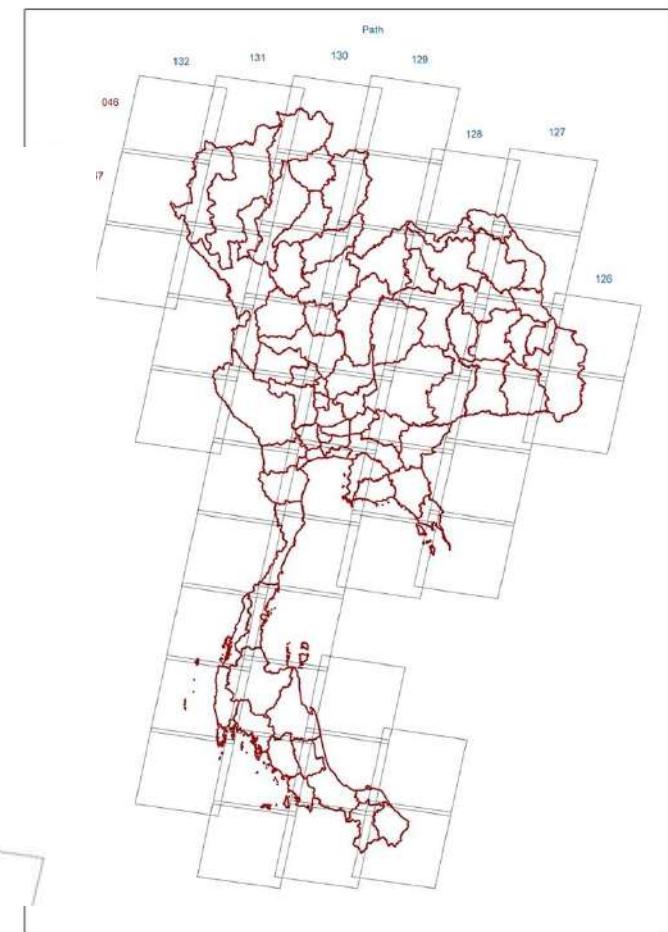
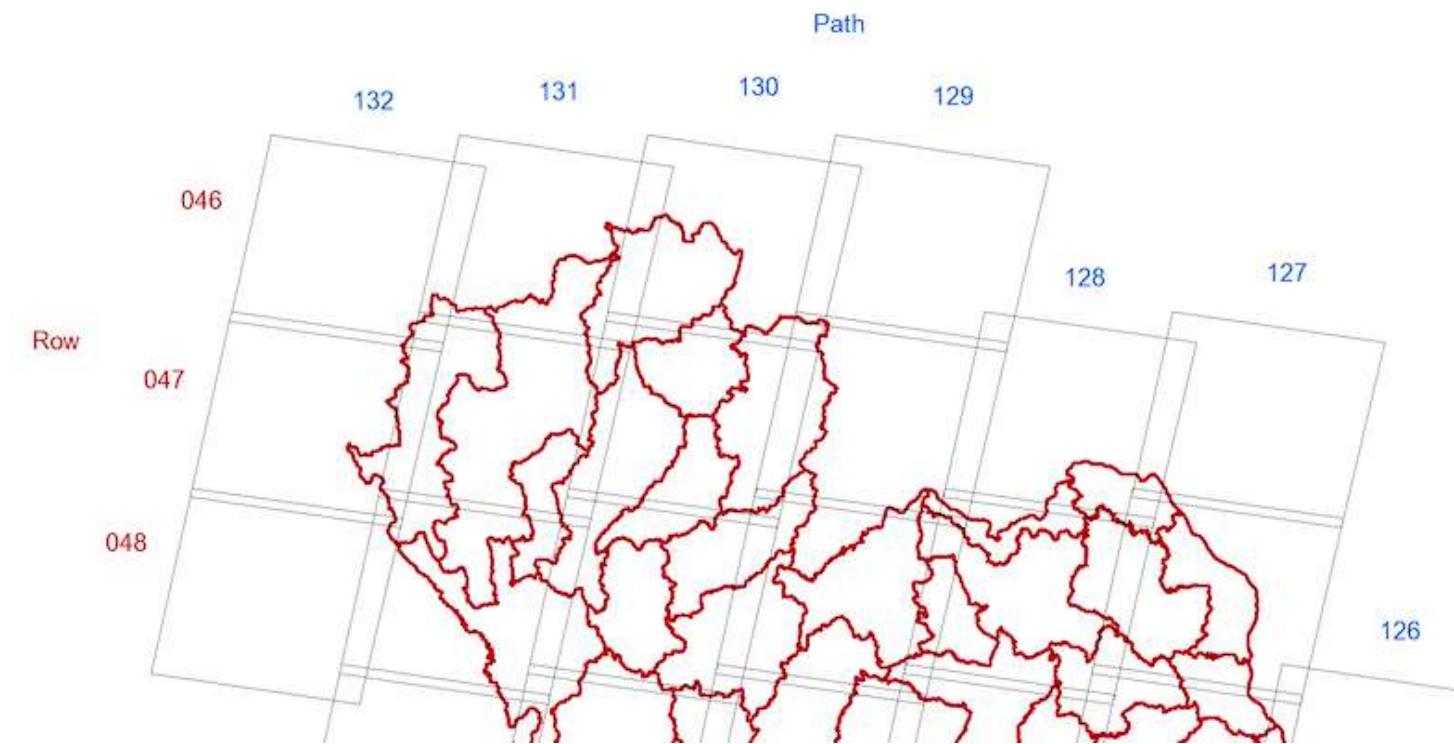
- For the Dissertation, we select **LC129048, LC130050** zone as the LC3W corpus



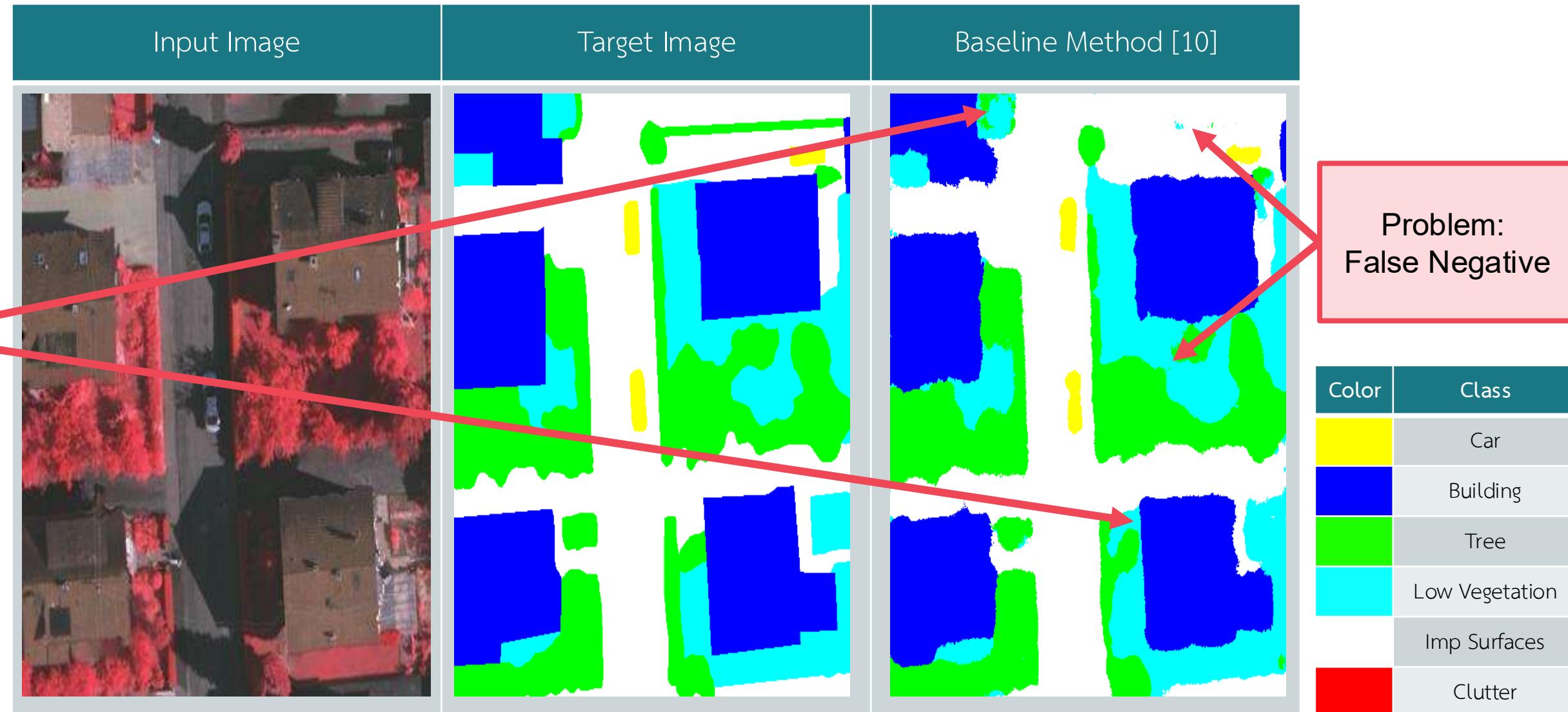
# Public and Private Corpora

## Private corpus (GISTDA ISAN Zone Corpus)

- For the Dissertation, we select **LC129048, LC130050** zone as the LC3W corpus
- Medium resolution (15,376x15,872) pixels
- 764 training
- 112 validating
- 100 testing

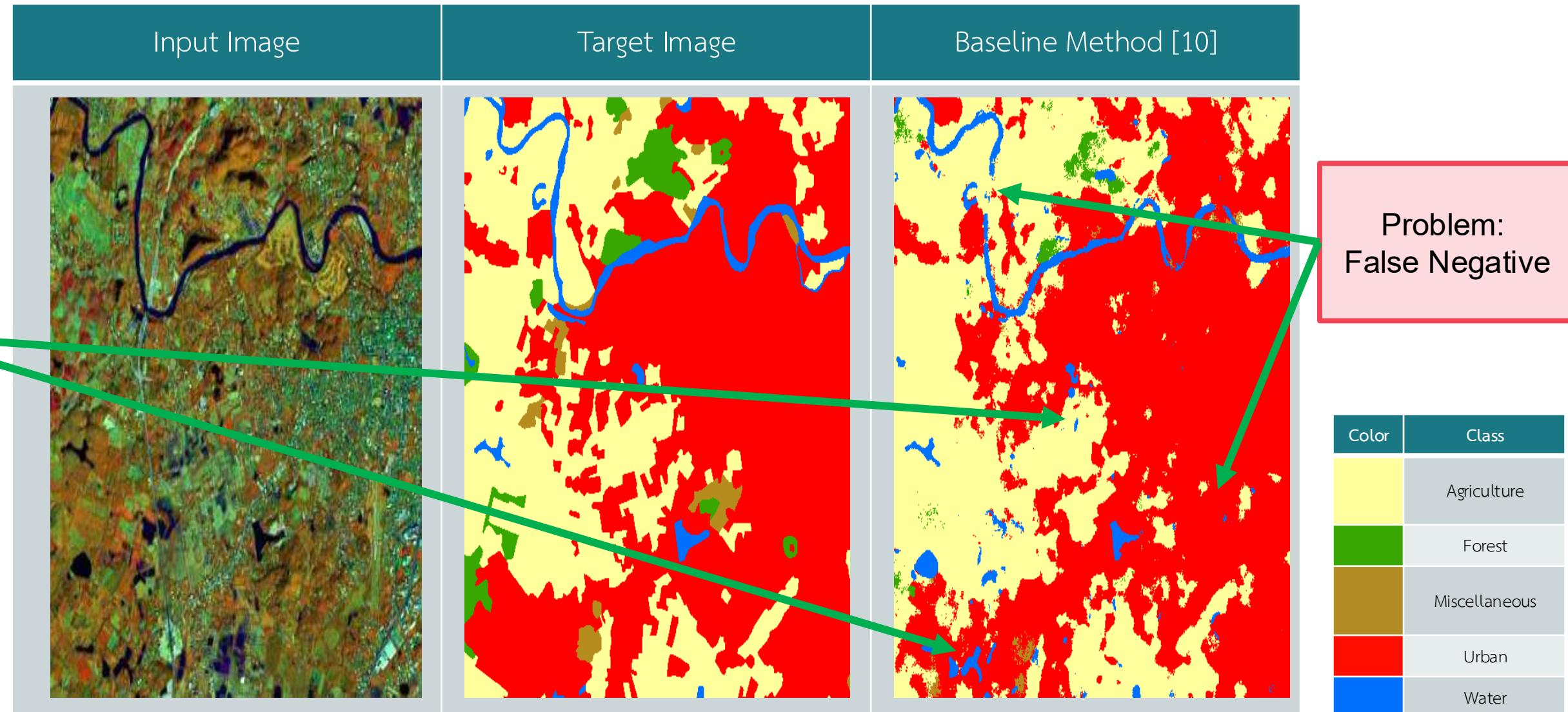


# Statement of Problem (1) Very High Resolution



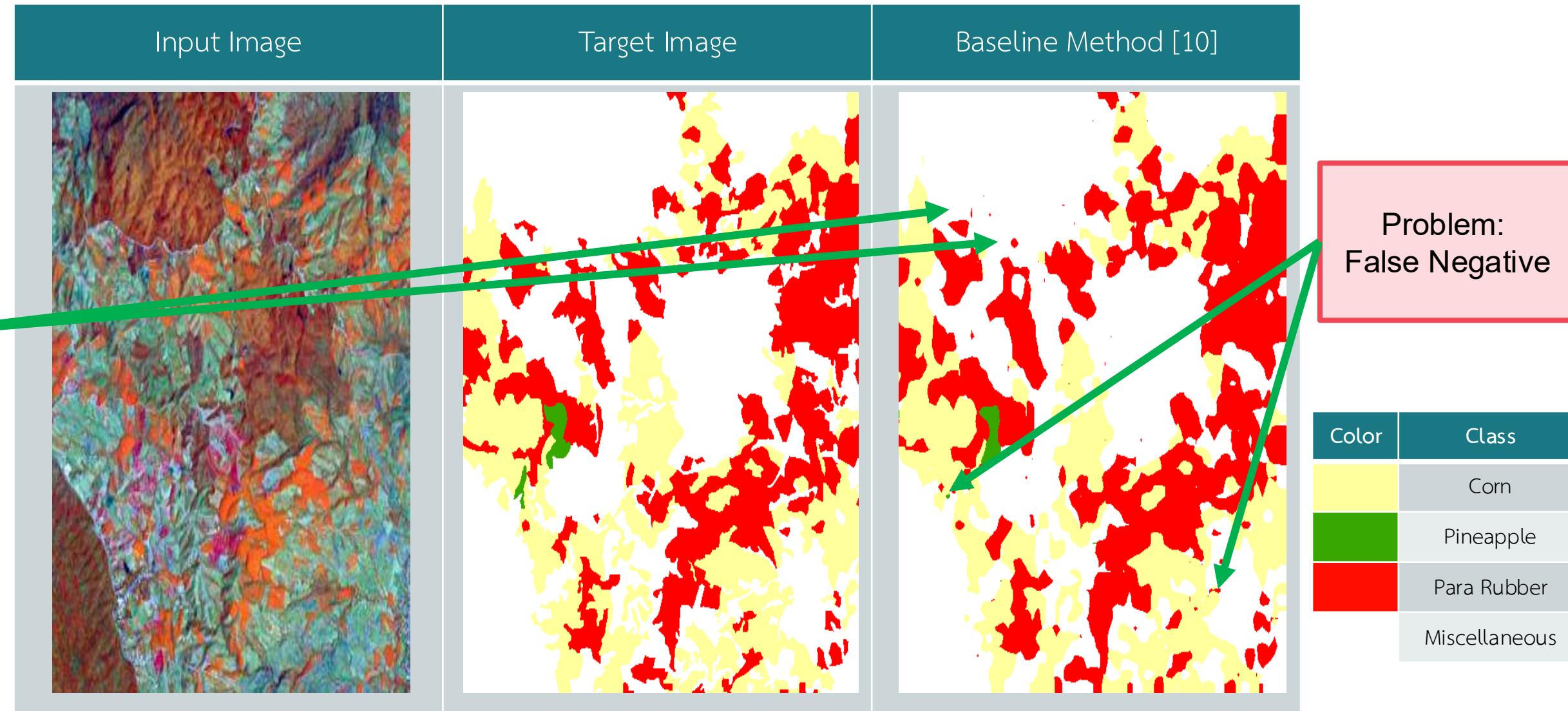
[10] Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., & Pan, C. (2018). Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 78-95.

# Statement of Problem (2) Medium Resolution



[10] Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., & Pan, C. (2018). Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 78-95.

# Statement of Problem (3) Medium Resolution



[10] Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., & Pan, C. (2018). Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 78-95.

# Statement of Problem (4)

- False Positive Problem
  - High Level (Sharp Boundary Object) such as Building Object, Rubber Tree (Zone)
- False Negative Problem
  - Rare Class (Low-Level Class) such as Water Class
- Motivation
  - This leads to some inconsistent results that suffer from accuracy performance
  - The primary challenge of this remote sensing task is a lack of training data
  - This, in fact, has become a motivation of this work

# Outline | Related Theory

- Introduction
- **Related Theory**
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

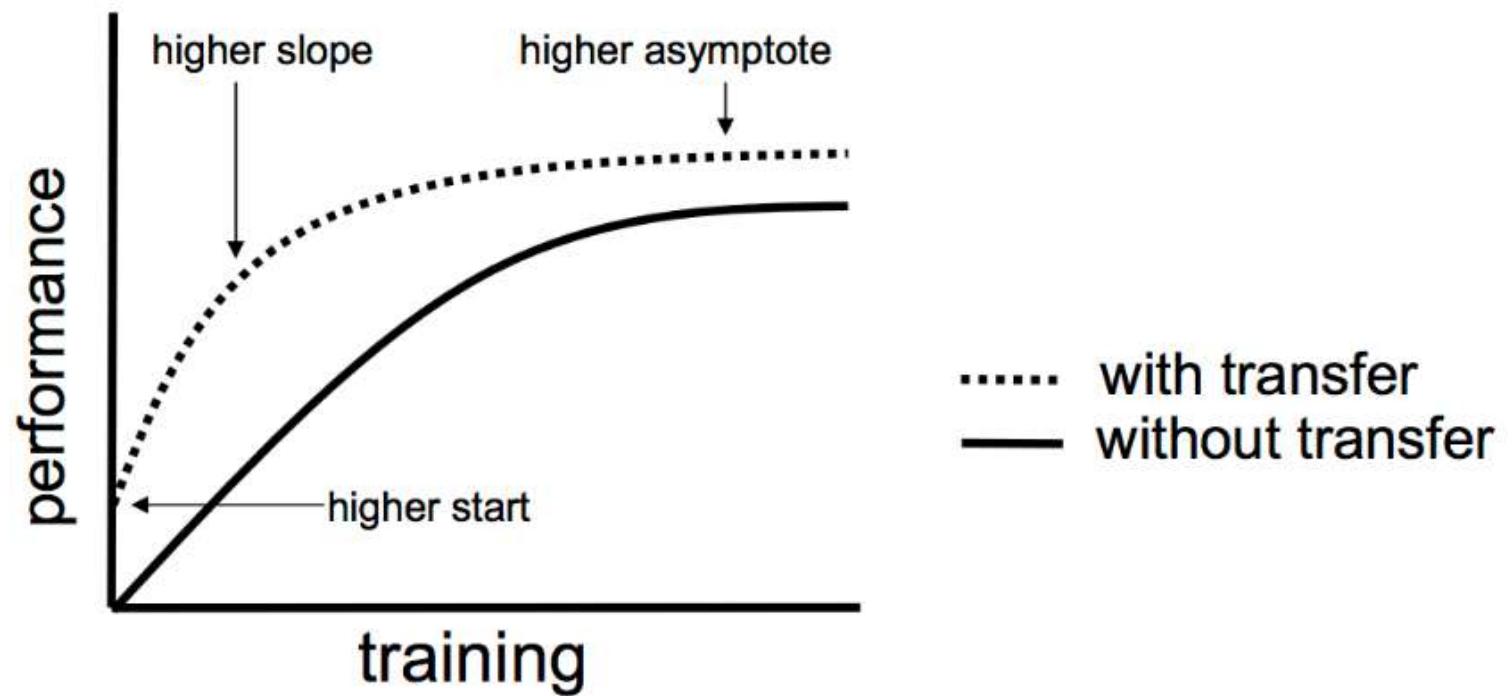
# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned."

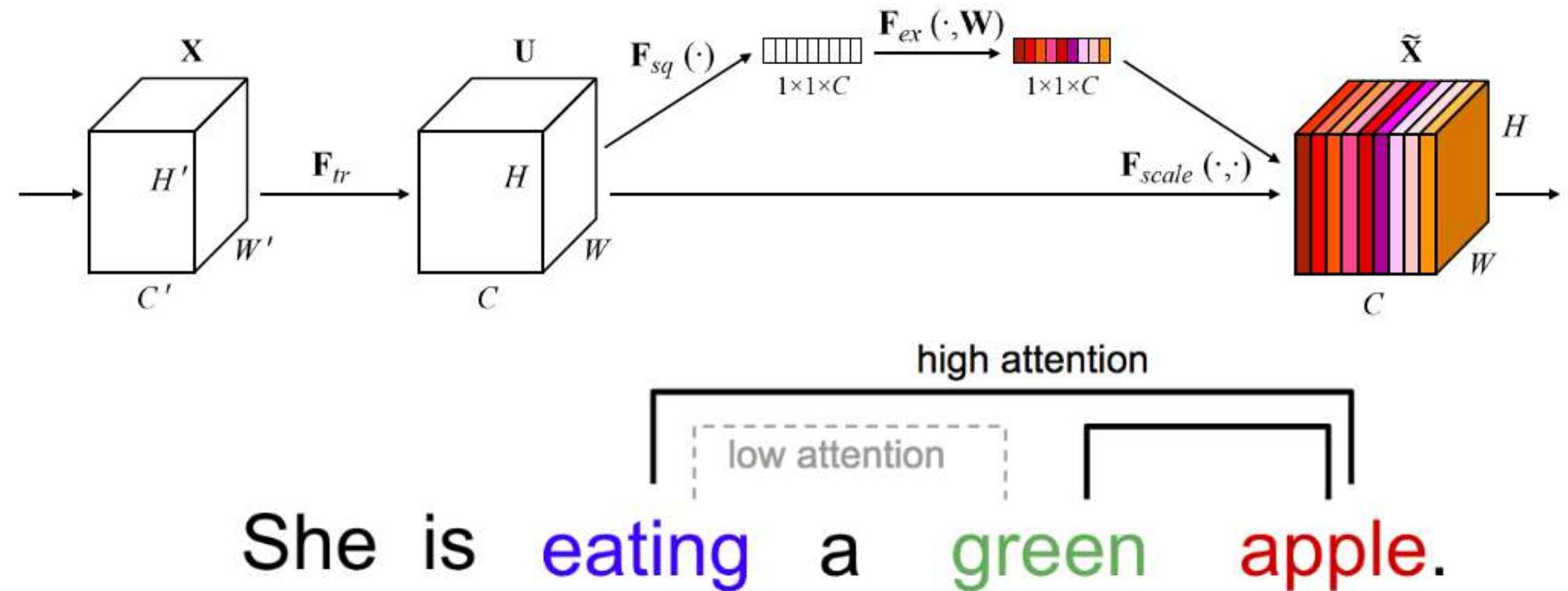


# Related Theory

Self Attention

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features

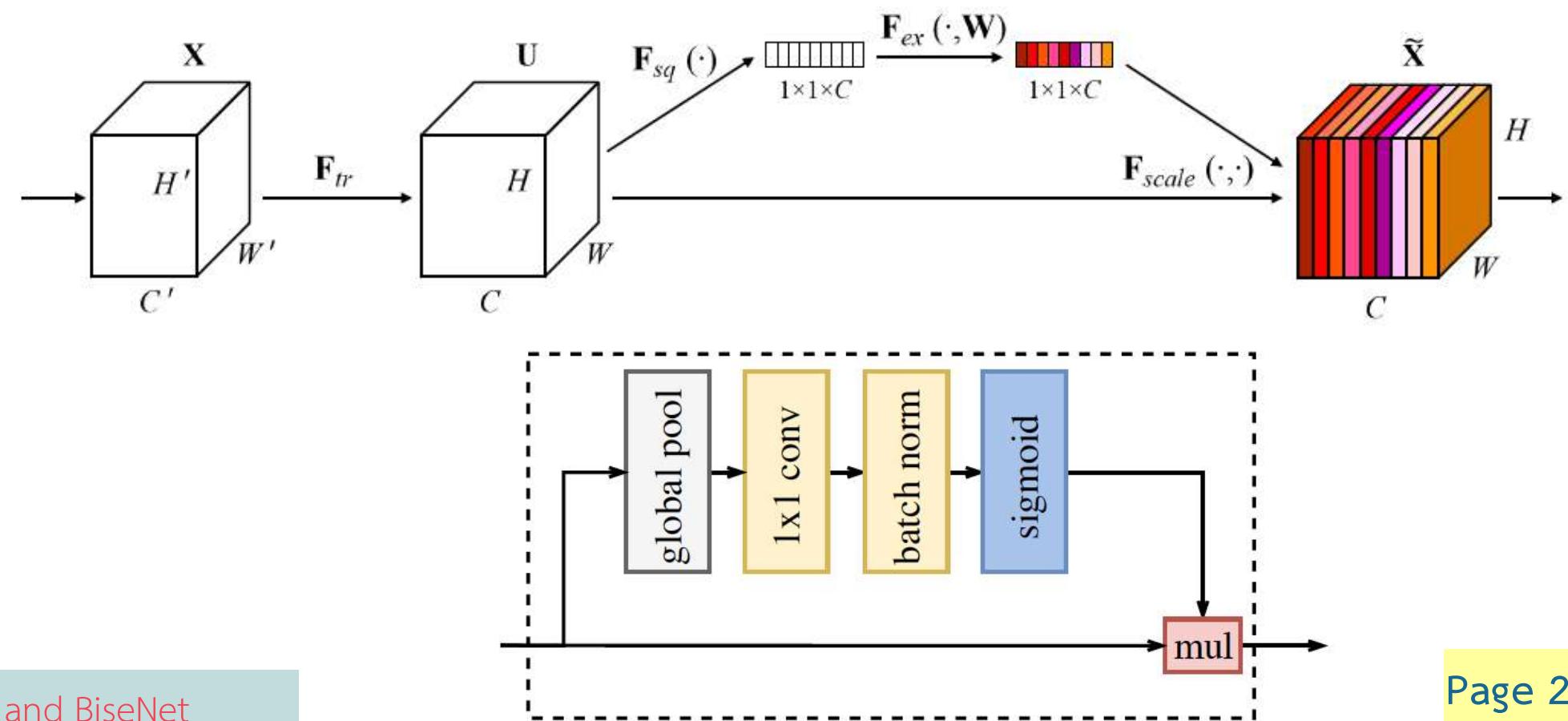


# Related Theory

Self Attention

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features



Refers to Squeeze-and-Excitation Networks and BiSeNet

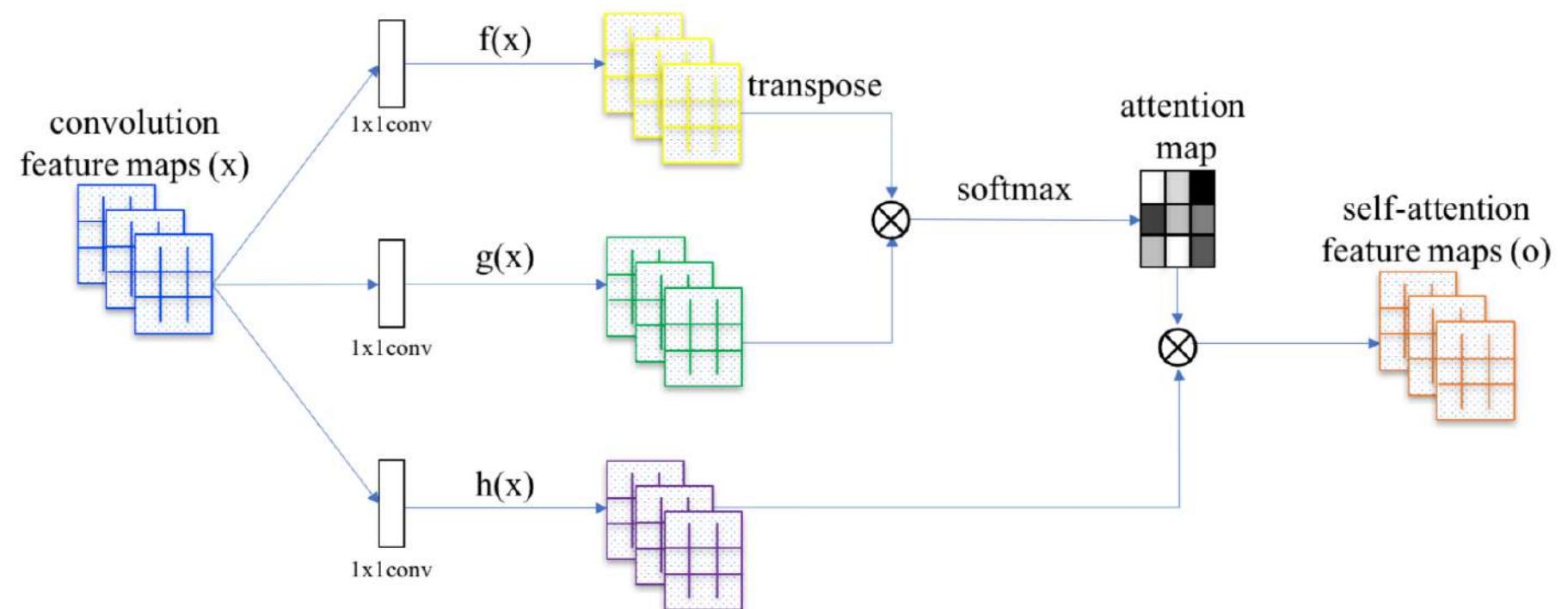
Page 27

# Related Theory

## Self Attention

- (1) Transfer Learning
- **(2) Channel Attention**
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features

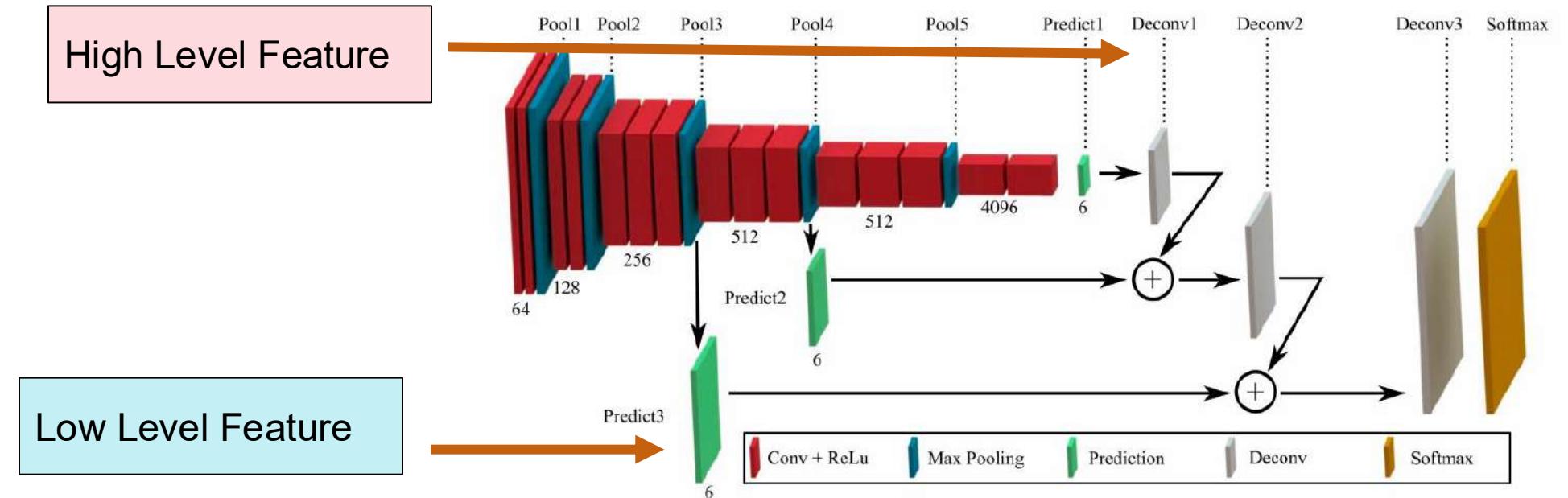


# Related Theory

## Feature Fusion (1)

- (1) Transfer Learning
- (2) Channel Attention
- **(3) Feature Fusion**
- (4) Depthwise Convolution
- (5) Design CNNs

- The features of the two paths are different in level of feature representation
- Simply **sum up** low and high features
  - Utilization of low-level features for objects refinement

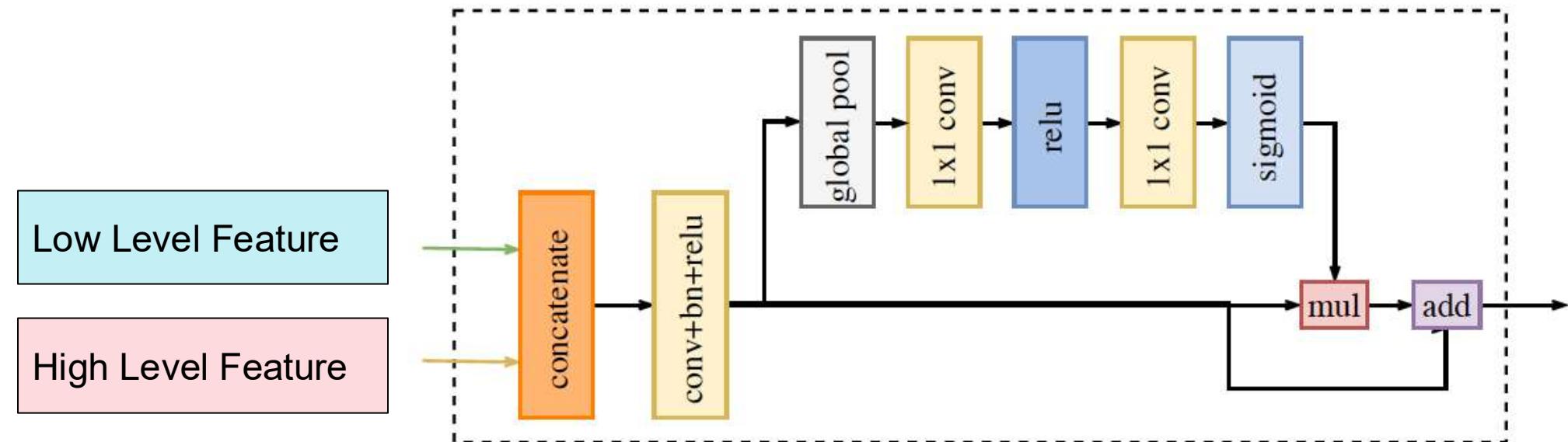


# Related Theory

## Feature Fusion (2)

- (1) Transfer Learning
- (2) Channel Attention
- **(3) Feature Fusion**
- (4) Depthwise Convolution
- (5) Design CNNs

- The features of the two paths are different in level of feature representation
- Fuse spatial path (low level features) and context path (high level feature) together

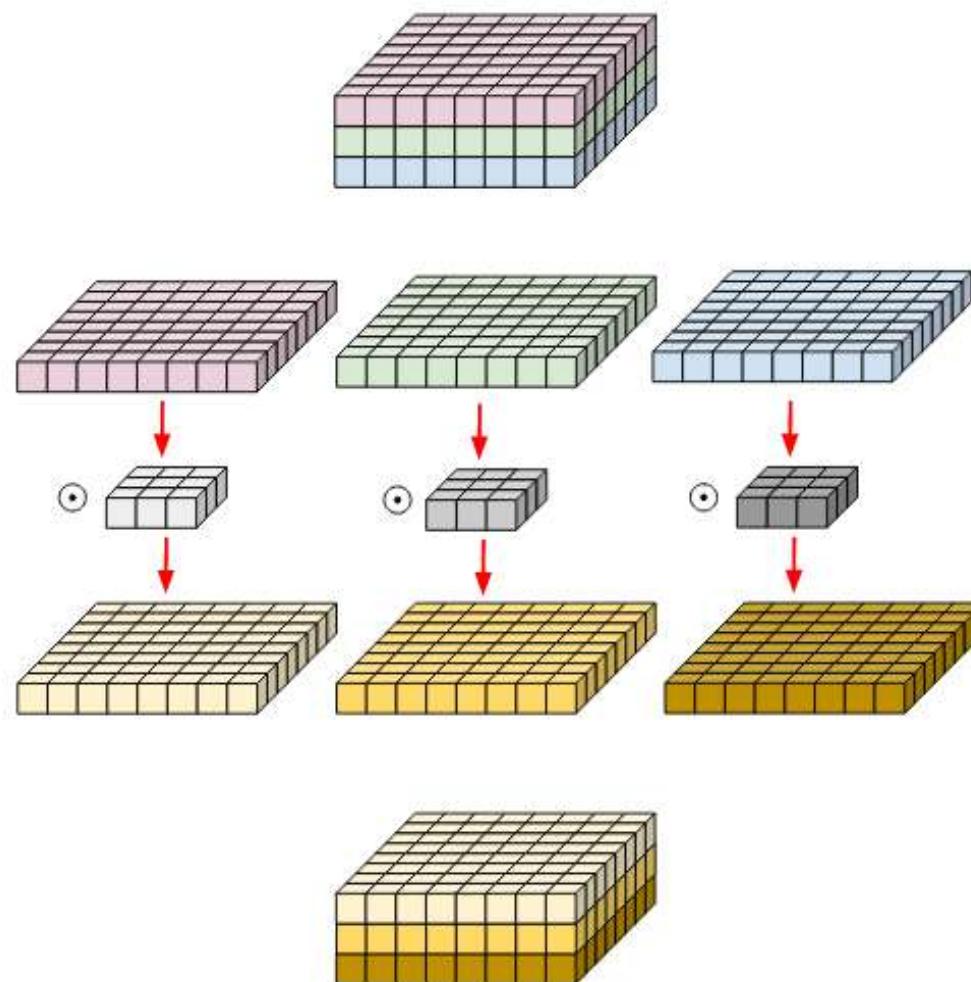


(c) Feature Fusion Module

# Related Theory

## Depth-wise Convolution

- (1) Transfer Learning
  - (2) Channel Attention
  - (3) Feature Fusion
  - **(4) Depthwise Convolution**
  - (5) Design CNNs
- Filters and image **have been broken into three different channels** and then convolved separately and stacked thereafter



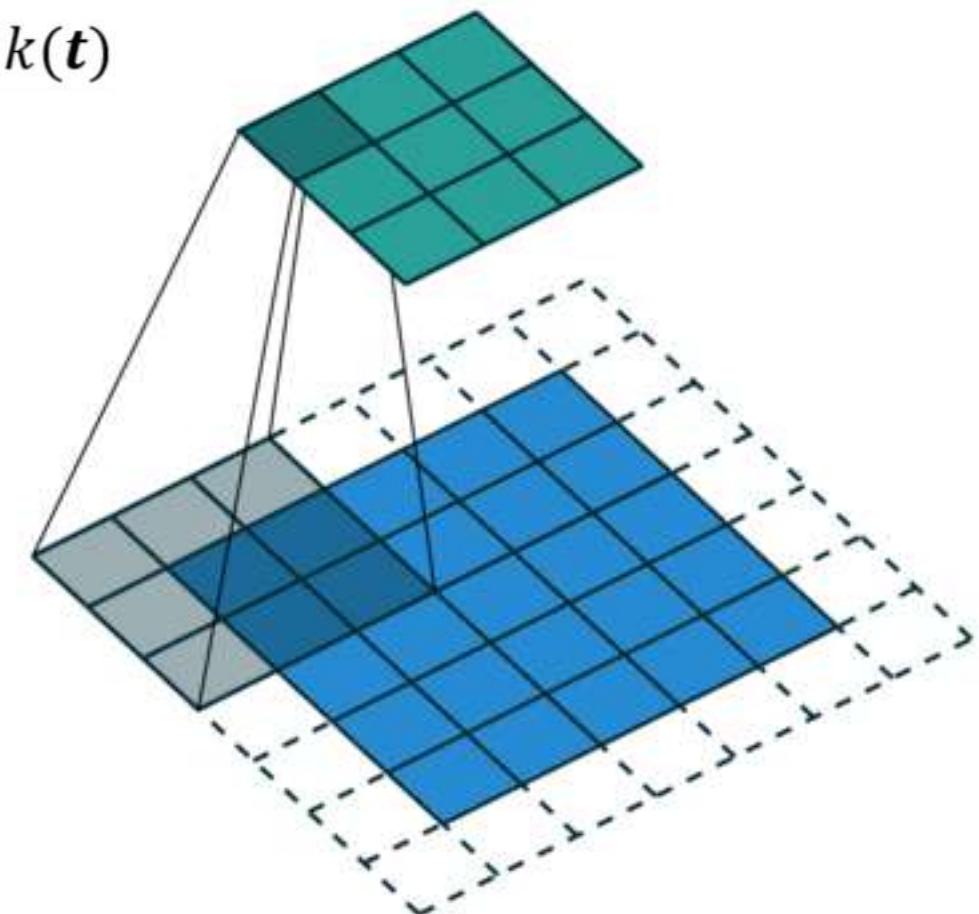
# Related Theory

## Point-wise Convolution

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- **(4) Depthwise Convolution**
- (5) Design CNNs

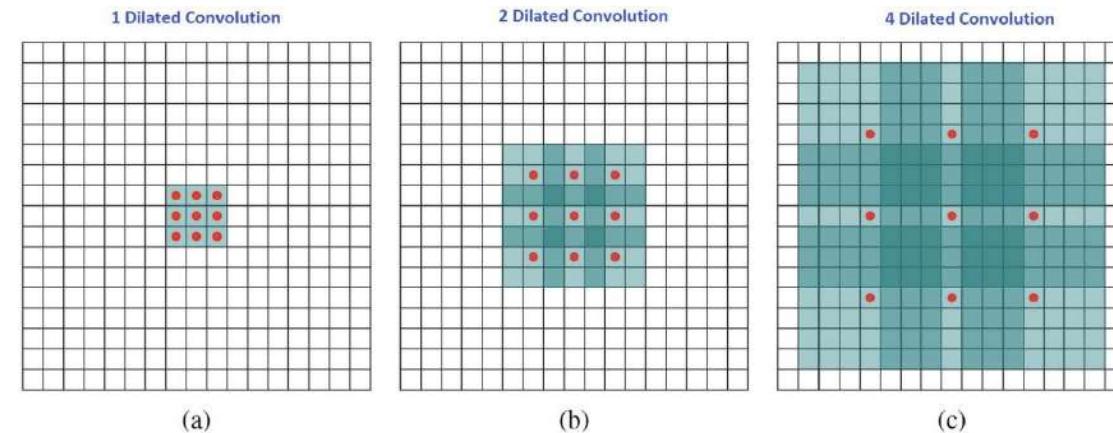
This is the standard discrete convolution:

$$(F * k)(\mathbf{p}) = \sum_{s+t=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$



# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- **(4) Depthwise Convolution**
- (5) Design CNNs



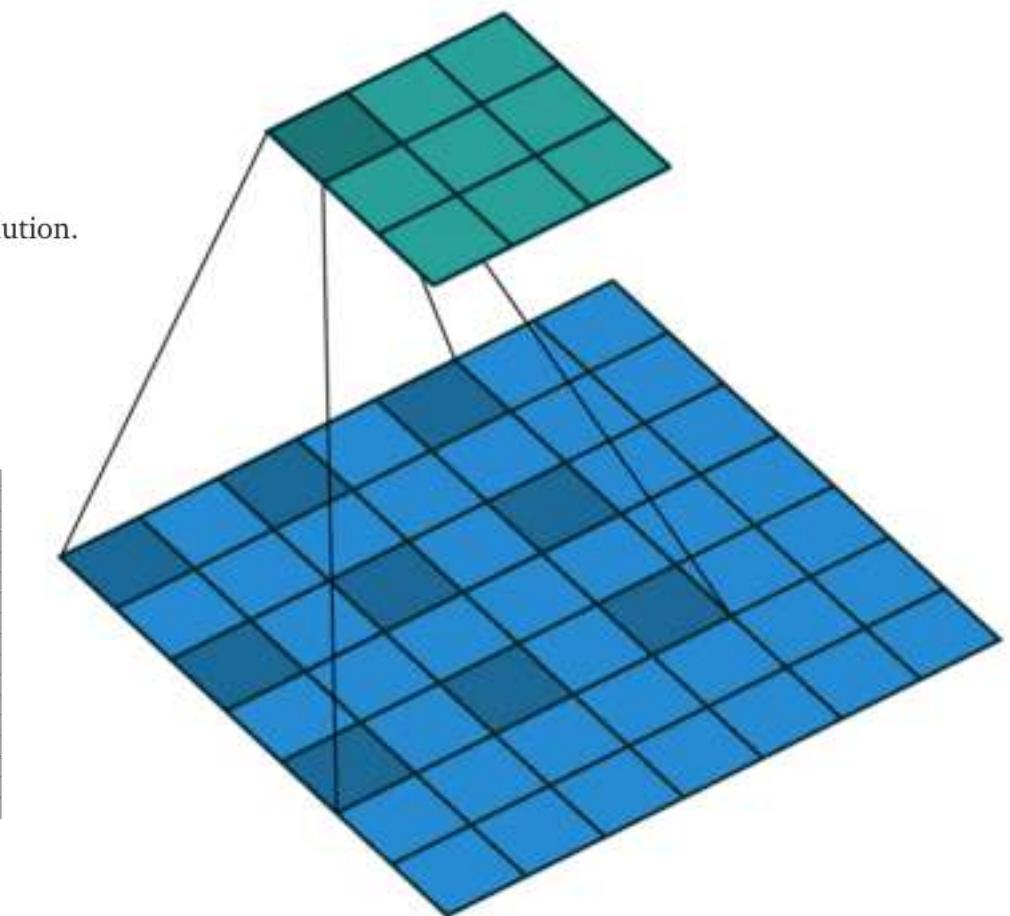
## Dilated Convolution (Atrous Convolution)

- Multi-scale context aggregation by dilated convolutions

The dilated convolution follows:

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s} + l\mathbf{t} = \mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$

When  $l = 1$ , the dilated convolution becomes as the standard convolution.

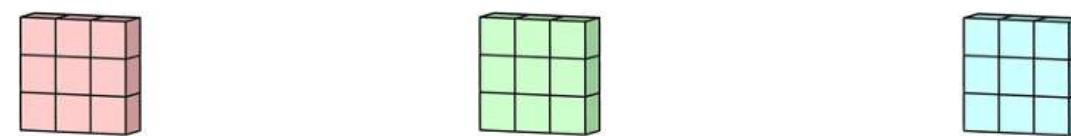
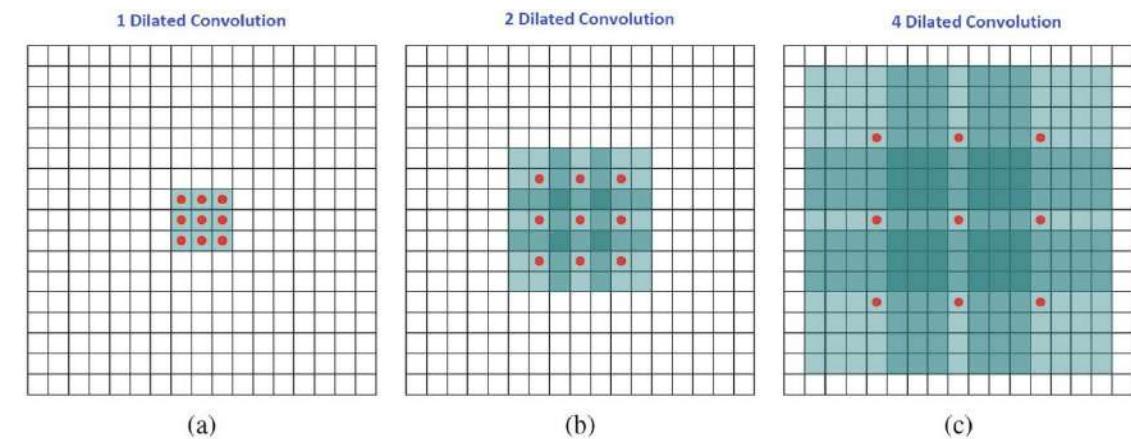


# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- **(4) Depthwise Convolution**
- (5) Design CNNs

## Dilated Convolution (Atrous Convolution)

- Multi-scale context aggregation by dilated convolutions

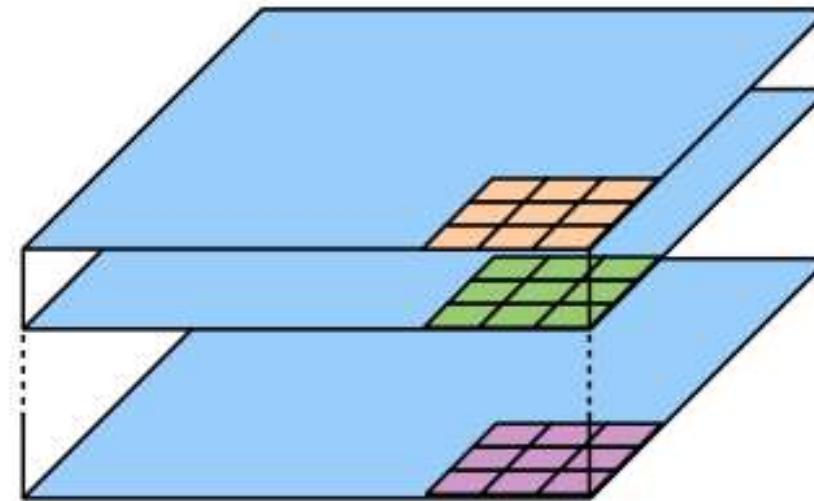


# Related Theory

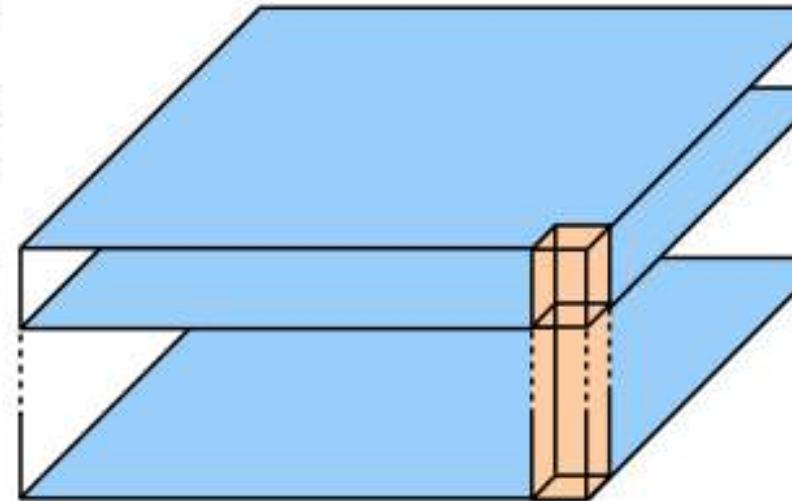
## Dilated Convolution (Atrous Convolution)

- **(4) Depthwise Convolution**

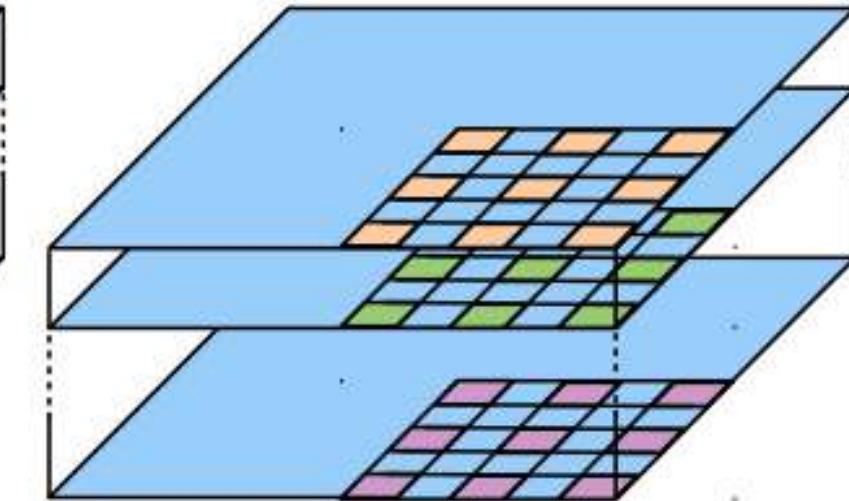
- **Multi-scale context aggregation** by dilated convolutions
- 3x3 Depthwise separable convolution decomposes a standard convolution into
  - (a) a depthwise convolution (applying a single filter for each input channel)
  - (b) a pointwise convolution (combining the outputs from depthwise convolution across channels).
- In this example, we explore atrous separable convolution where atrous convolution is adopted in the depthwise convolution, as shown in (c) with rate = 2.



(a) Depthwise conv.



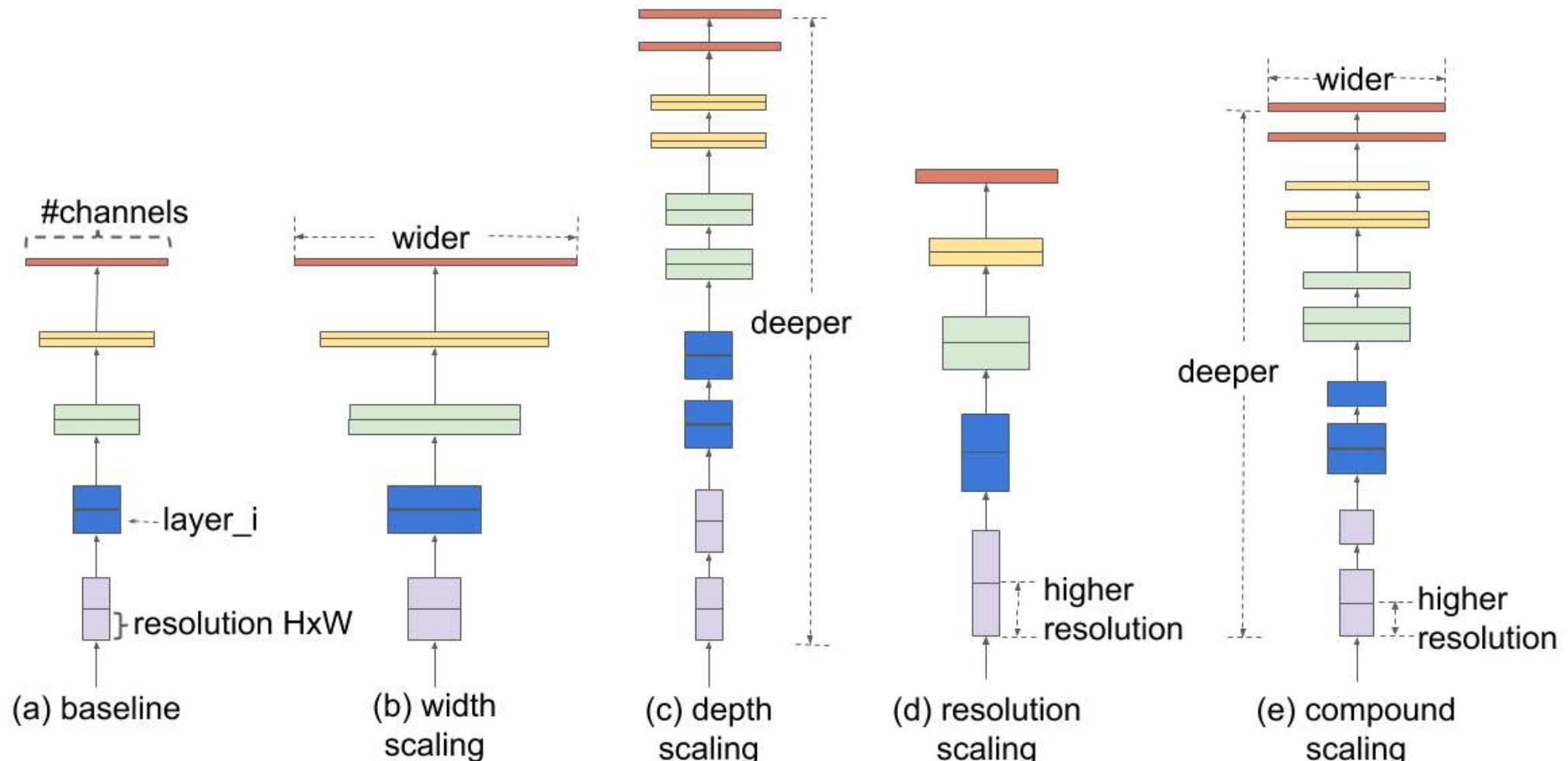
(b) Pointwise conv.



(c) Atrous depthwise conv.

# Related Theory

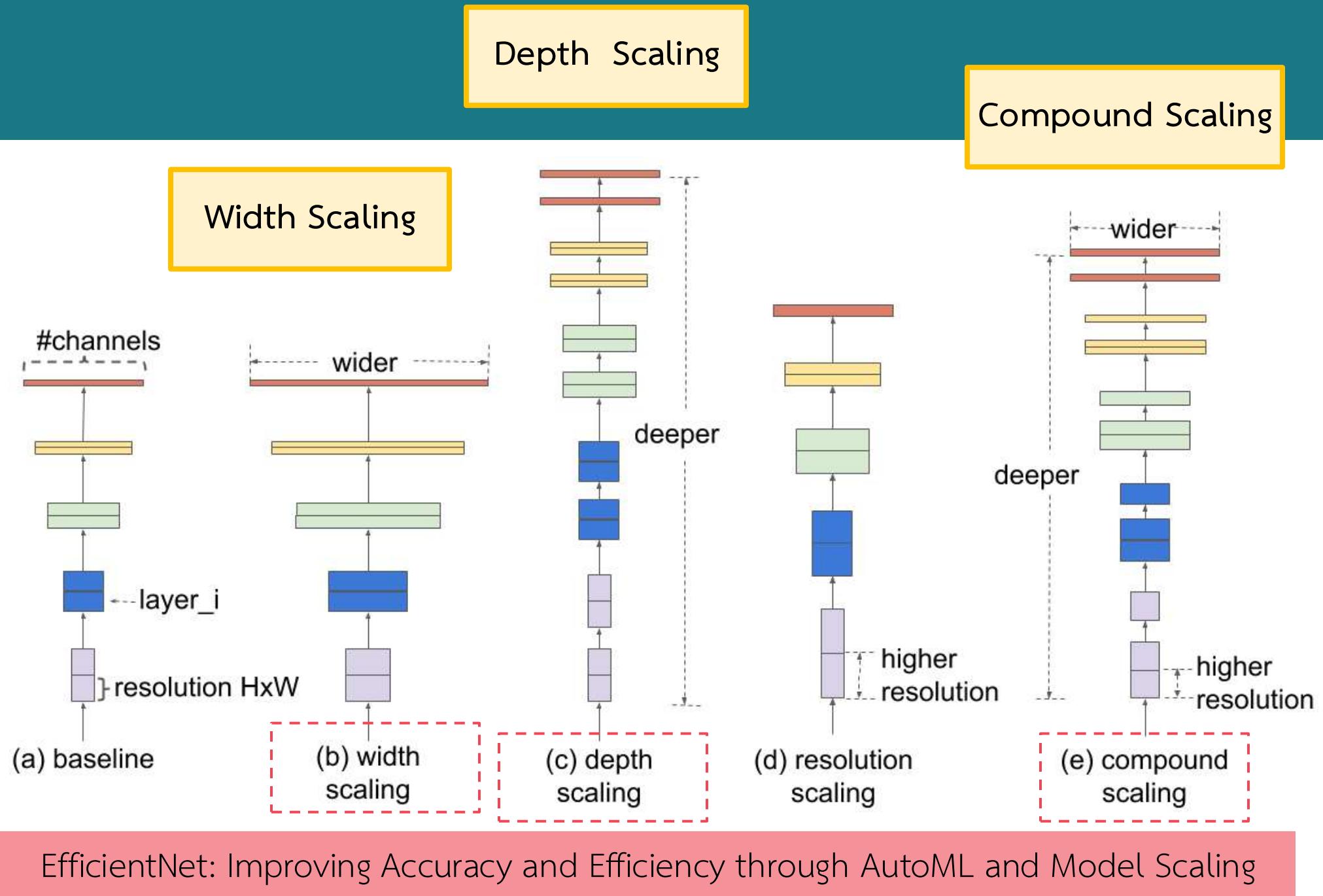
- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs



EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling

# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

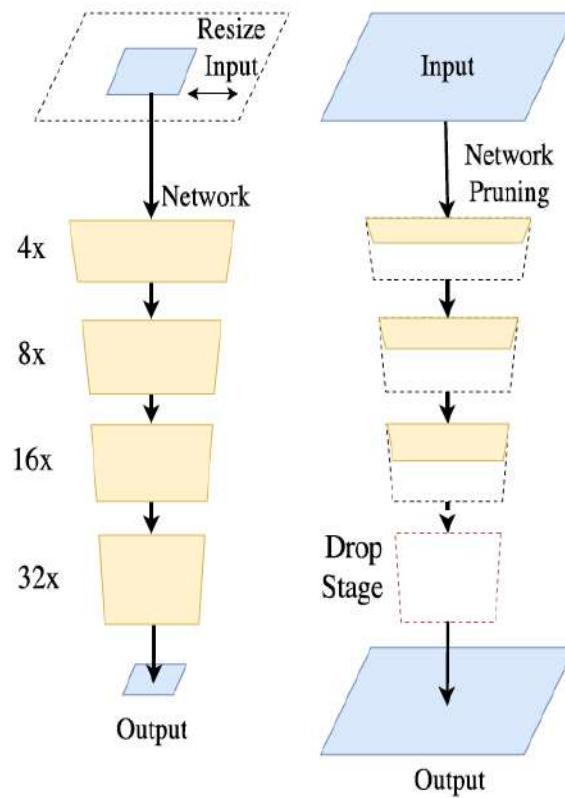


# Related Theory

- (1) Transfer Learning
- (2) Channel Attention
- (3) Feature Fusion
- (4) Depthwise Convolution
- (5) Design CNNs

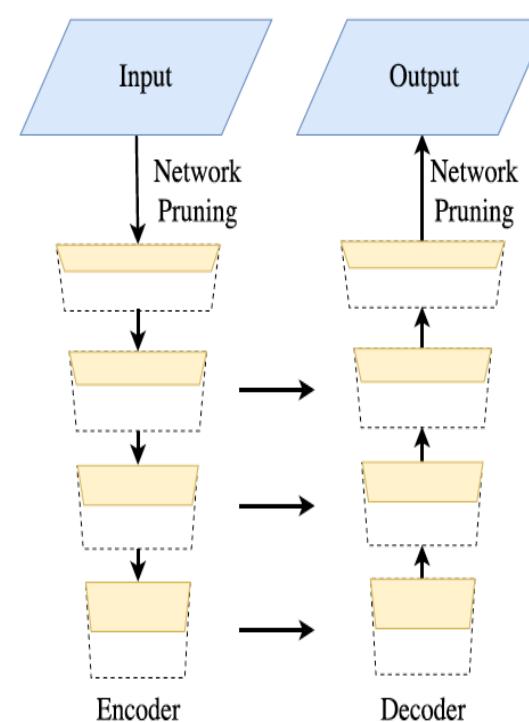
Dept Scaling

VGG Style  
(Depth Scaling)



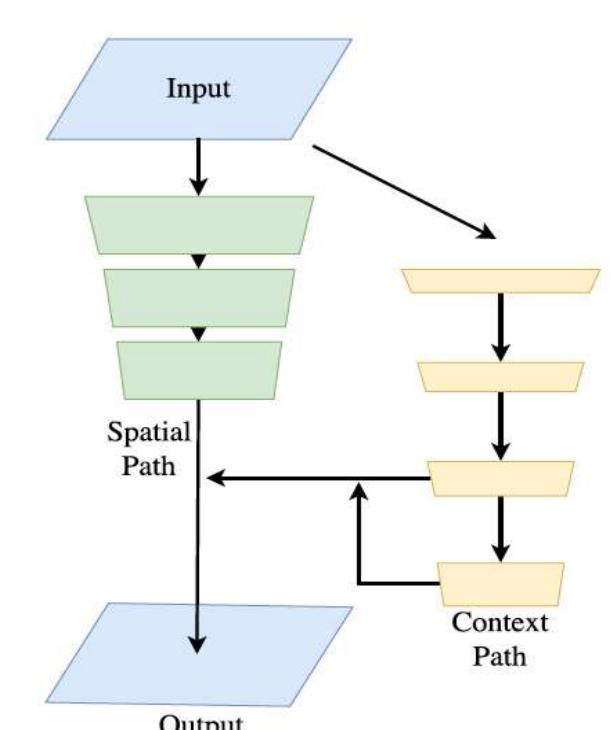
Width Scaling

U-Shape Style  
(Width Scaling)



Compound Scaling

Context Path Style  
(Compound Scaling)



# Outline | Related Works

- Introduction
- Related Theory
- **Related Works**
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - CamVid Corpus (<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>)

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - CamVid Corpus (<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>)



# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing, 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing, 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing, 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing, 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing, 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
    - F1-Score on Test Set is 80.8%
  - Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
    - F1-Score on Test Set is 84.7%
  - Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
    - F1-Score on Test Set is 83.5%
  - Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
    - F1-Score on Test Set is 85.2%
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing 2018)
    - F1-Score on Test Set is 85.4%

# Related Works

- **(1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)**

- Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
  - F1-Score on Test Set is 80.8%
- Segnet: A Deep Convolutional Encoder-Decoder Architecture by Badrinarayanan, V. et al. (PAMI, 2017)
  - F1-Score on Test Set is 84.7%
- Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
  - F1-Score on Test Set is 83.5%
- Gated Convolutional Neural Network by Wang, H. et al. (Remote Sensing 2017)
  - F1-Score on Test Set is 85.2%
- **Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing 2018)**
  - F1-Score on Test Set is **85.4%**



Winner is Encoder-Decoder (ScasNet-based)

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - CamVid Corpus (<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>)

Method	Imp surf	Building	Low veg	Tree	Car	F1-score
FCN-8s {Long, 2015 #6}	0.871	0.918	0.752	0.861	0.638	0.808
SegNet {Badrinarayanan, 2017 #7}	0.867	0.891	0.763	0.839	0.657	0.847
DeconvNet {Noh, 2015 #8}	0.891	0.932	0.814	0.857	0.684	0.835
GSN {Wang, 2017 #9}	0.892	0.945	0.749	0.875	0.798	0.852
Encoder-Decoder {Liu, 2018 #10}	0.872	0.893	0.841	0.914	0.815	0.854

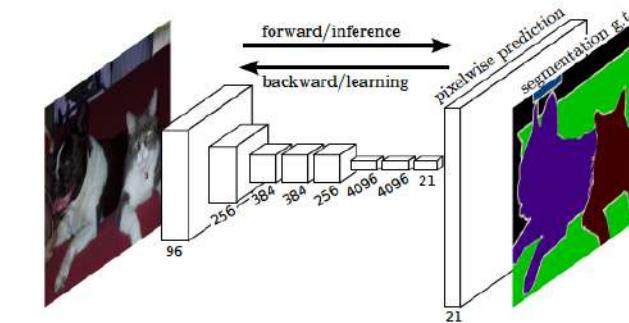
Winner

# Related Works

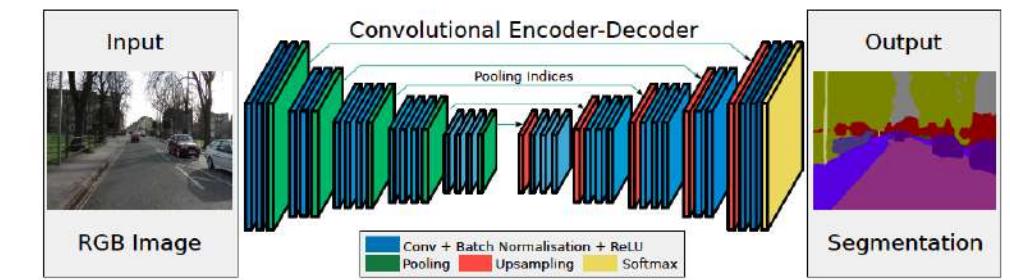
Point of view in the previous work

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)

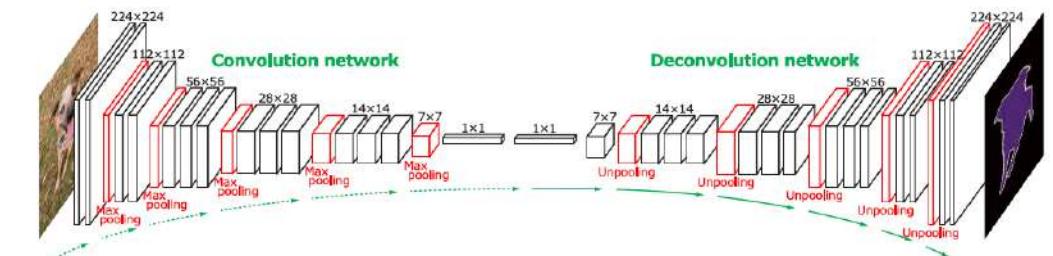
- Fully Convolutional Networks by Long, J. et. al. (CVPR 2015)
  - F1-Score on Test Set is 80.8%



- Segnet: A Deep Convolutional Encoder-Decoder Architecture
- by Badrinarayanan, V. et al. (PAMI, 2017)
  - F1-Score on Test Set is 75.5%



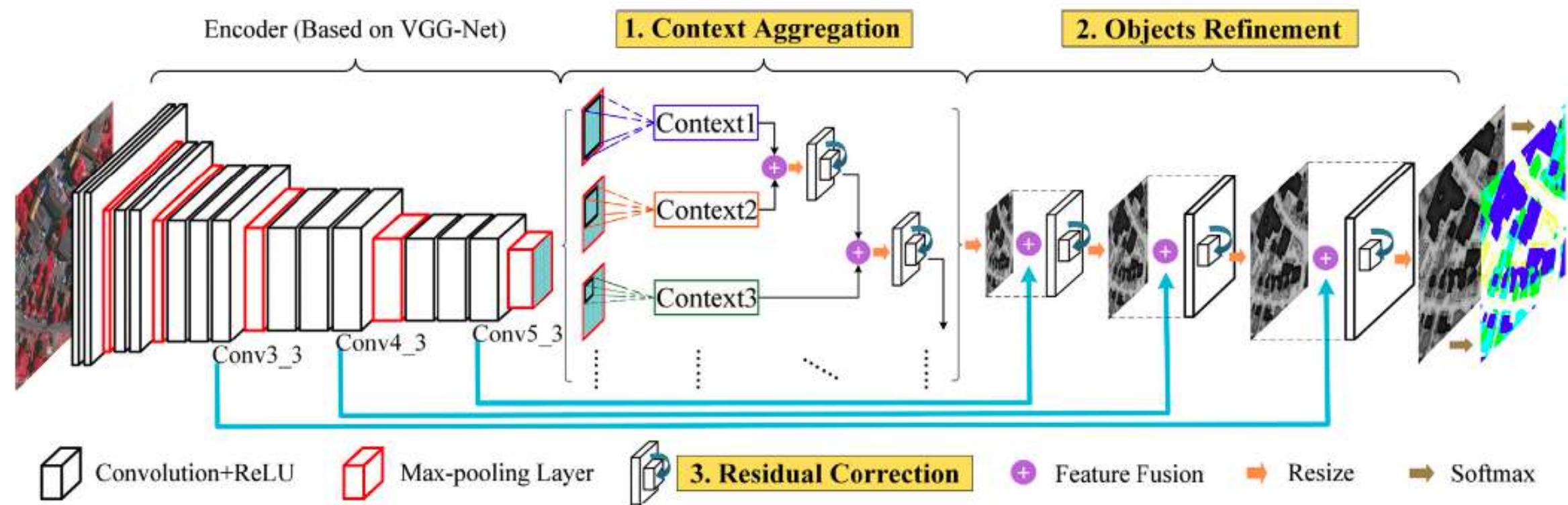
- Learning Deconvolution Network by Noh, H. et al. (CVPR 2015)
  - F1-Score on Test Set is 83.5%



# Related Works

Point of view in the previous work

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
  - Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing)
  - F1-Score on Test Set is **85.4% (Winner)**



# Related Works

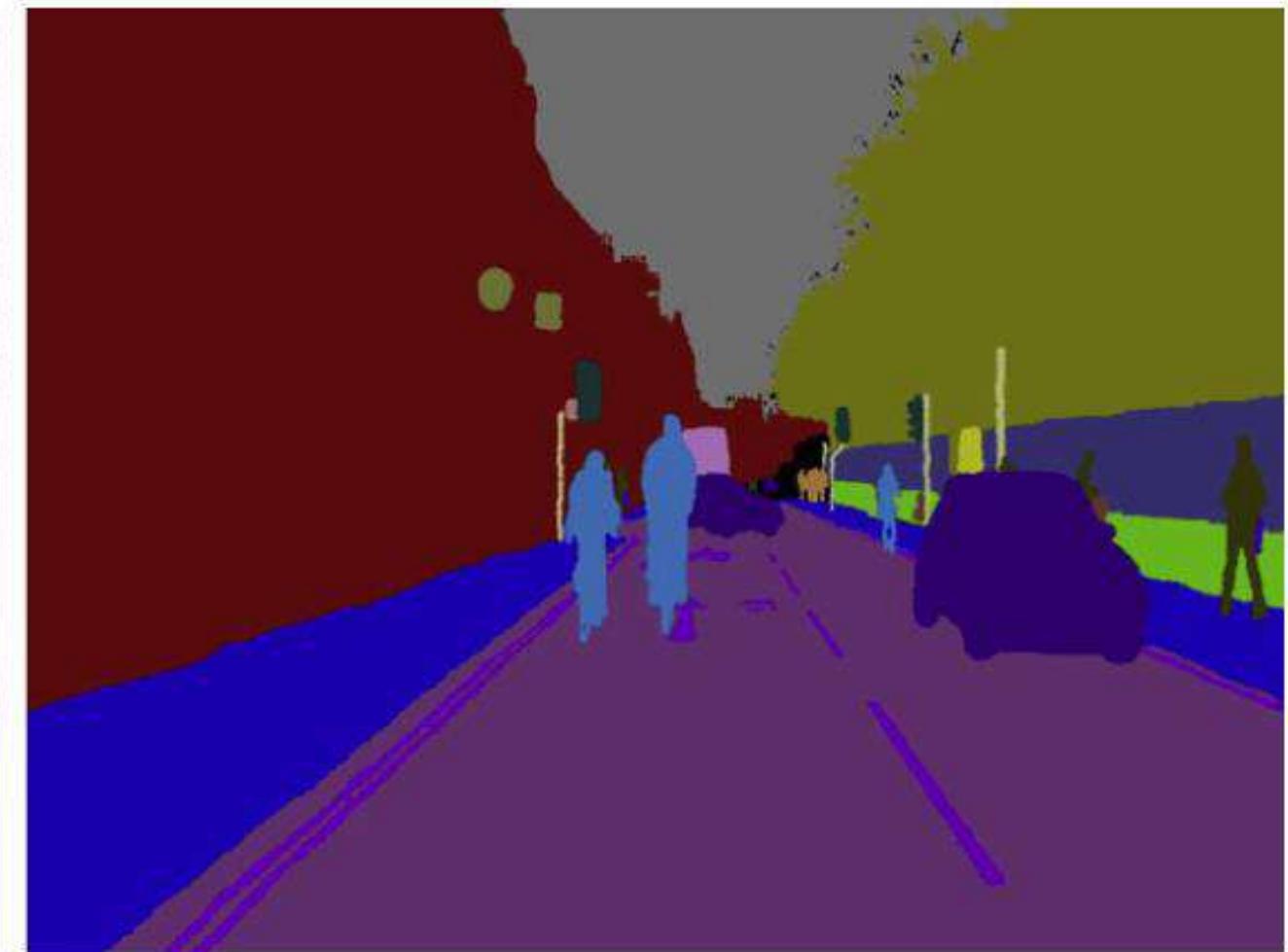
- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - CamVid Corpus (<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>)

# CamVid Corpus

The Cambridge-driving Labeled Video Database

Void	Building	Wall	Tree	VegetationMisc
Fence	Sidewalk	ParkingBlock	Column_Pole	TrafficCone
Bridge	SignSymbol	Misc_Text	TrafficLight	Sky
Tunnel	Archway	Road	RoadShoulder	LaneMkgsDriv
LaneMkgsNonDriv	Animal	Pedestrian	Child	CartLuggagePram
Bicyclist	MotorcycleScooter	Car	SUVPickupTruck	Truck_Bus
Train	OtherMoving			

32 semantic classes



<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is **86.1%**
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is **67.2%**
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - **Bilateral Network (Bisenet)** by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is **83.1%**

# Related Works

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Pyramid Scene Parsing Network by Zhao, H. et al. (CVPR 2017)
    - F1-Score on Test Set is 80.8%
  - DenseNet (Tiramisu) by Jégou, S. et al. (CVPR 2017)
    - F1-Score on Test Set is 75.1%
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is 86.1%
  - Encoder-Decoder (DeepLabV3) by Chen, L. C. (ECCV 2018)
    - F1-Score on Test Set is 67.2%
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is 83.1%



Winner

Winner is Global Convolution Network (GCN)

# Related Works

- (1) Deep Learning on Remote Sensing Corpus (ISPRS Vaihingen Corpus)
- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - CamVid Corpus (<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>)

Deep Learning Model	Precision	Recall	F1-Score
PSPNet {Zhao, 2017 #1}	0.74	0.74	0.74
DenseNet (Tiramisu) {Badrinarayanan, 2017 #2}	0.74	0.77	0.75
GCN {Peng, 2018 #3}	0.85	0.87	0.86
DeepLabV3 {Chen, 2018 #4}	0.72	0.63	0.67
BiseNet {Yu, 2018 #5}	0.84	0.82	0.83

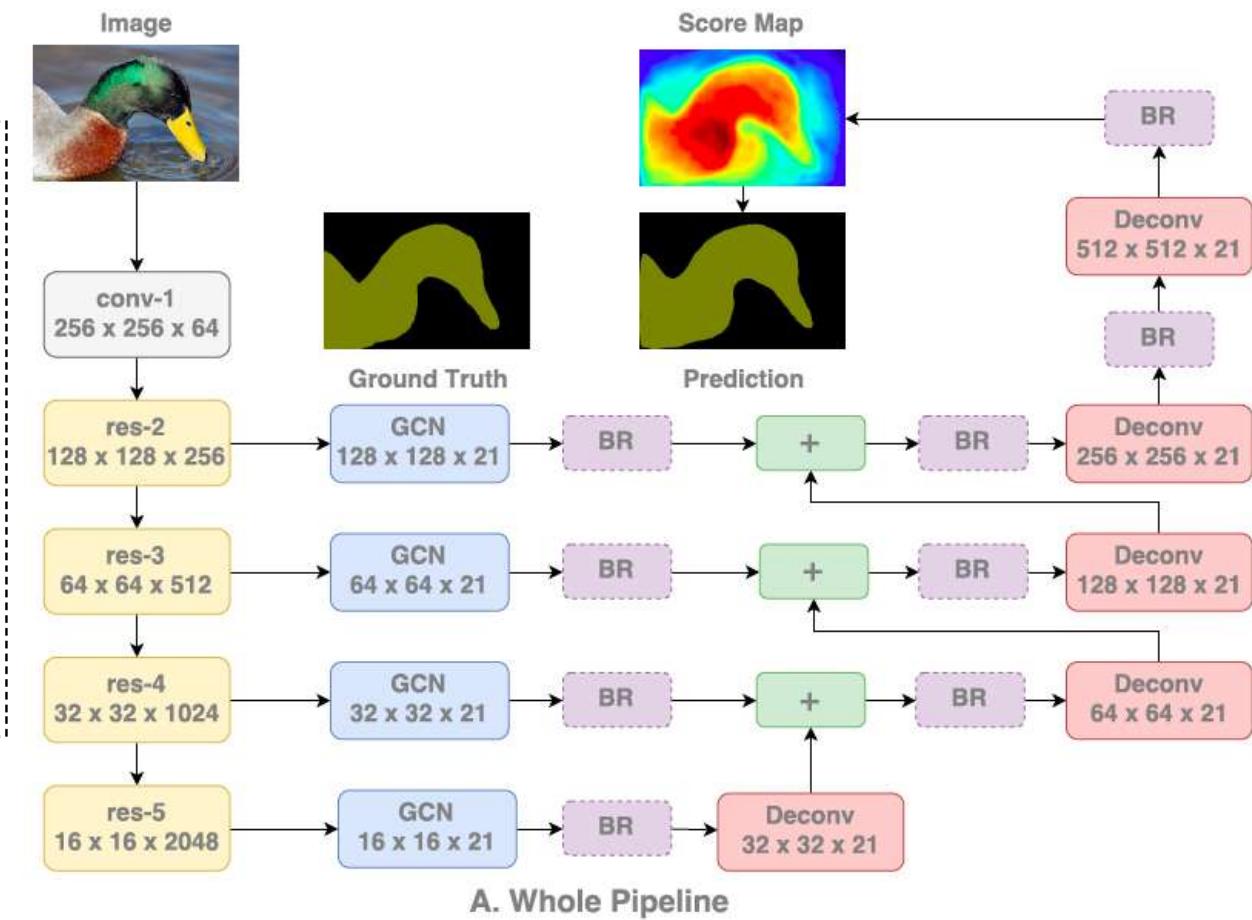
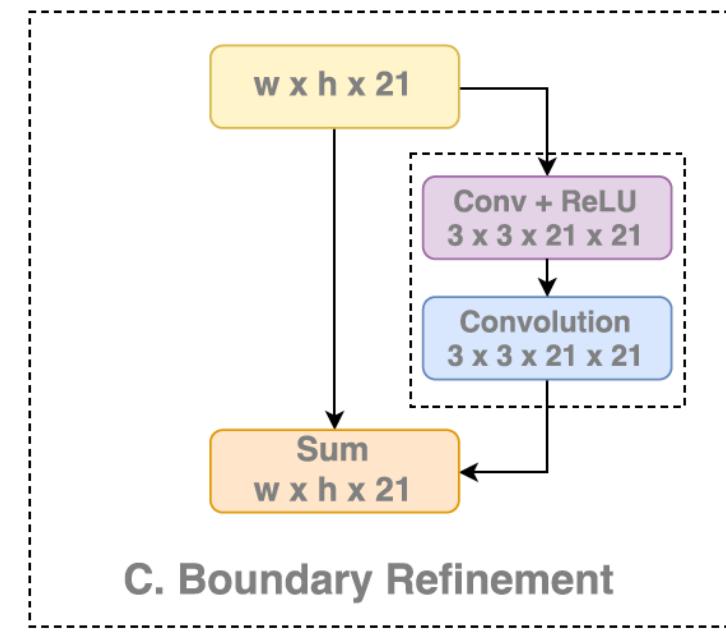
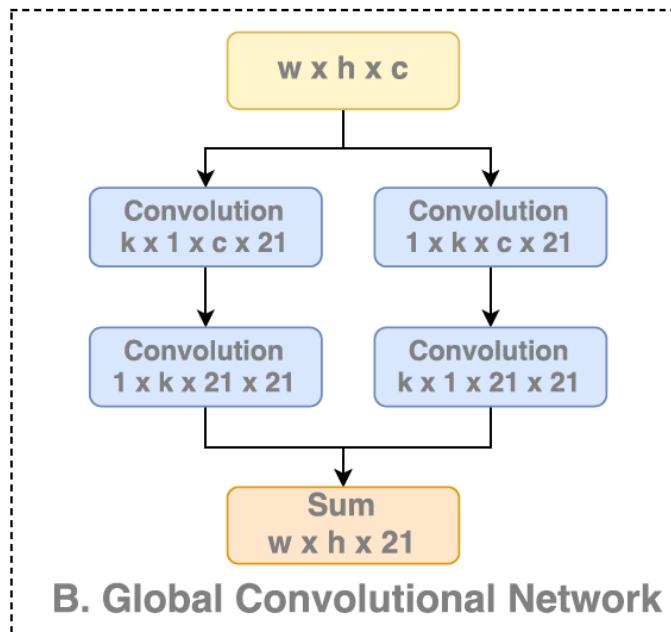


Winner

# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
  - F1-Score on Test Set is **86.1% (Winner)**



# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
    - F1-Score on Test Set is **86.1% (Winner)**

\*\* Valid Receptive Field (VRF)



A



B



C

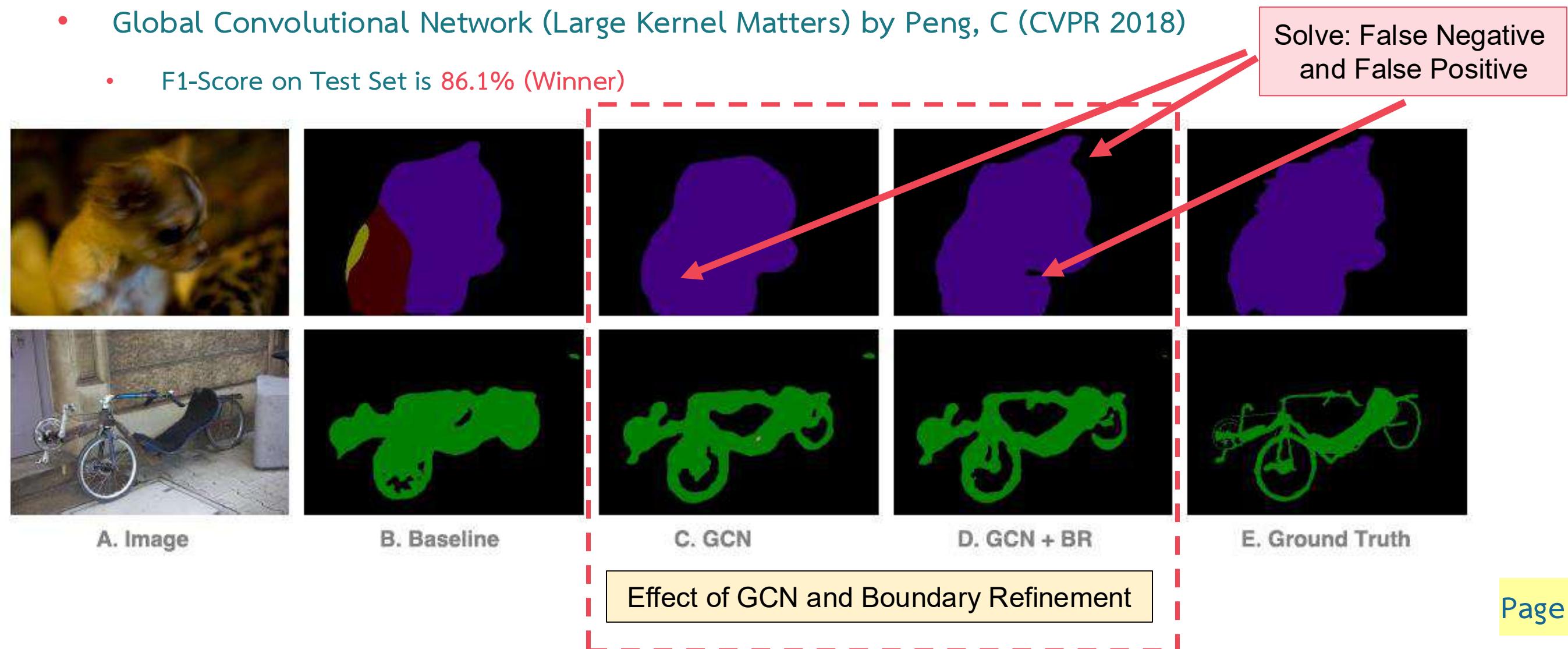
(A) and fails to hold the entire object if the input resized to a larger scale (B). As a comparison, their Global Convolution Network significantly enlarges the VRF (C).

# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)

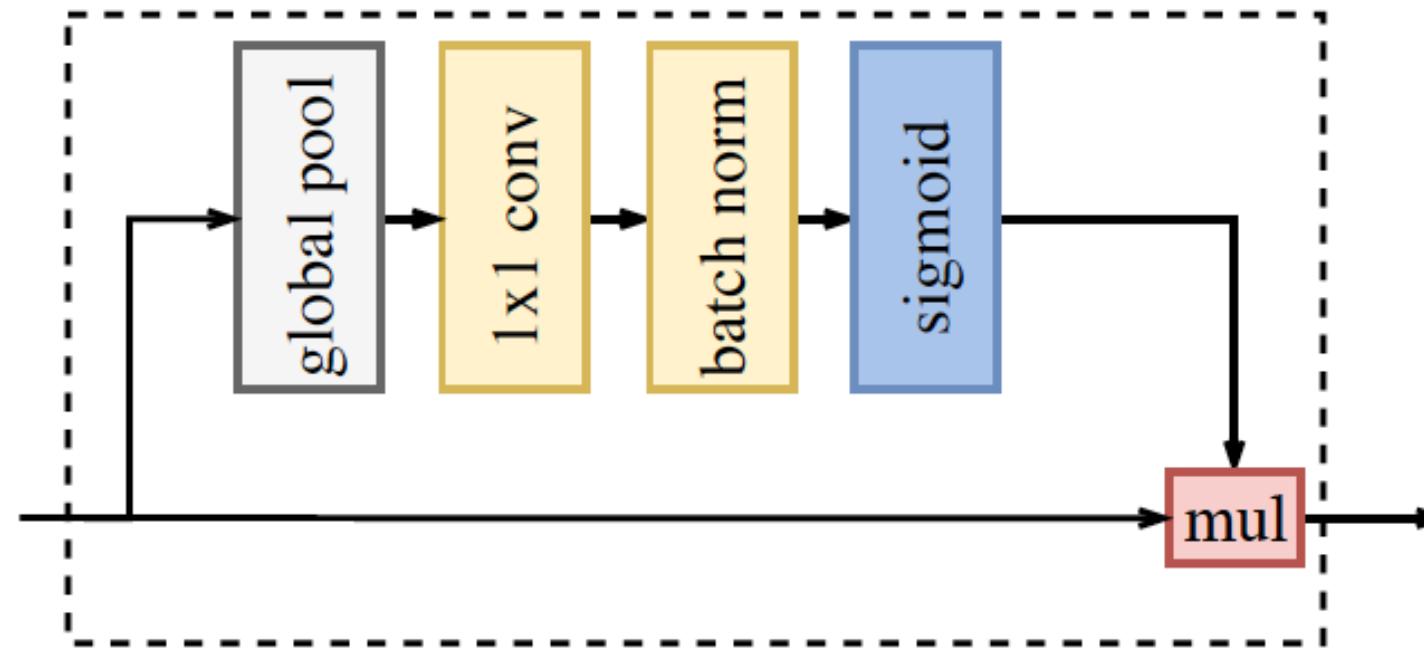
- Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018)
  - F1-Score on Test Set is **86.1% (Winner)**



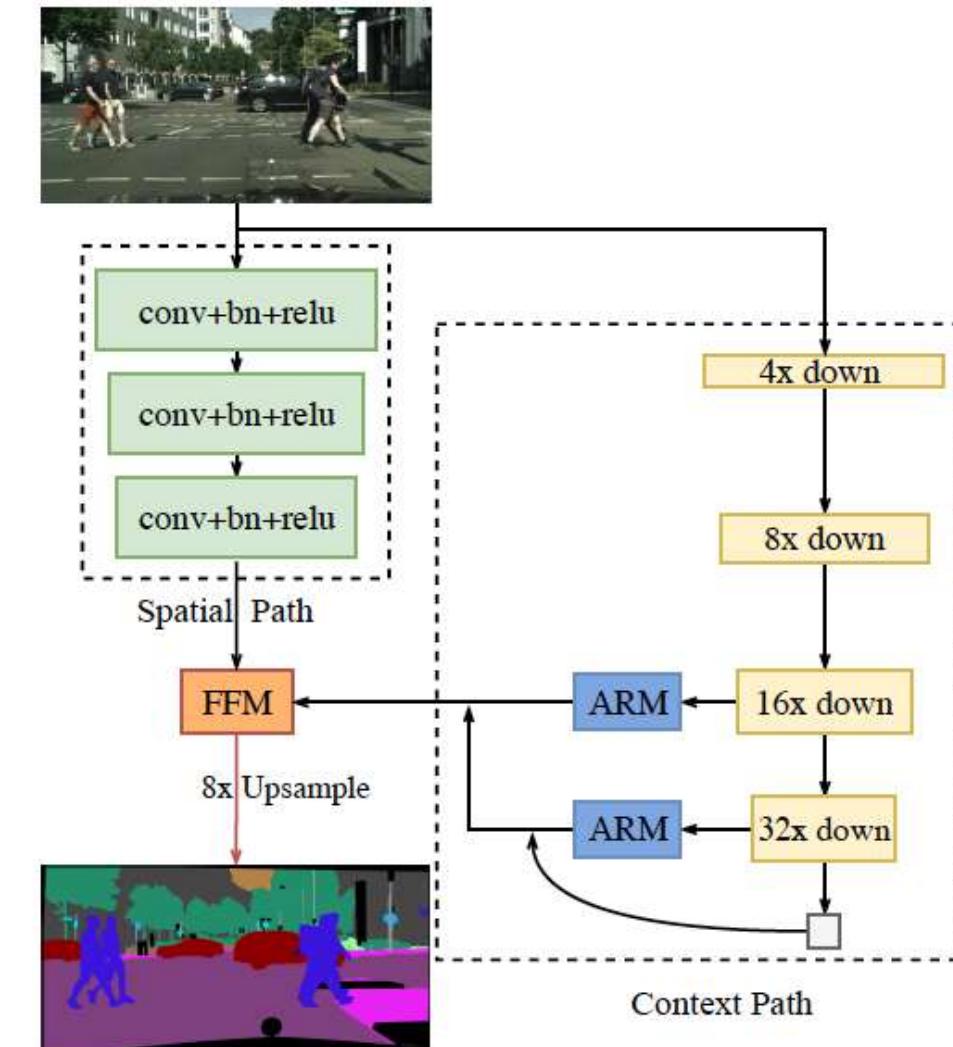
# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is **83.1%** (first runner-up)



(b) Attention Refinement Module

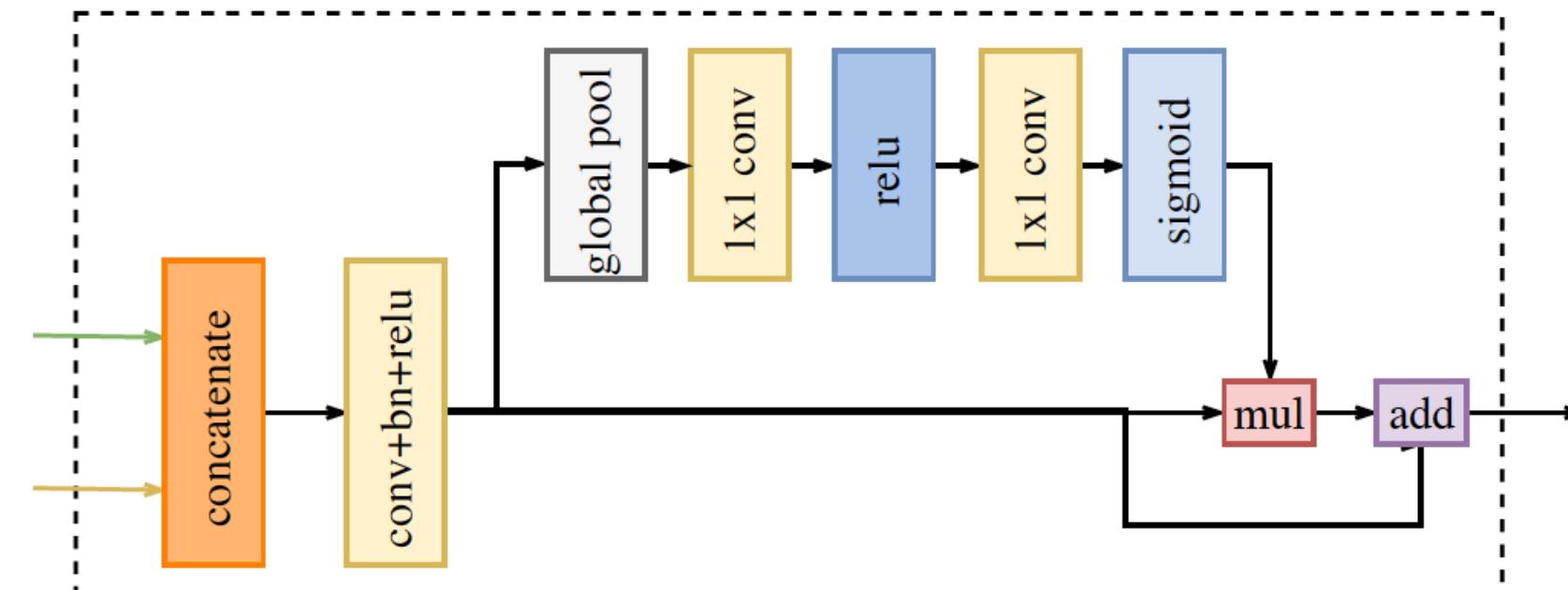


(a) Network Architecture

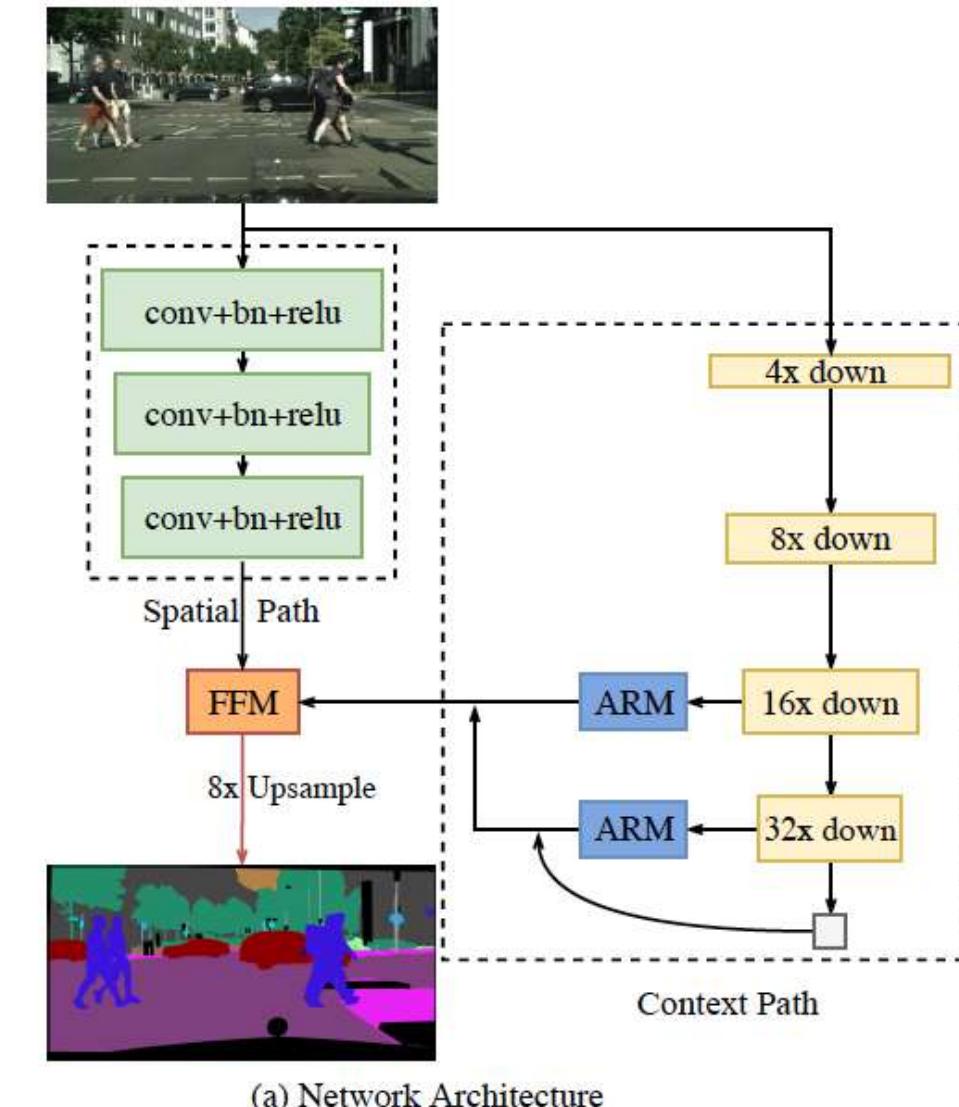
# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)
  - Bilateral Network (Bisenet) by Yu, C. (ECCV 2018)
    - F1-Score on Test Set is **83.1%** (first runner-up)



(c) Feature Fusion Module



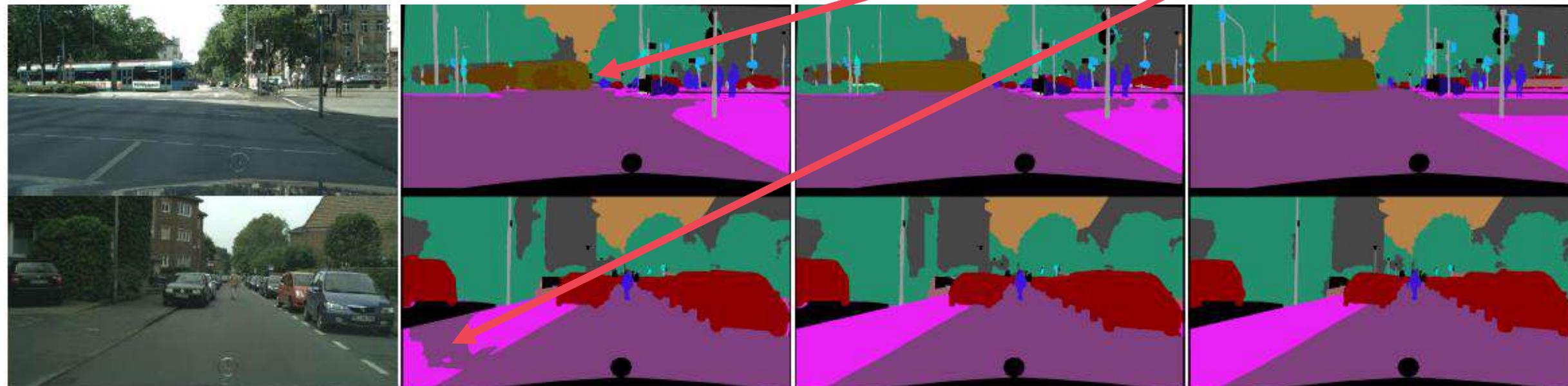
(a) Network Architecture

# Related Works

Point of view in the previous work

- (2) Modern Deep Learning on Challenge Corpora (based on CVPR, ECCV since 2017 to present)

- Bilateral Network (BiSeNet) by Yu, C. (ECCV 2018)
  - F1-Score on Test Set is 83.1% (first runner-up)



(a) Image

(b) U-Shape

(c) BiSeNet

(d) GT

# Recap: Each Techniques from Related Theory and Work

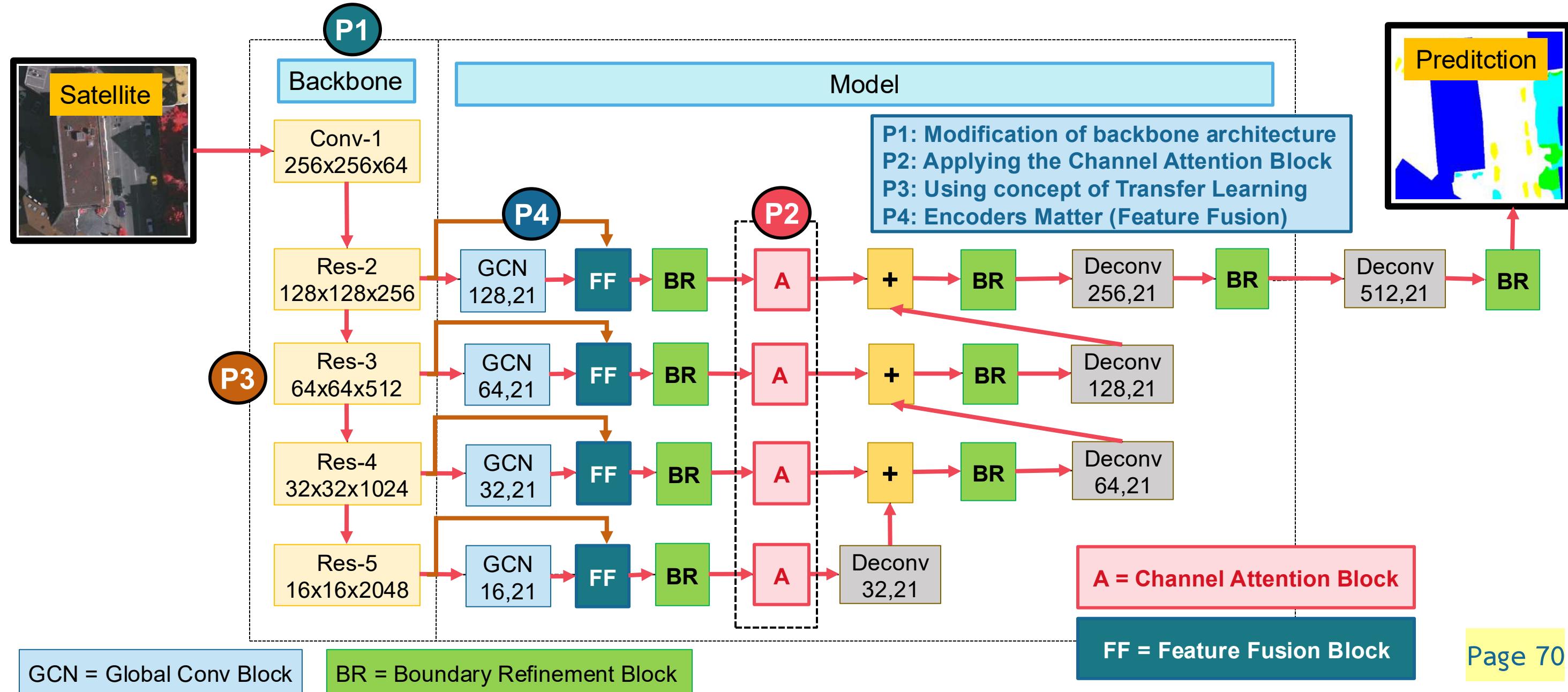
- From Remote Sensing Challenge, Encoder-Decoder ScasNet-based by Liu, Y. et al. (ISPRS Journal of Photogrammetry and Remote Sensing 2018) is the winner.
- From CamVid Challenge, Global Convolutional Network (Large Kernel Matters) by Peng, C (CVPR 2018) is the winner.
- Modern Technique from modern deep learning researches:
  - Global Convolutional (Large Kernel Matter, Dynamic Kernel Size)
  - Channel Attention
  - Domain Specific Transfer Learning
  - Feature Fusion
  - Depthwise Atrous Convolution

# Outline | Methodology (Proposed Method)

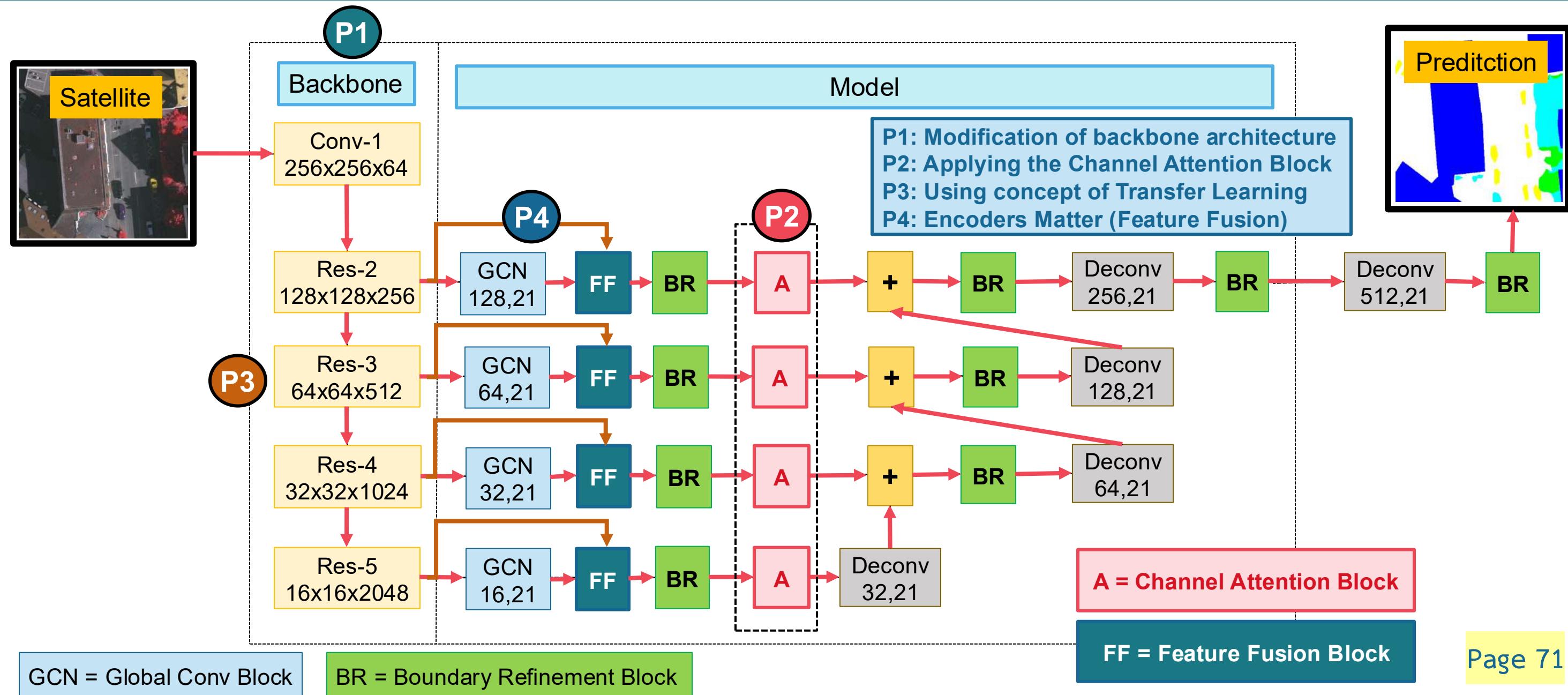
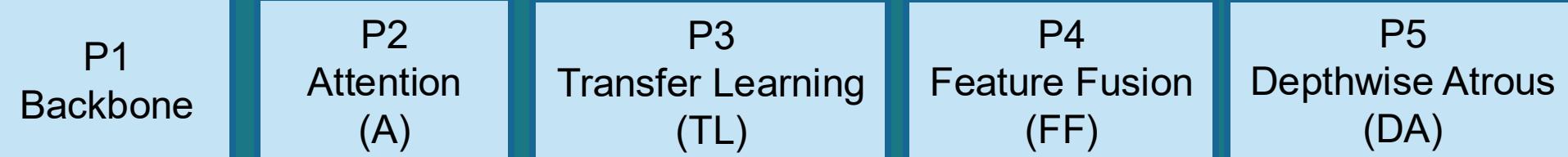
- Introduction
- Related Theory
- Related Works
- **Methodology (Proposed Method)**
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Proposed Method

Stage: Design Deep Learning Architecture



# Proposed Method



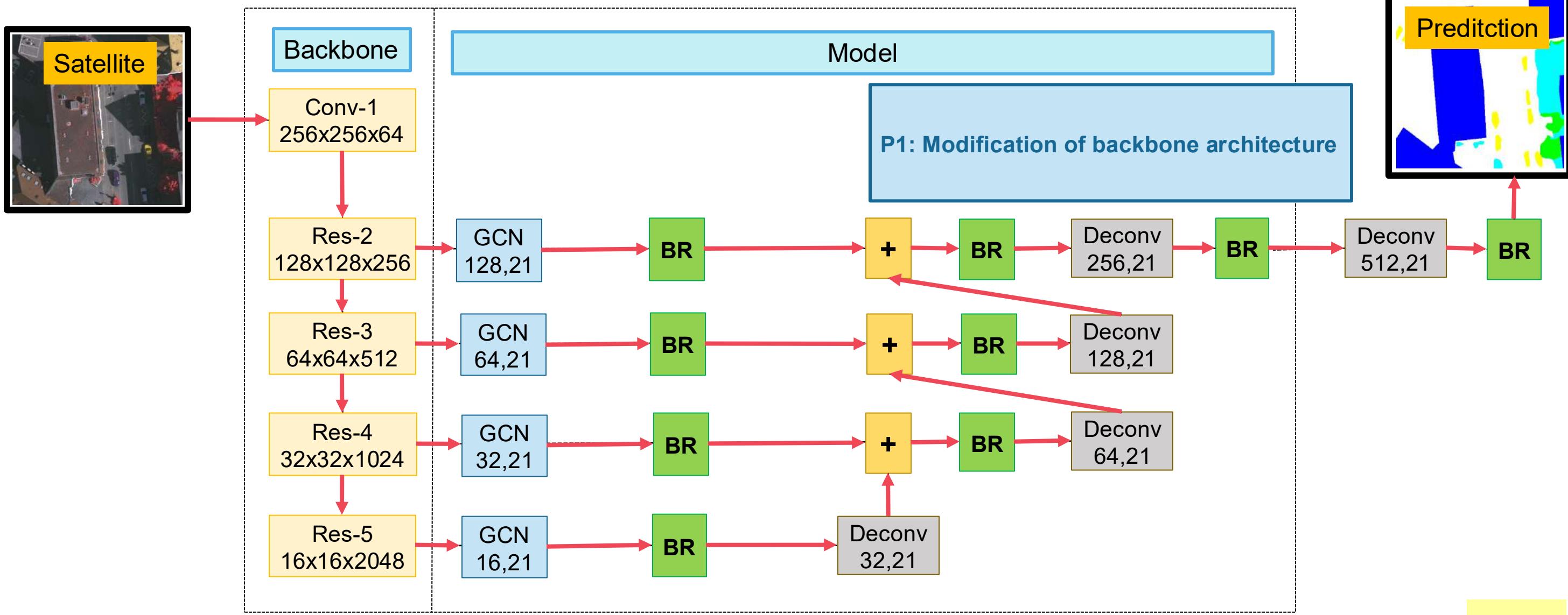
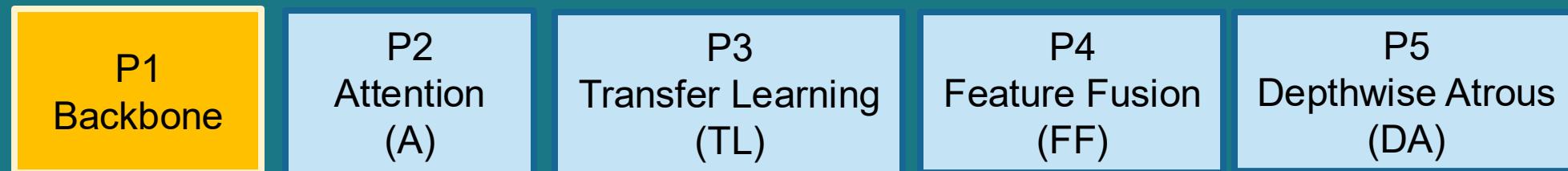
GCN = Global Conv Block

BR = Boundary Refinement Block

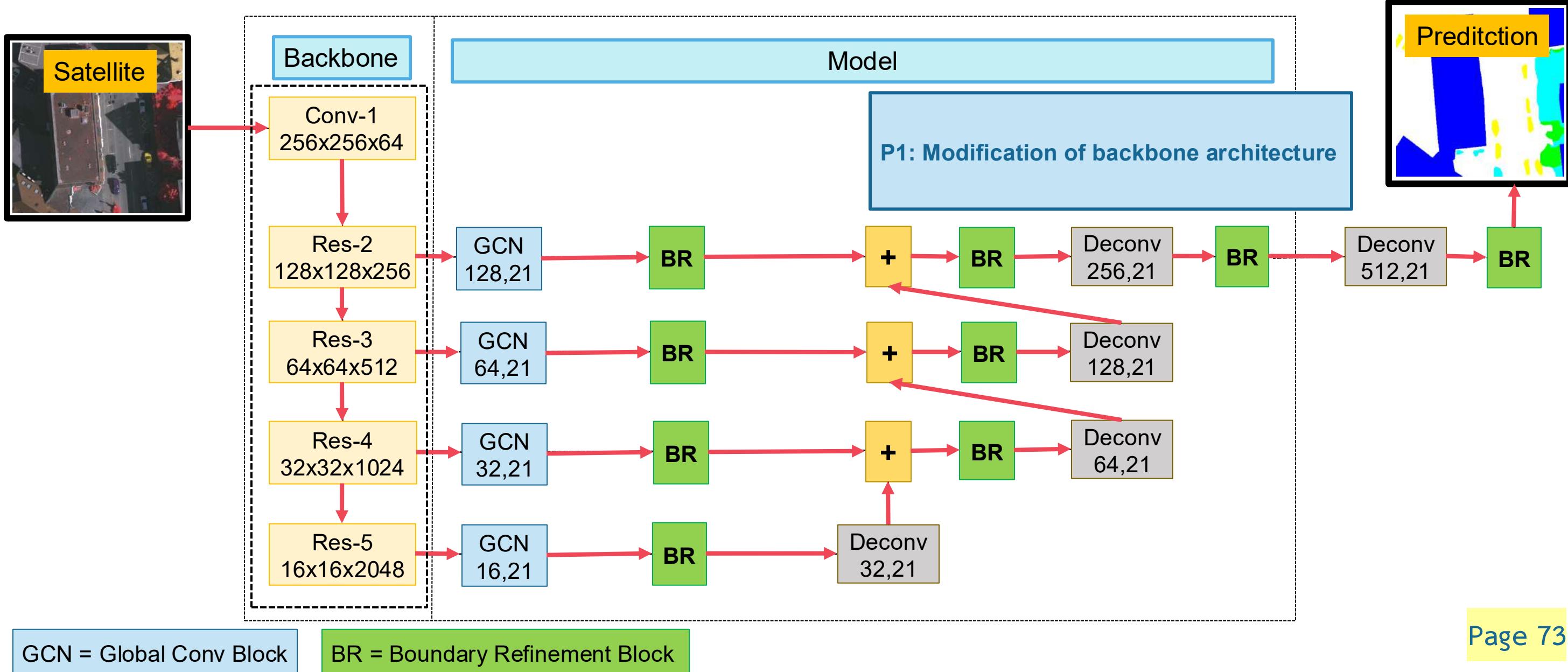
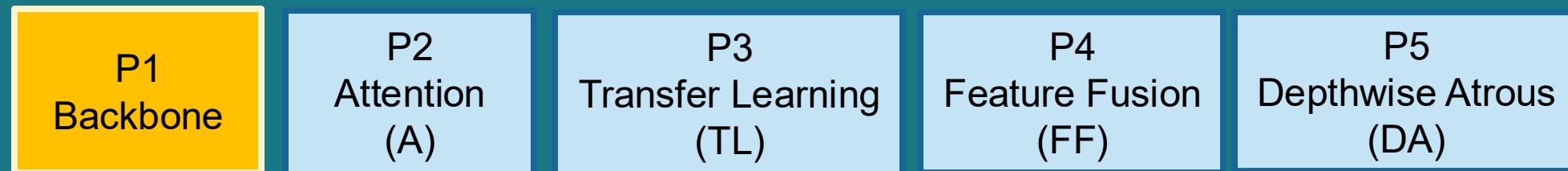
FF = Feature Fusion Block

Page 71

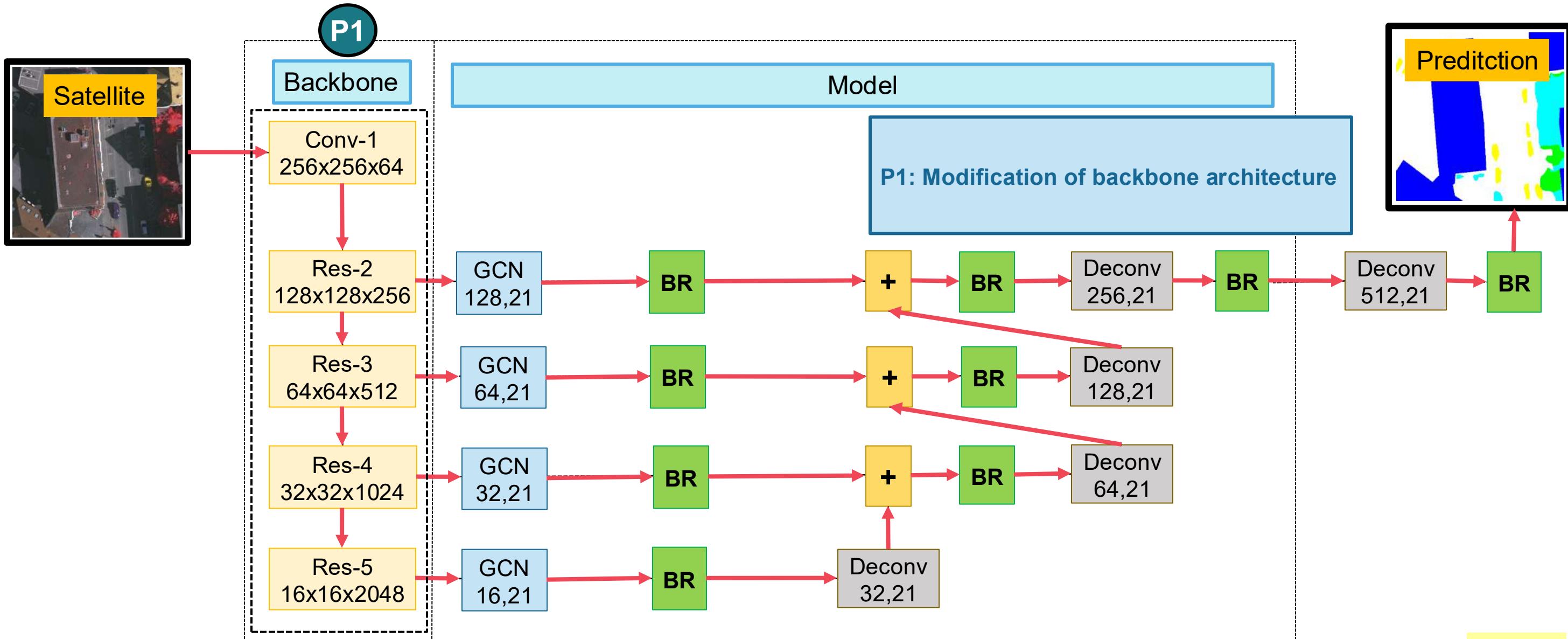
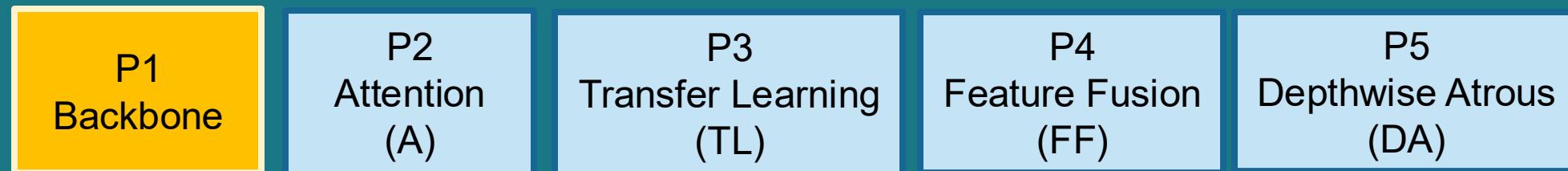
# Proposed Method



# Proposed Method



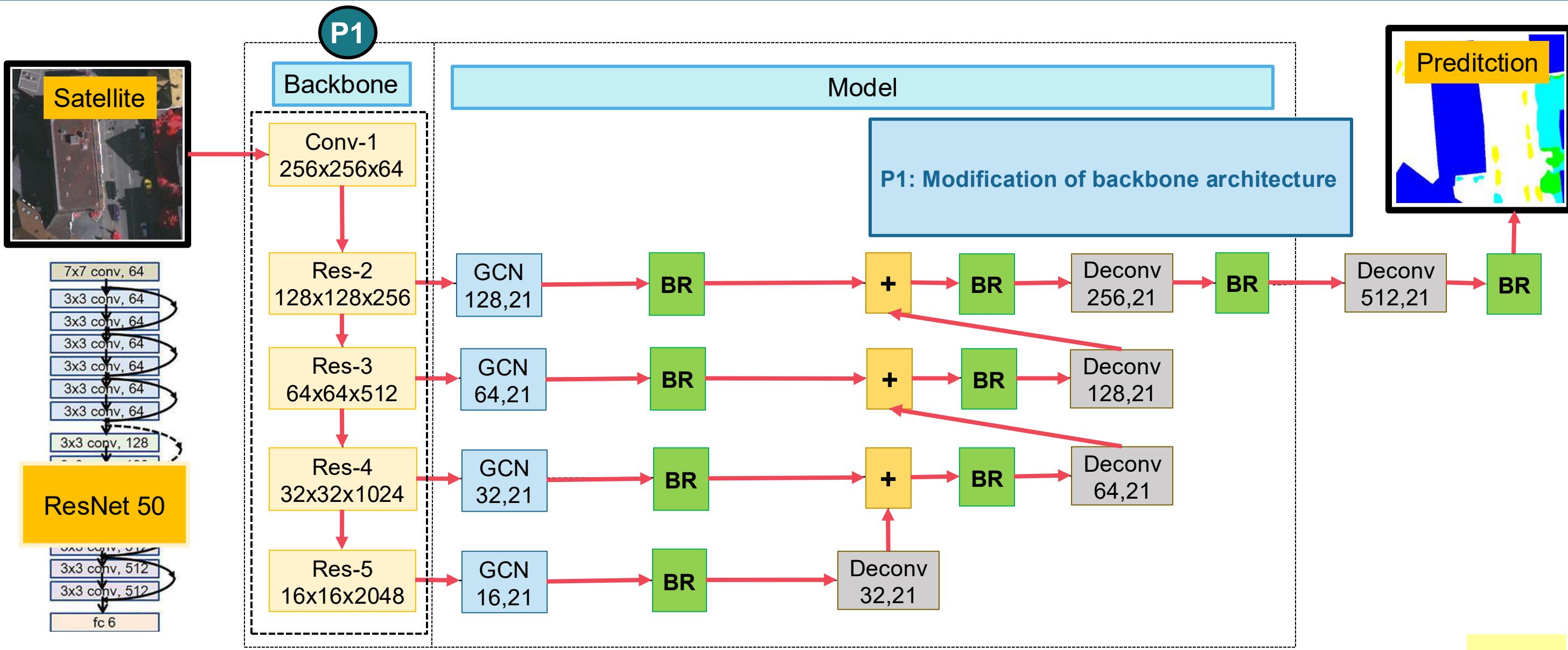
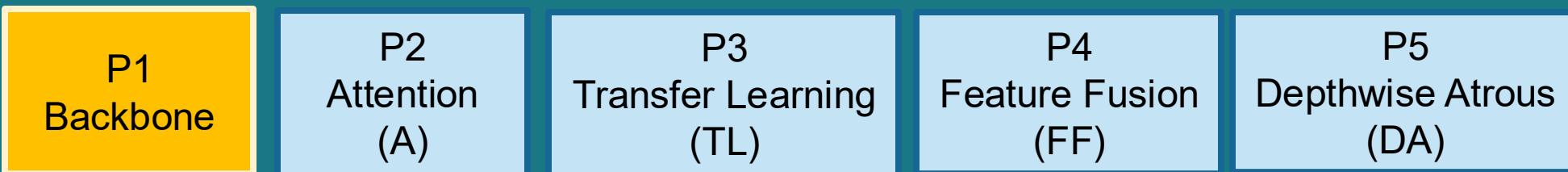
# Proposed Method



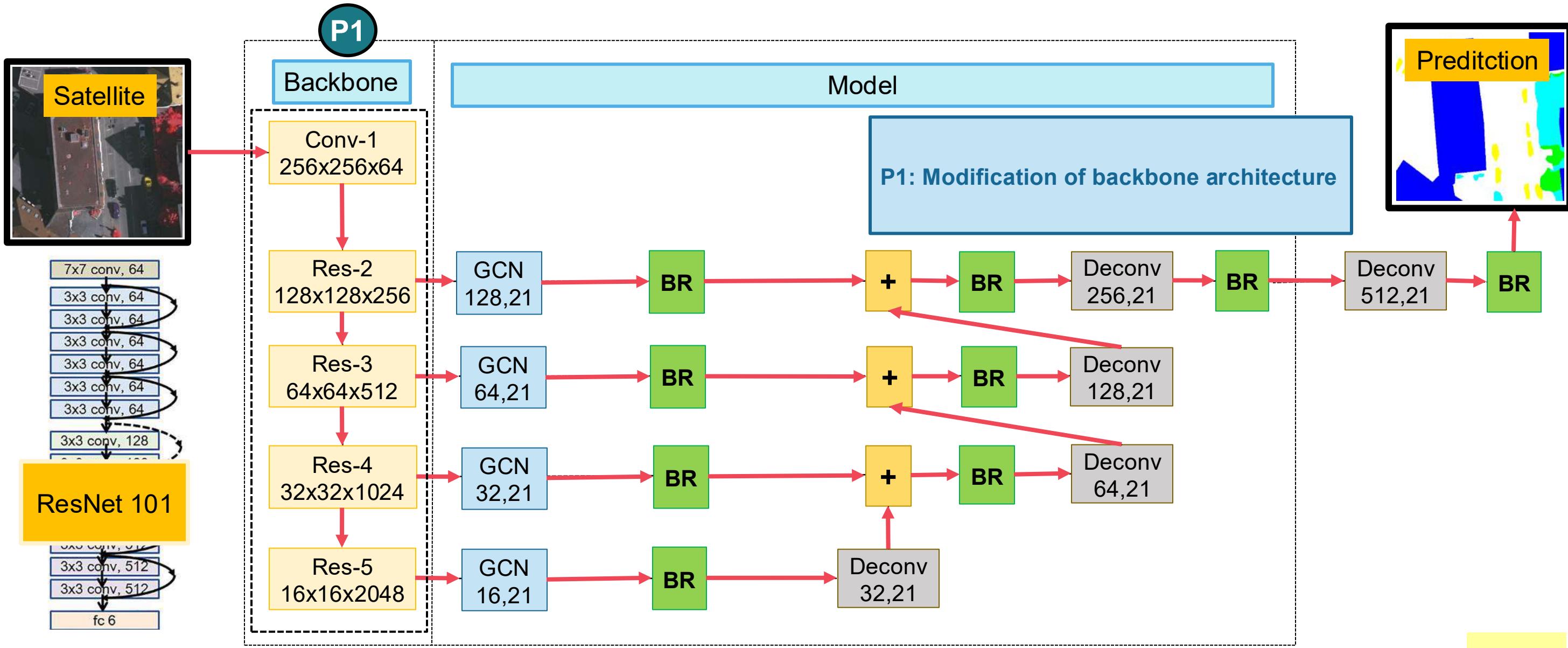
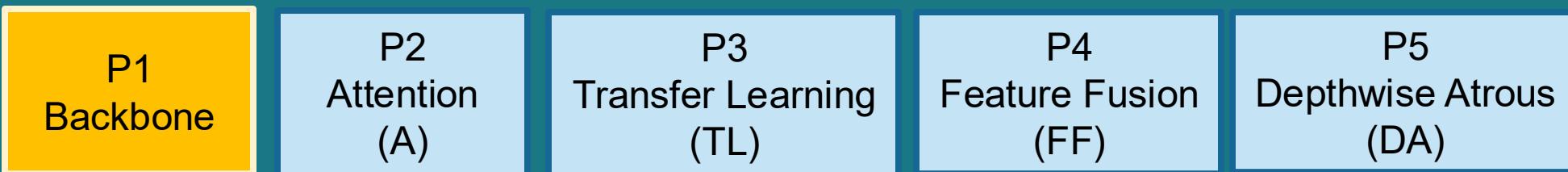
GCN = Global Conv Block

BR = Boundary Refinement Block

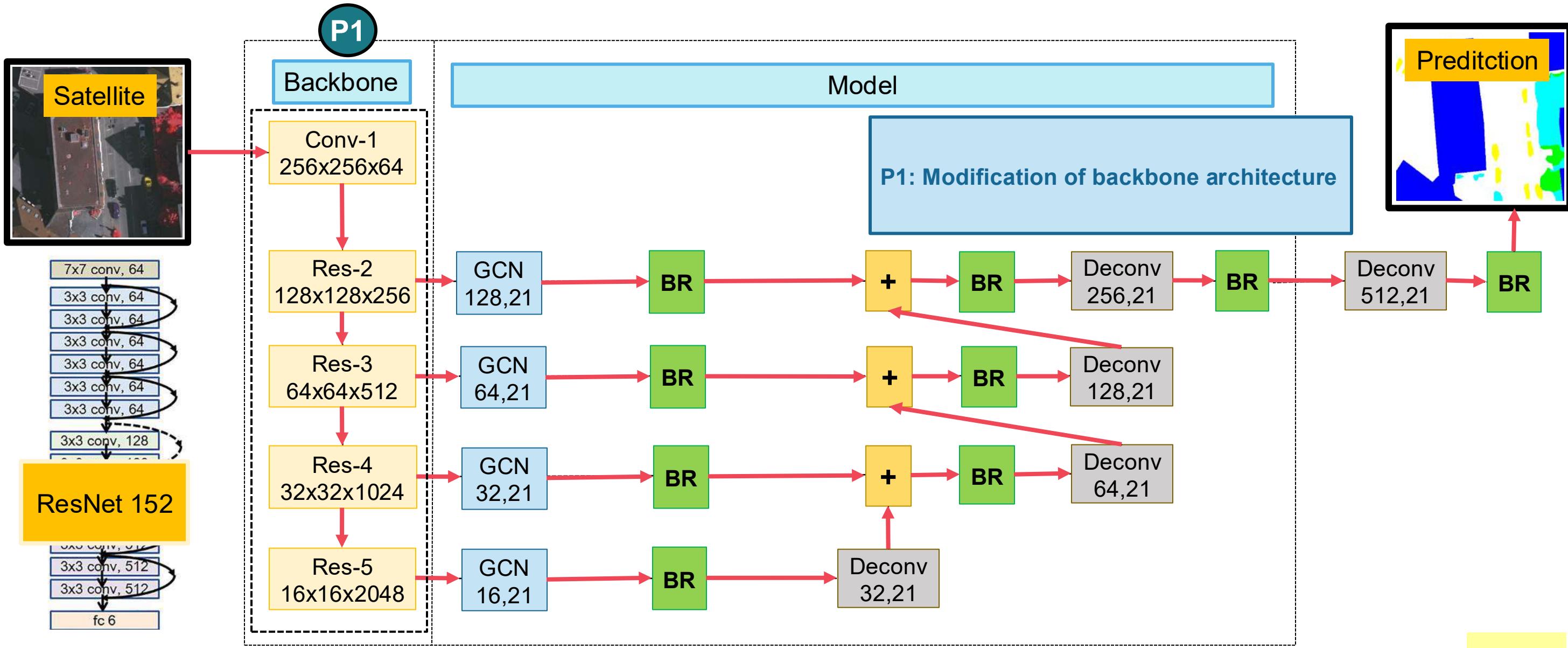
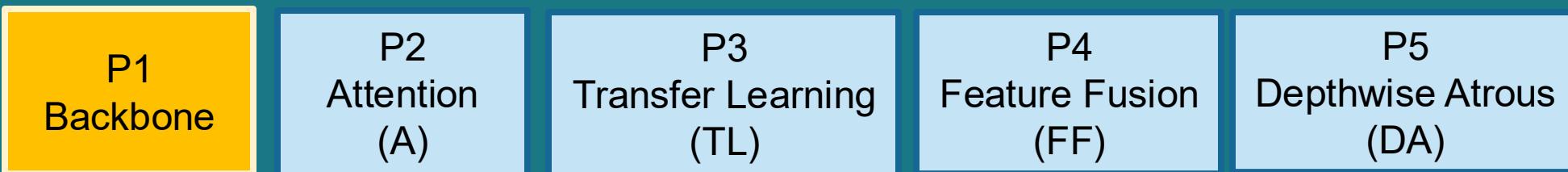
# Proposed Method



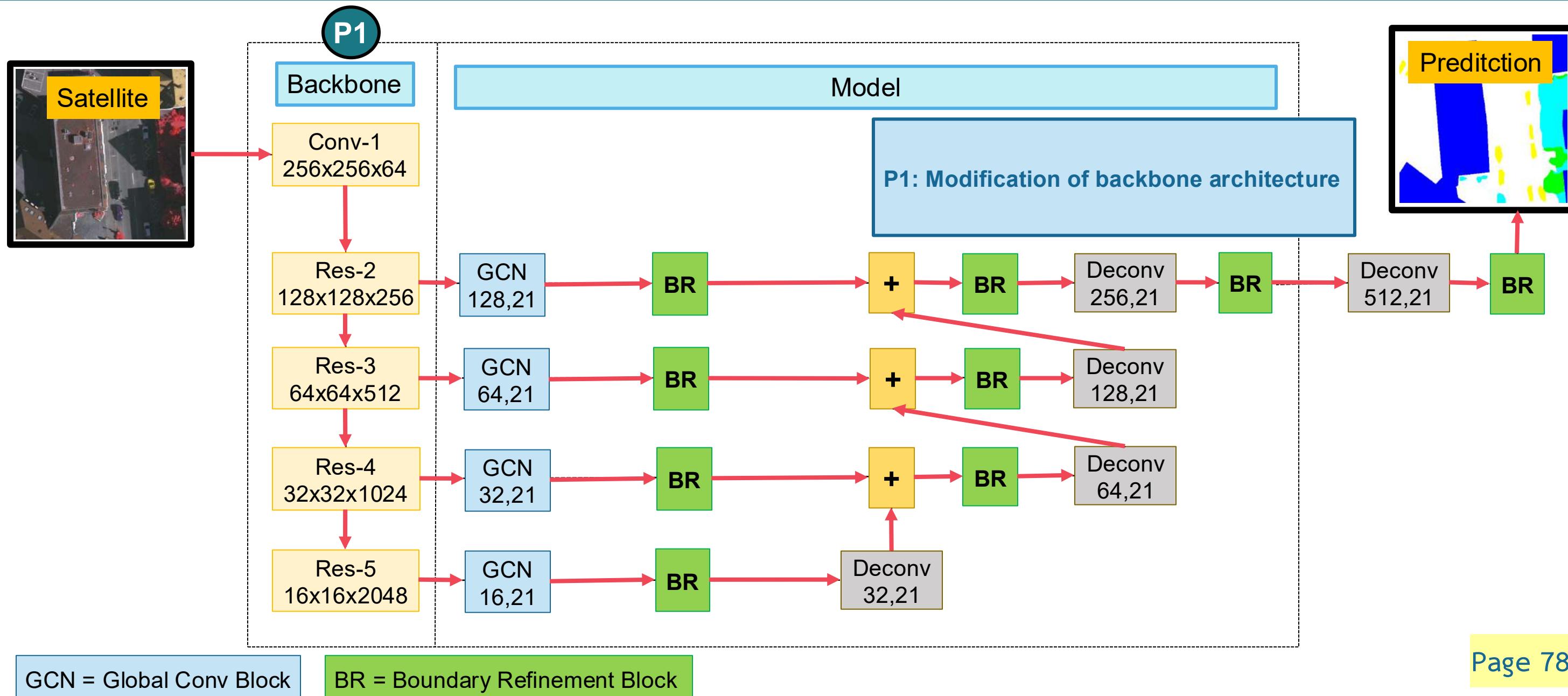
# Proposed Method



# Proposed Method



# Proposed Method



## GCN = Global Conv Block

**BR** = Boundary Refinement Block

Page 78

# Proposed Method

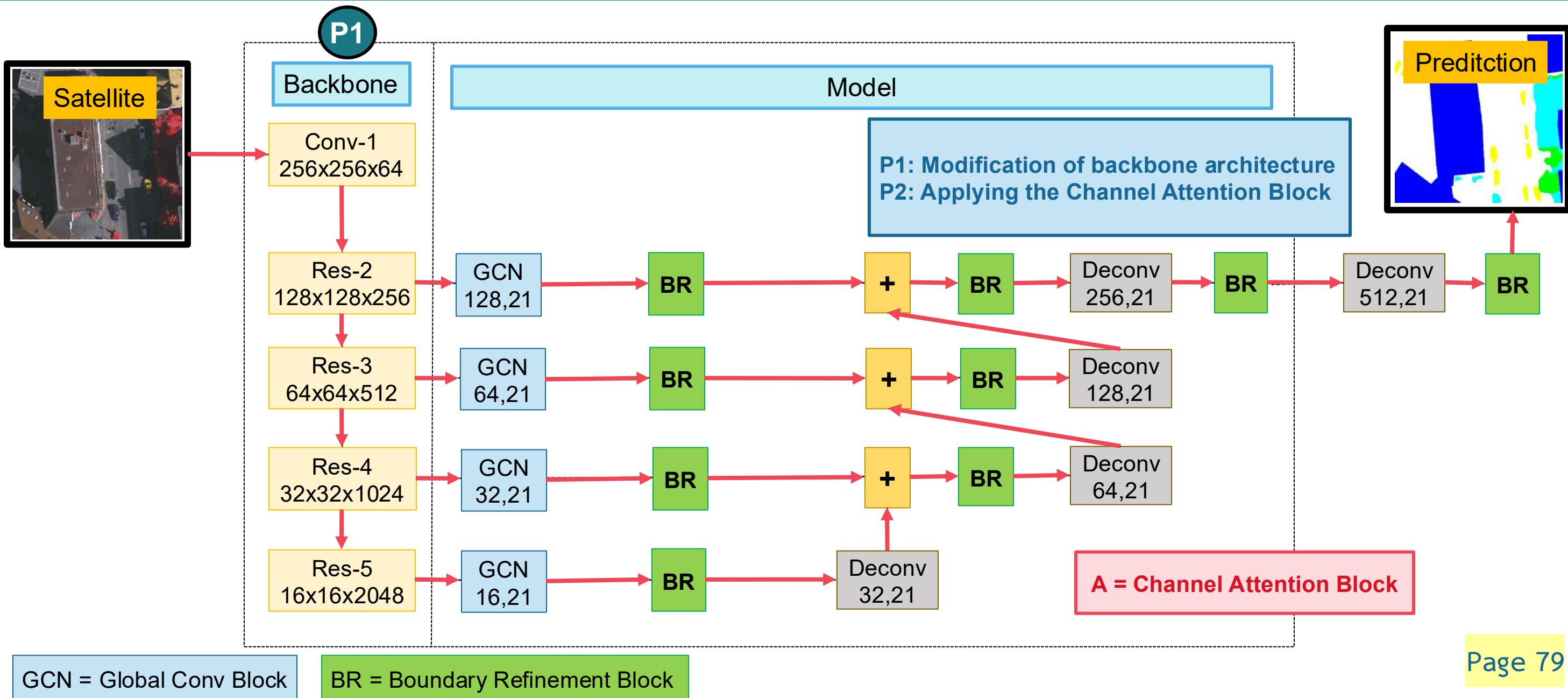
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

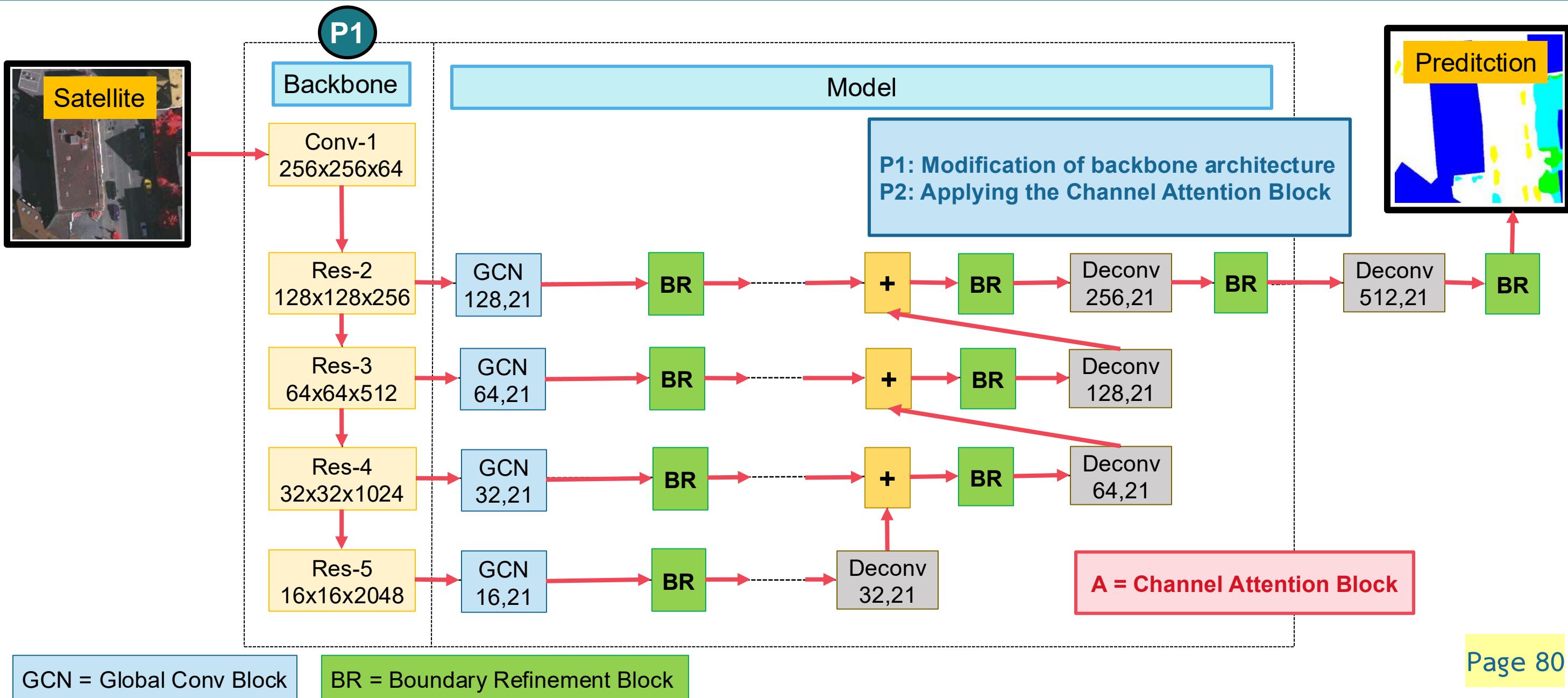
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

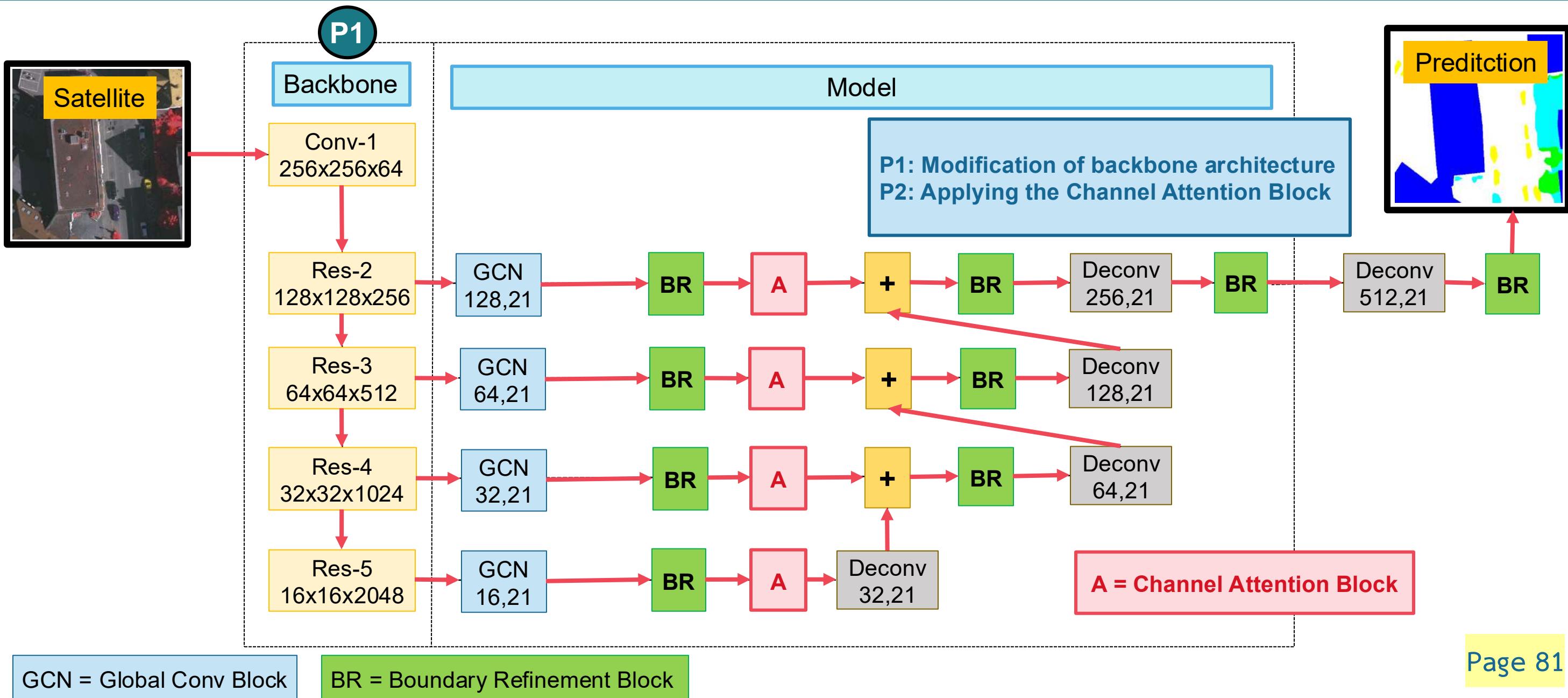
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

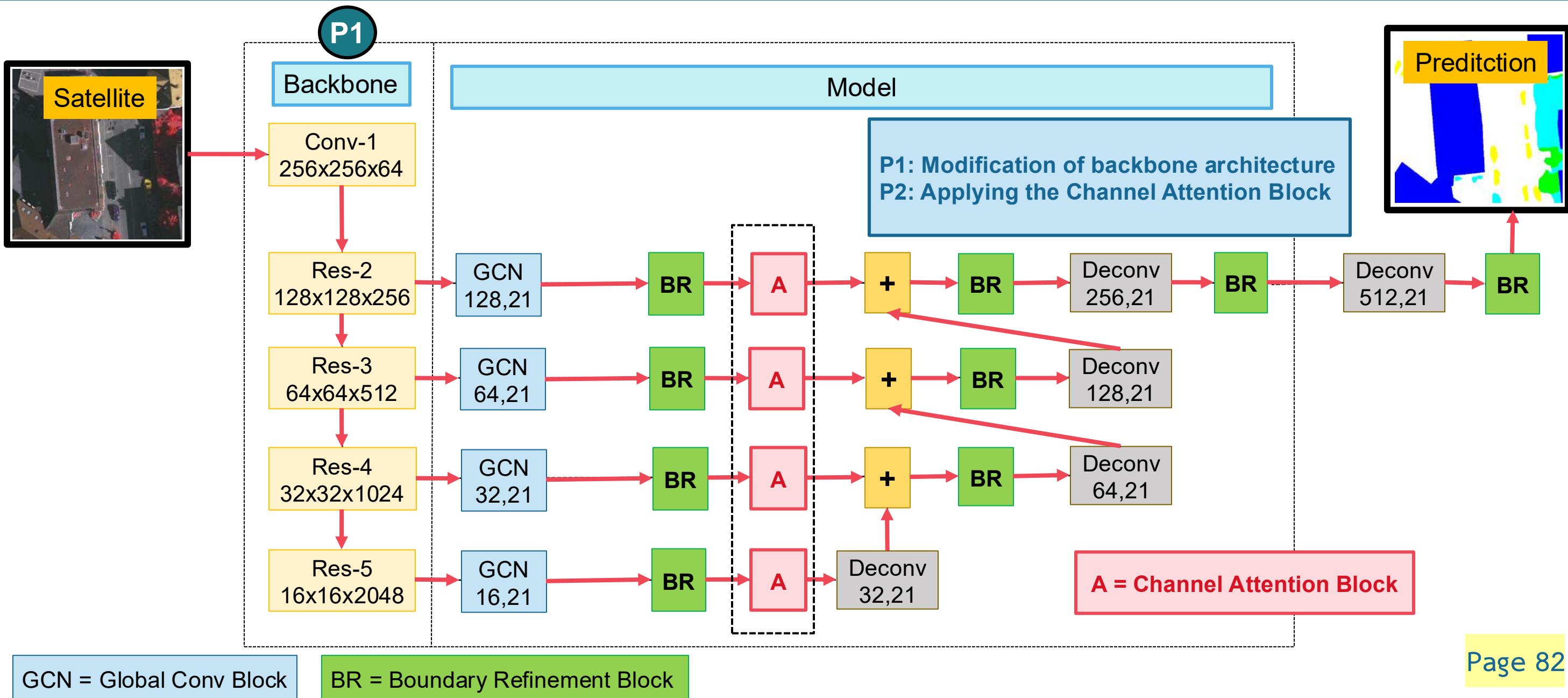
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

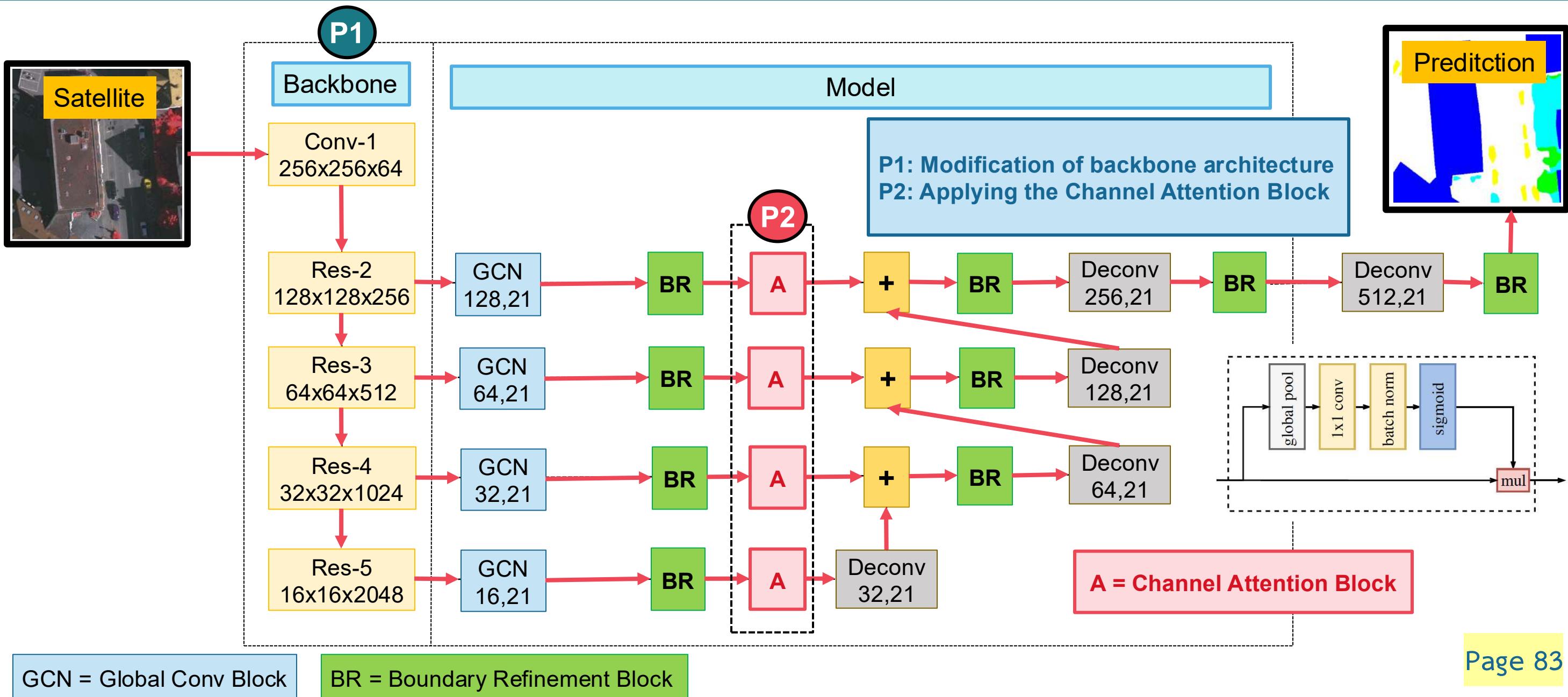
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

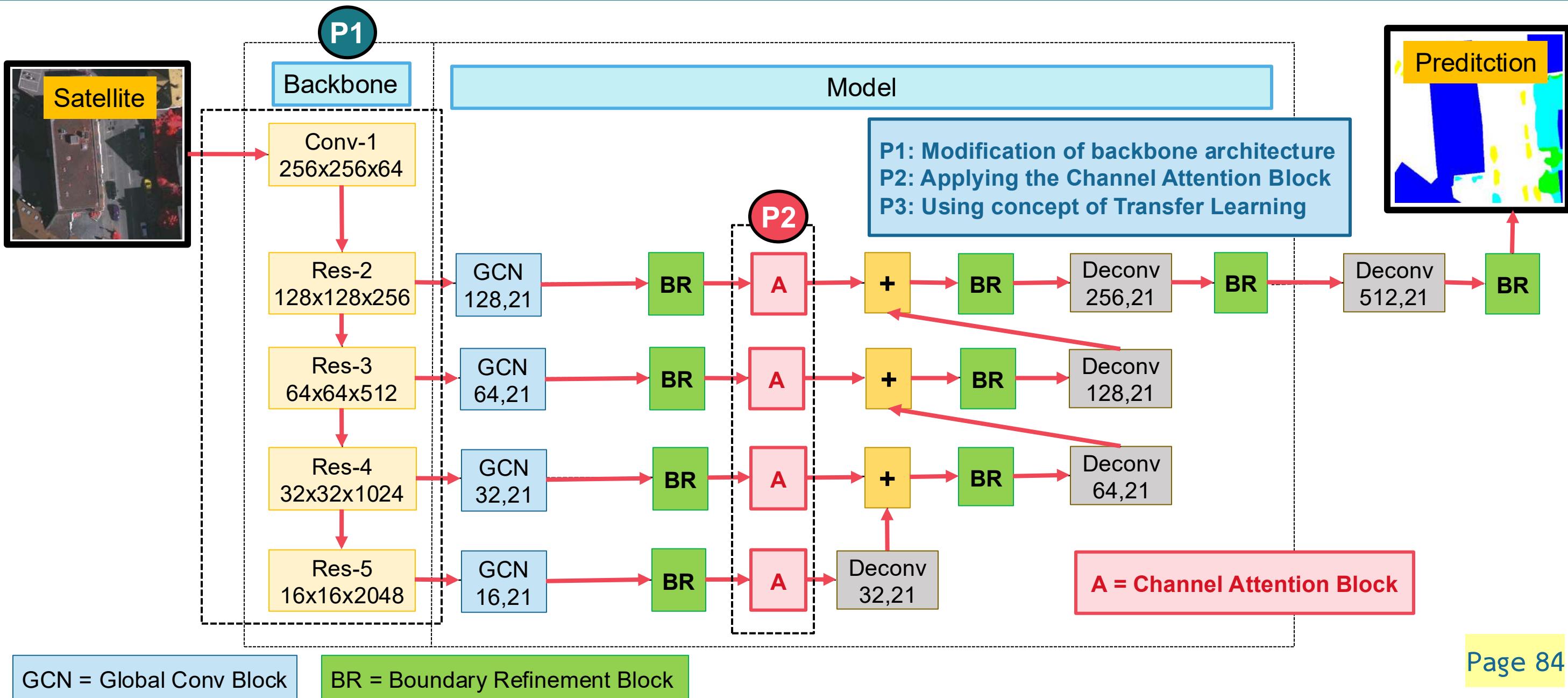
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

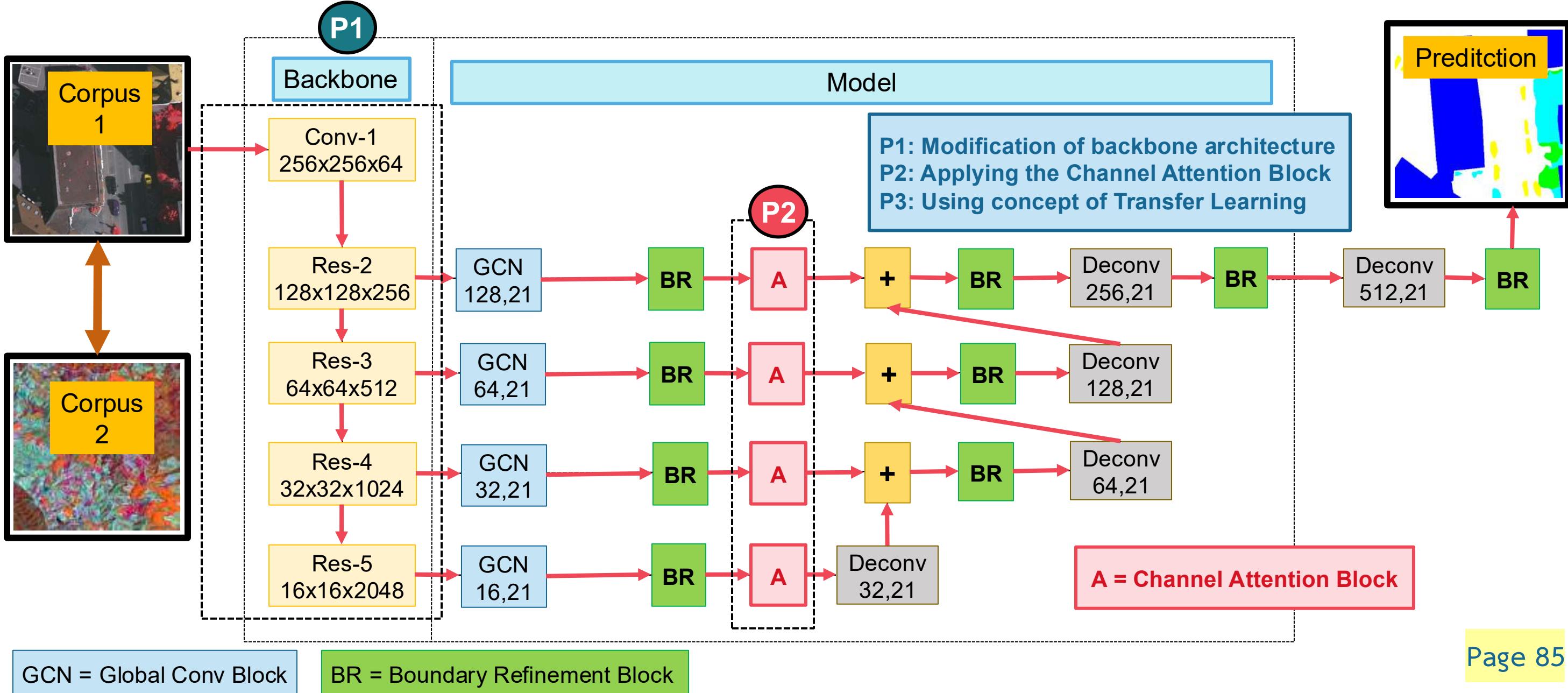
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

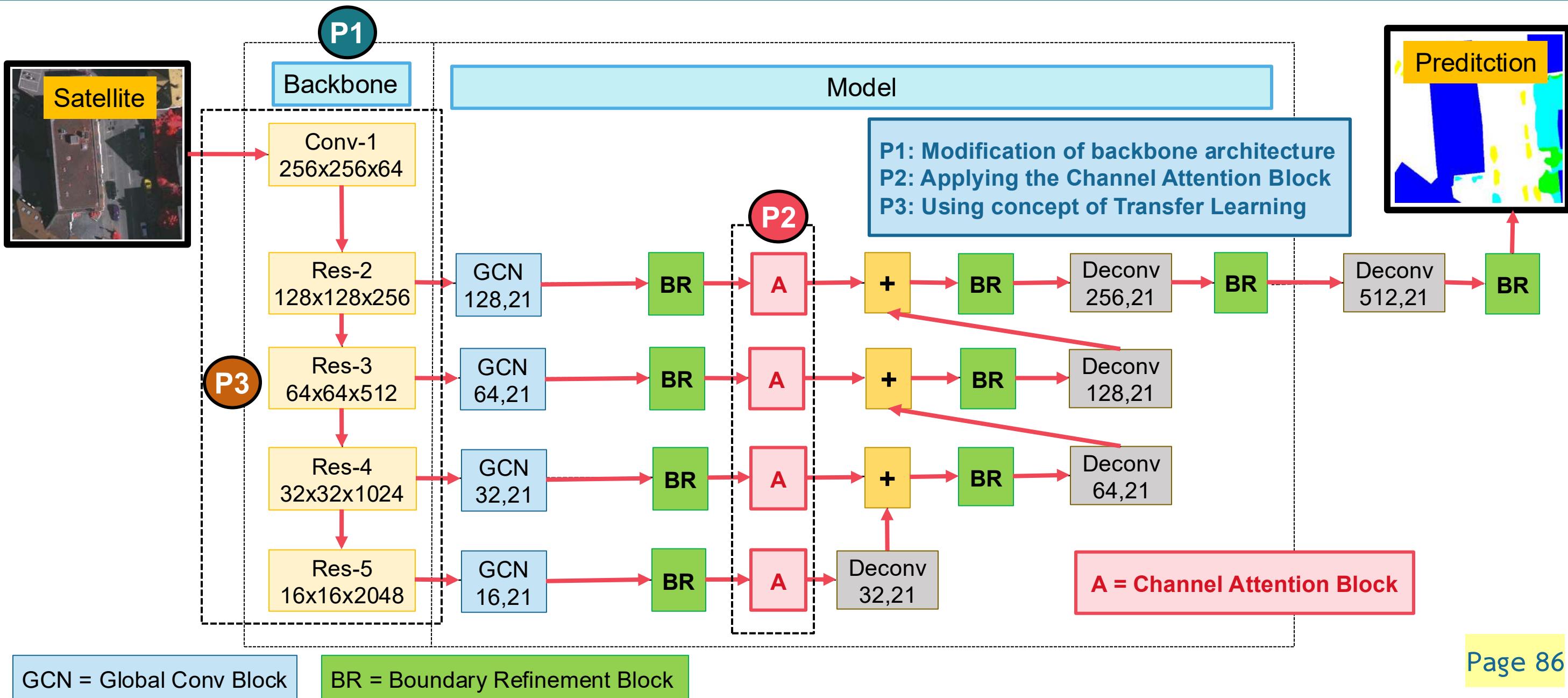
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

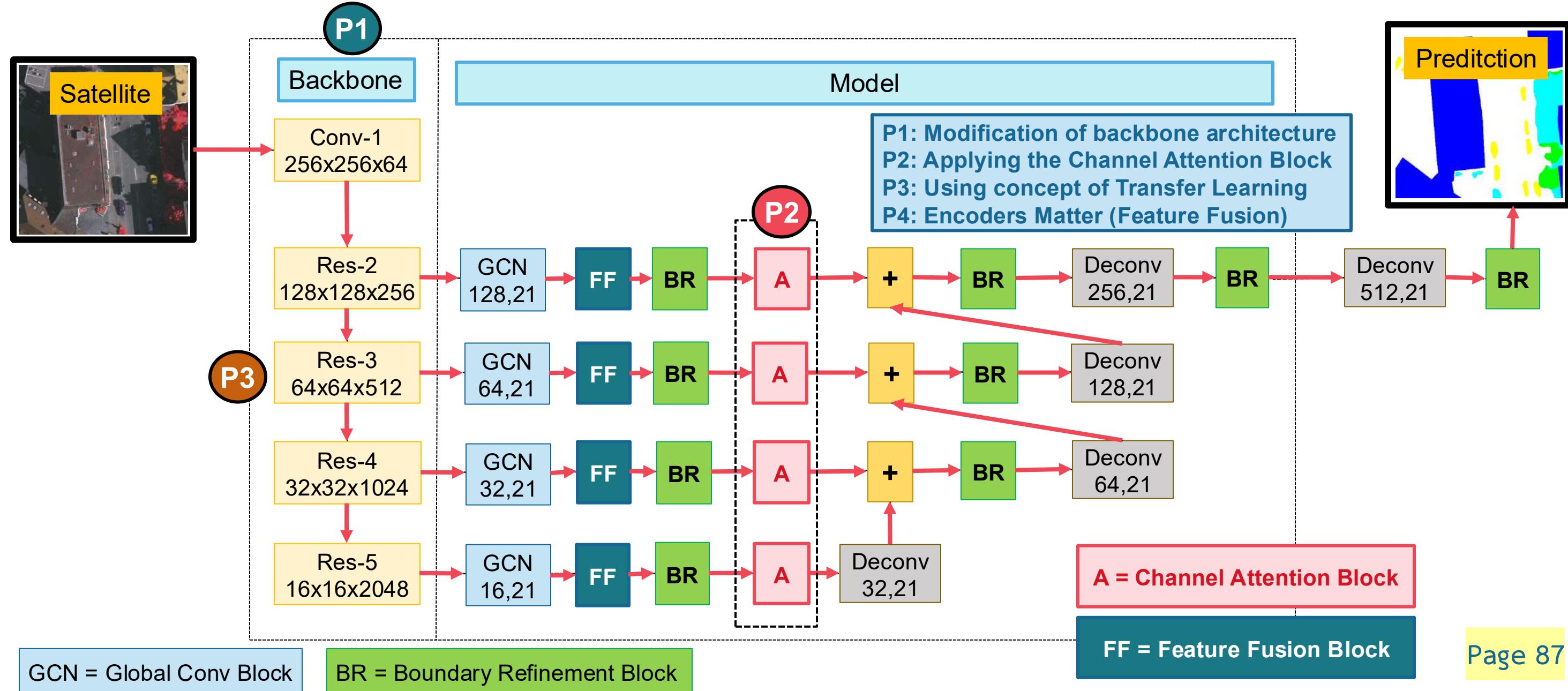
P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)



# Proposed Method

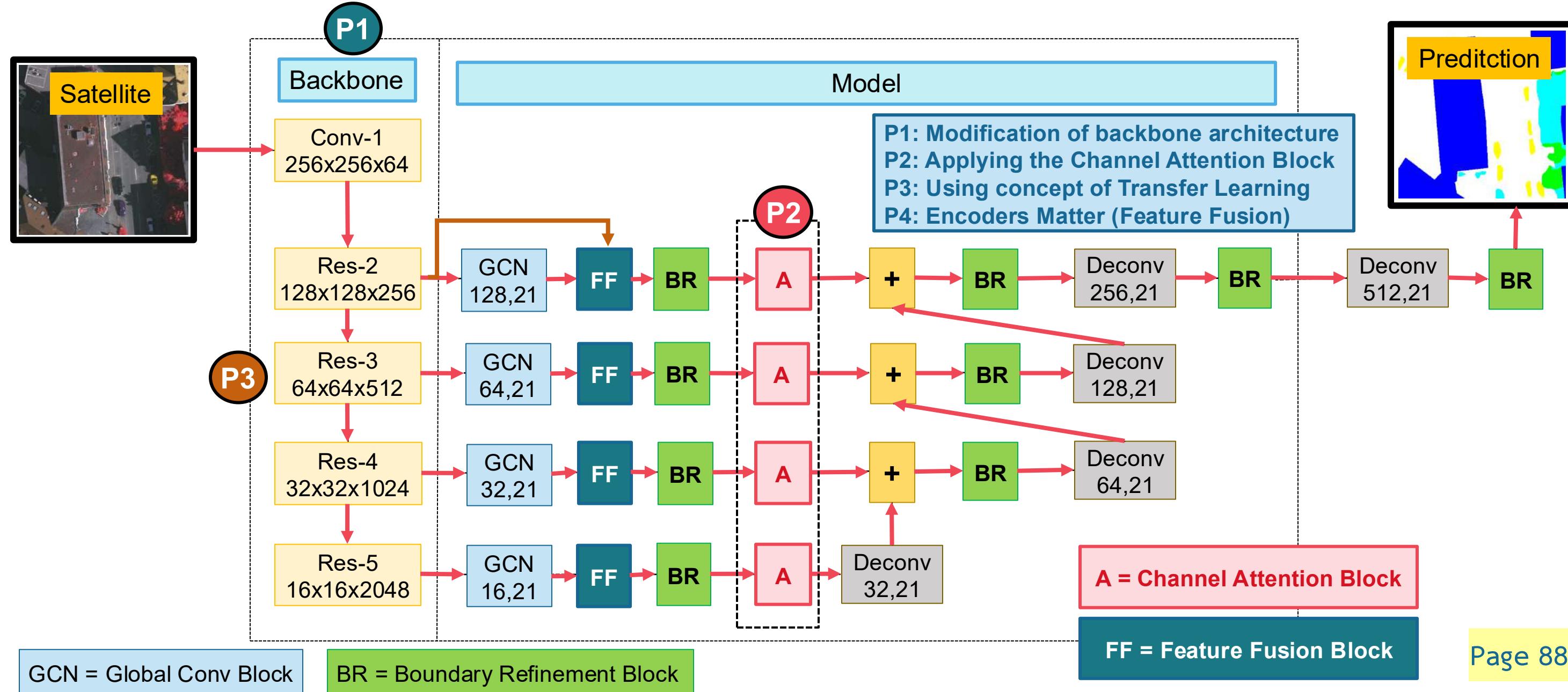
P1  
Backbone

P2  
Attention  
(A)

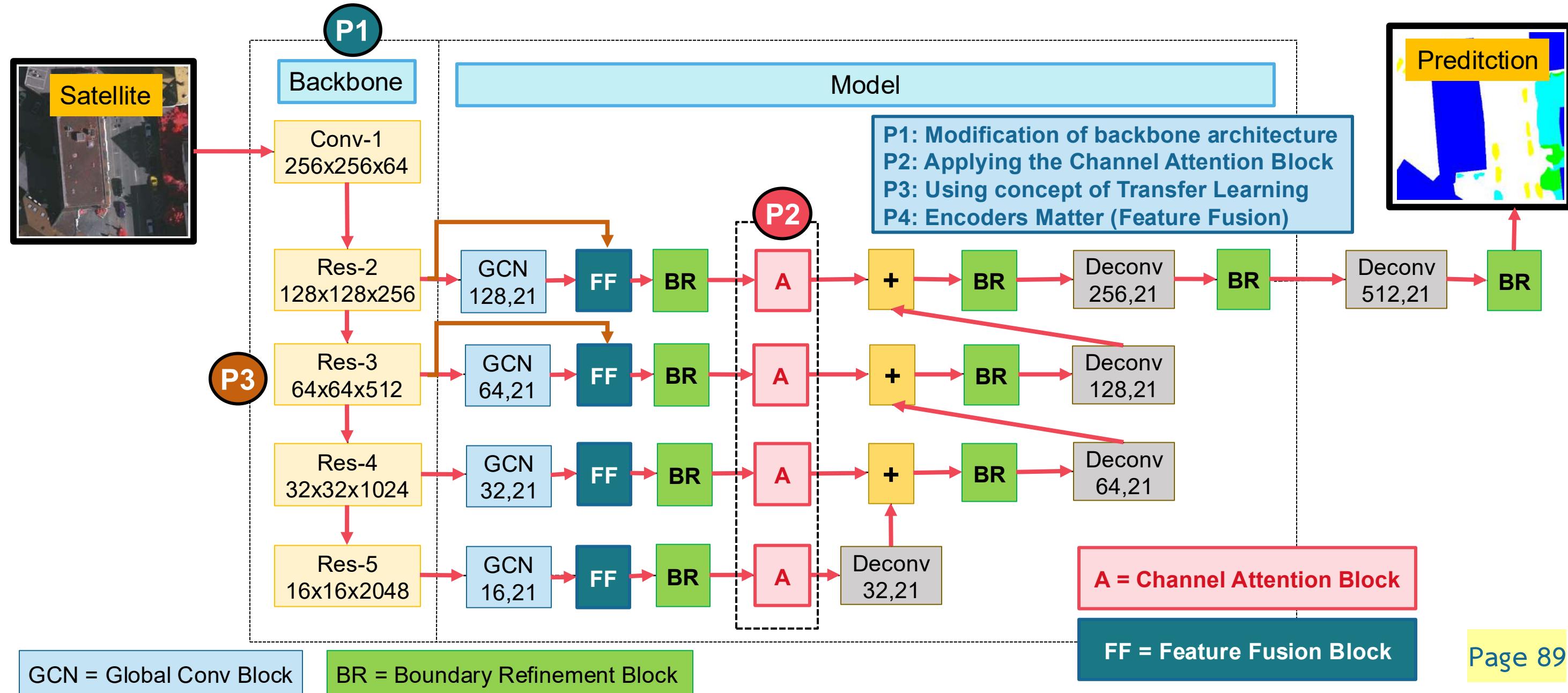
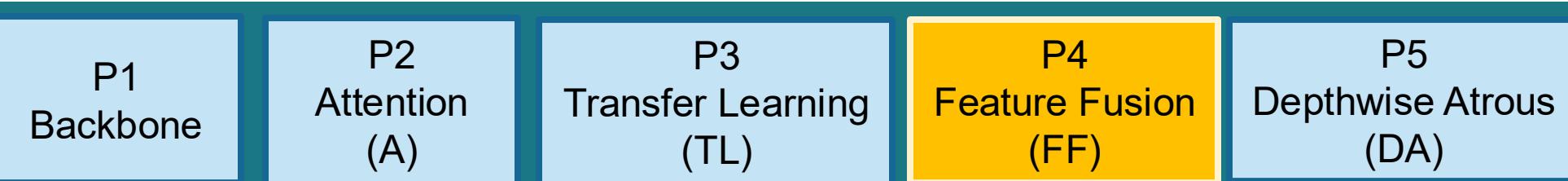
P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

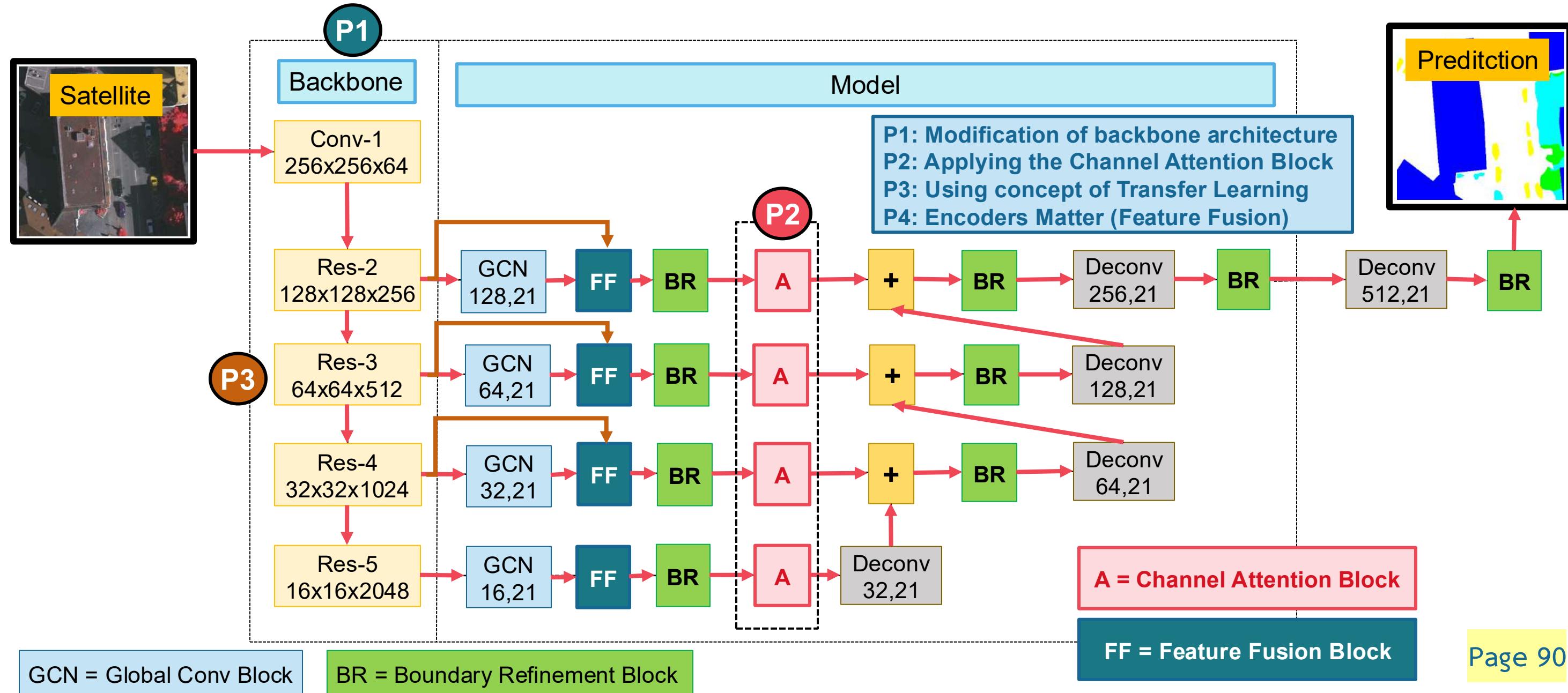
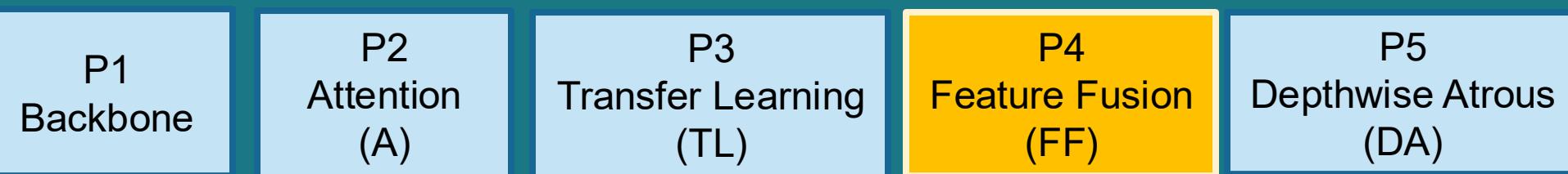
P5  
Depthwise Atrous  
(DA)



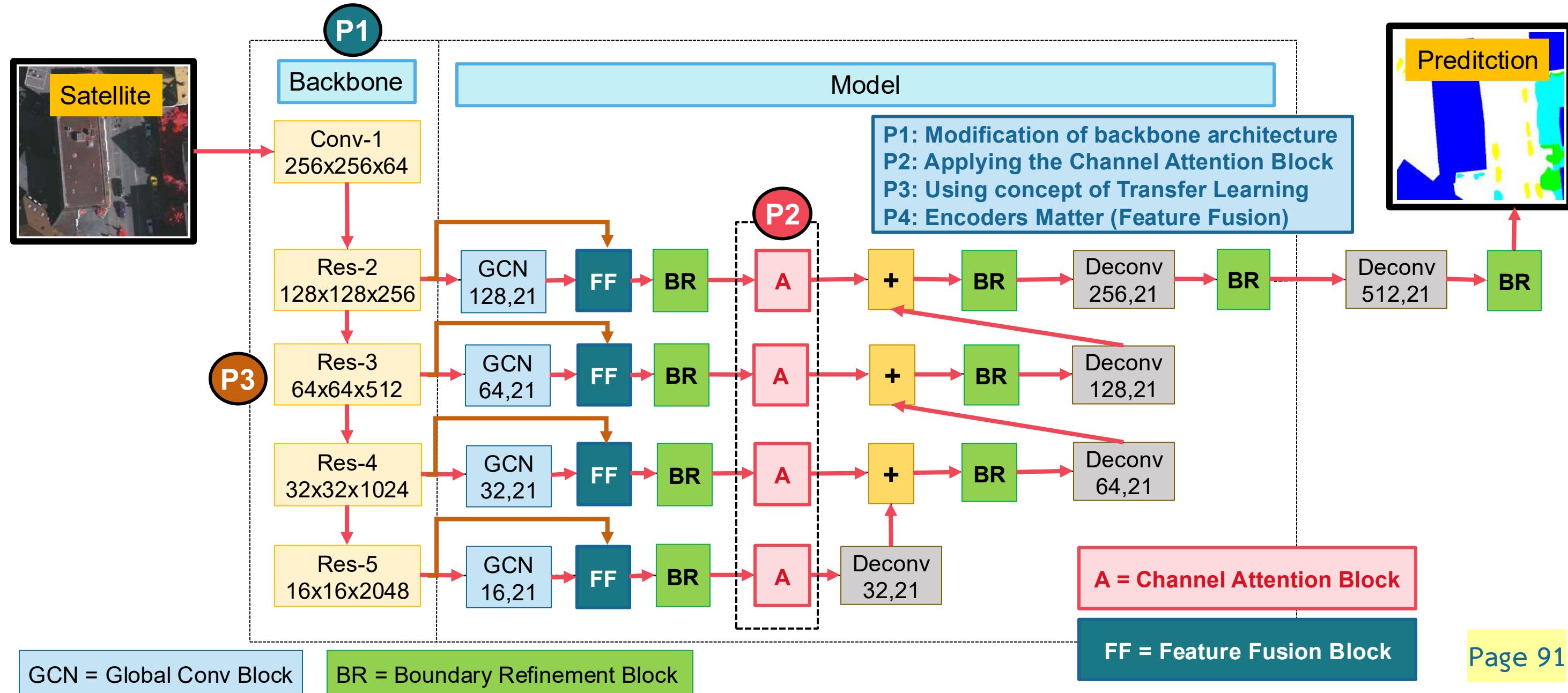
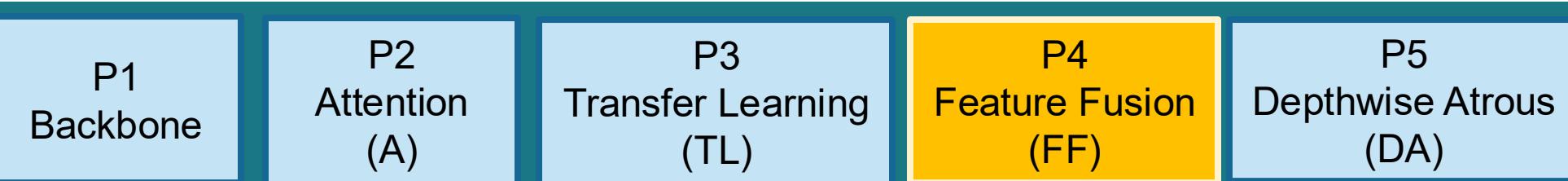
# Proposed Method



# Proposed Method



# Proposed Method



# Proposed Method

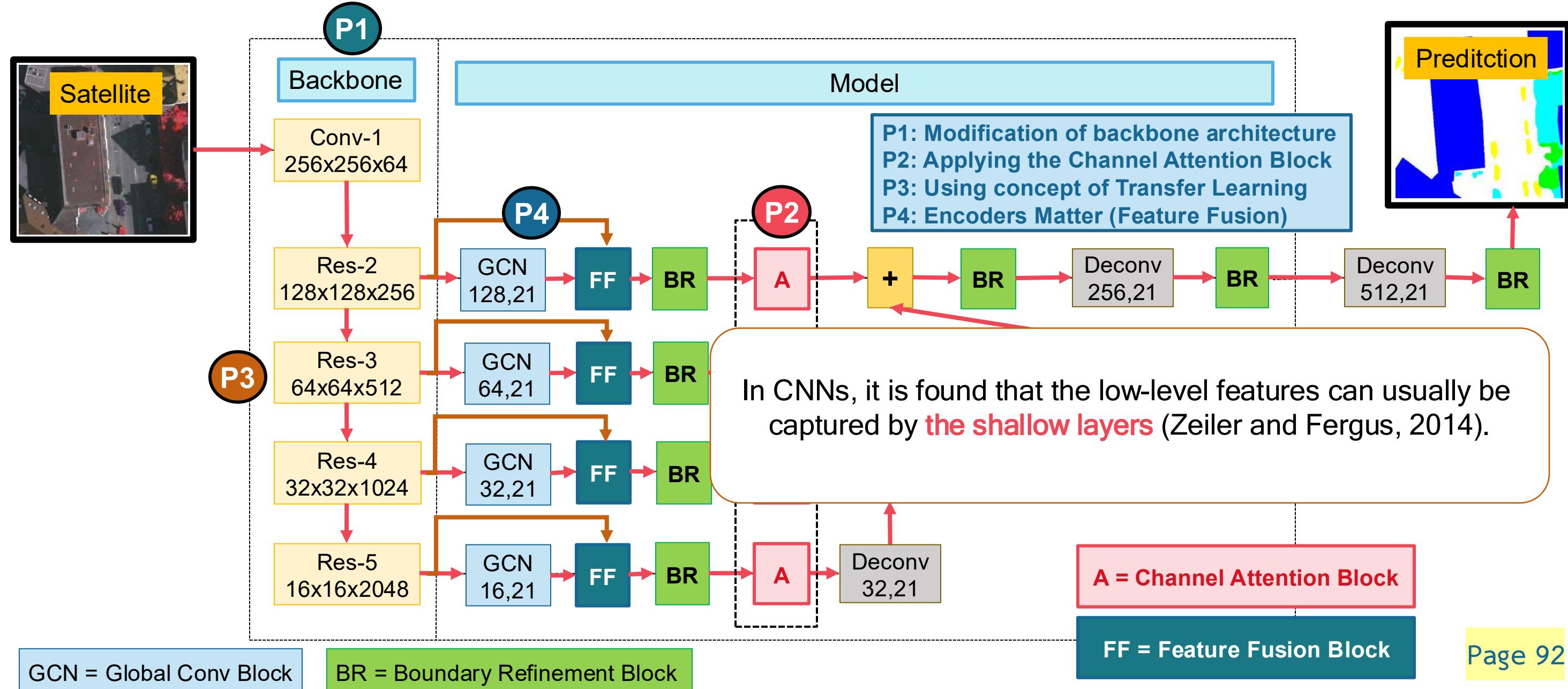
P1  
Backbone

P2  
Attention  
(A)

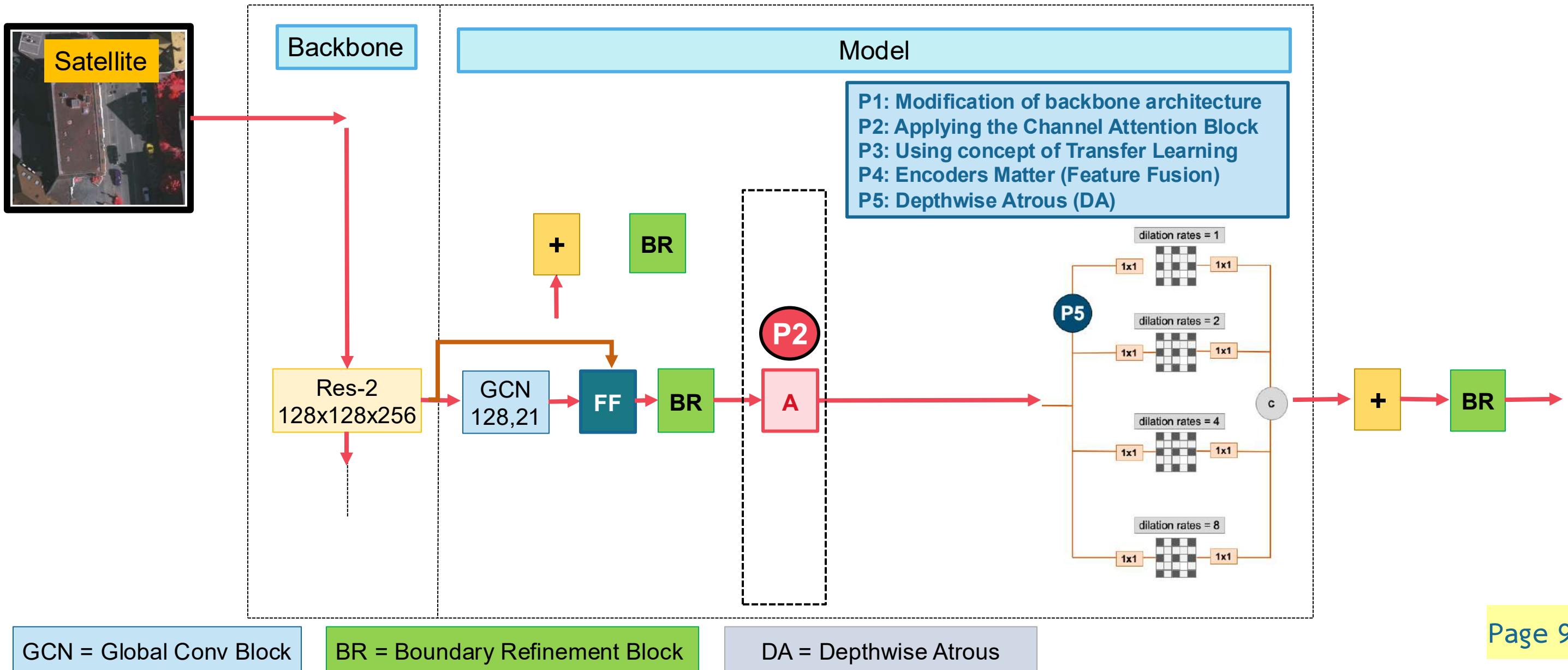
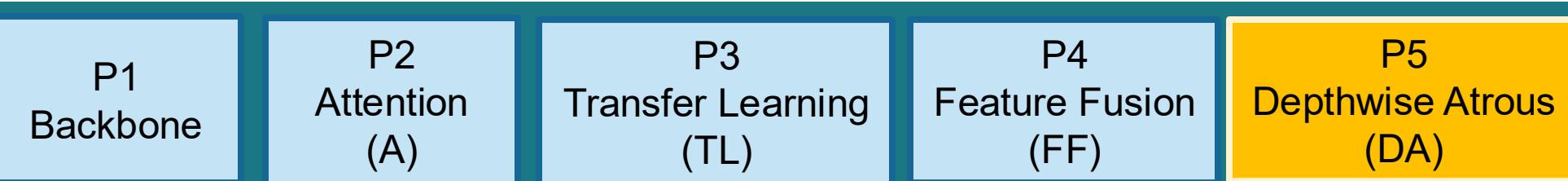
P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

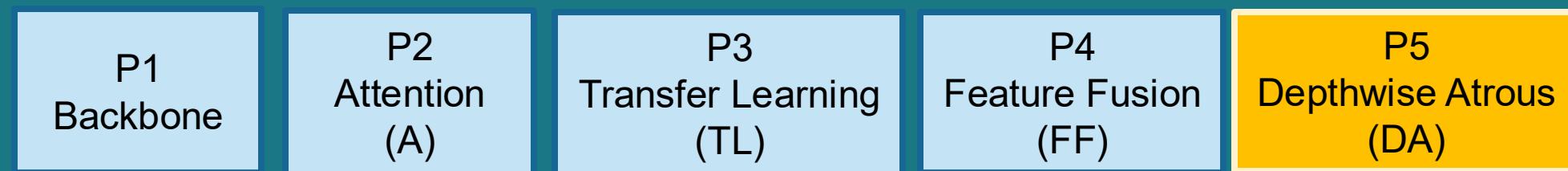
P5  
Depthwise Atrous  
(DA)



# Proposed Method



# Proposed Method



GCN = Global Conv Block

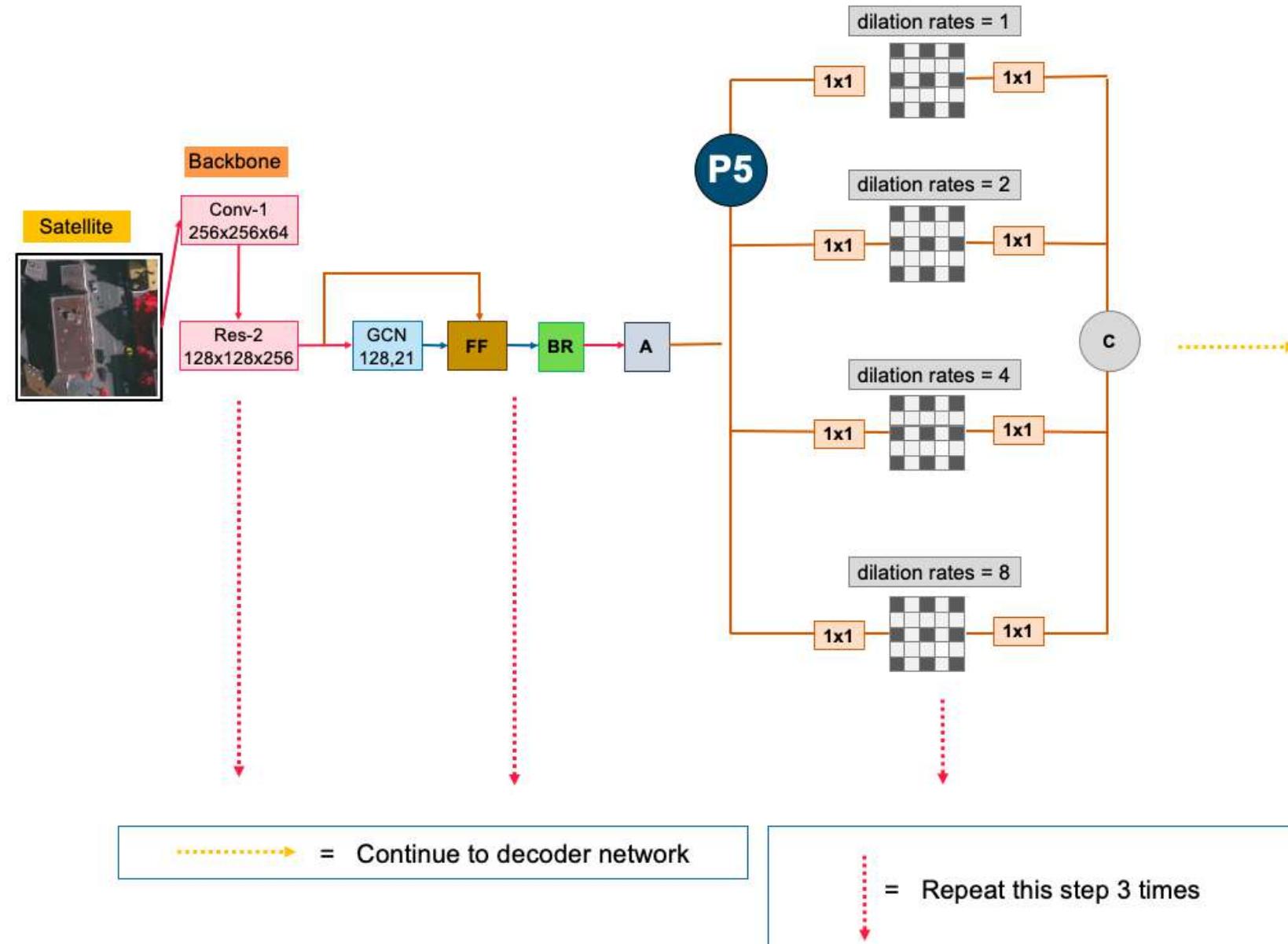
BR = Boundary Refinement Block

A = Channel Attention Block

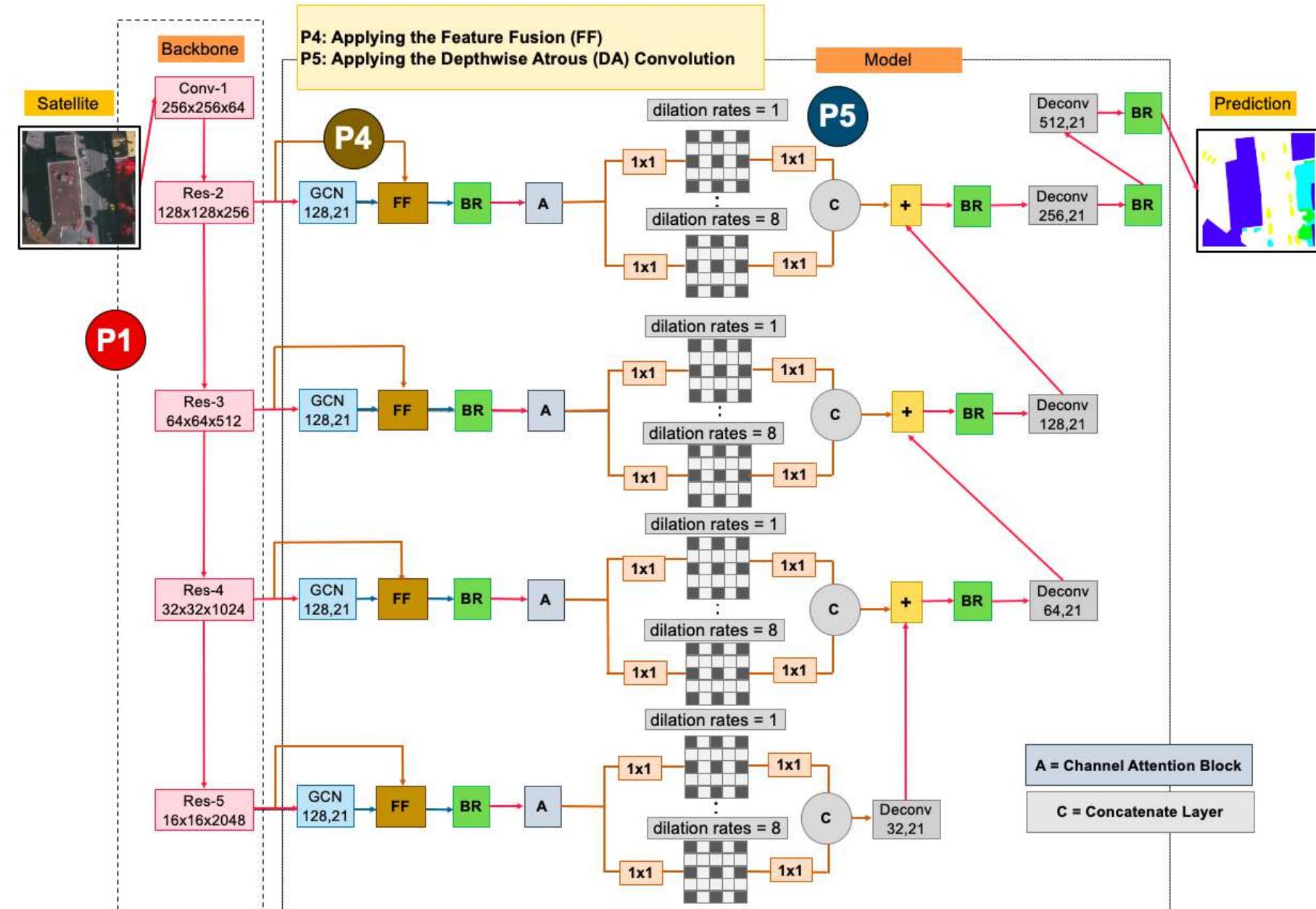
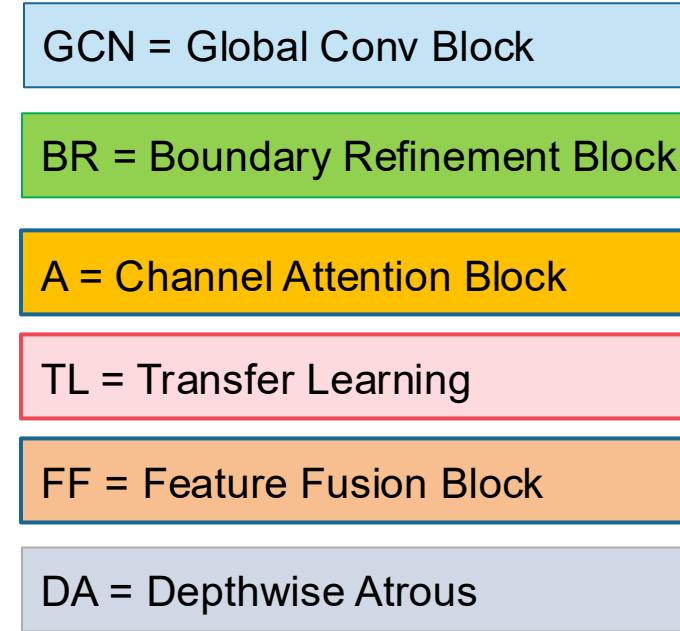
TL = Transfer Learning

FF = Feature Fusion Block

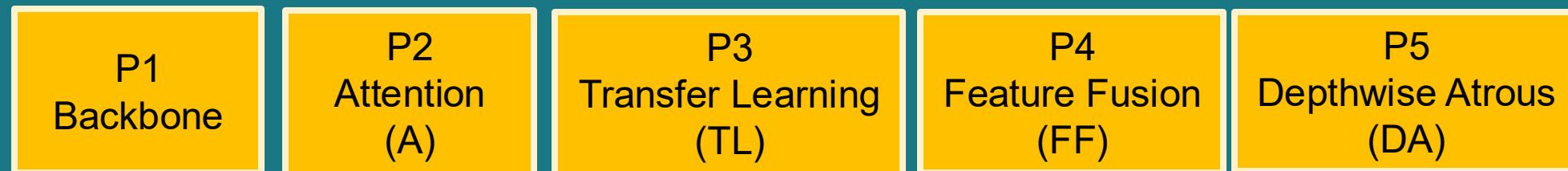
DA = Depthwise Atrous



# Proposed Method



# Proposed Method



GCN = Global Conv Block

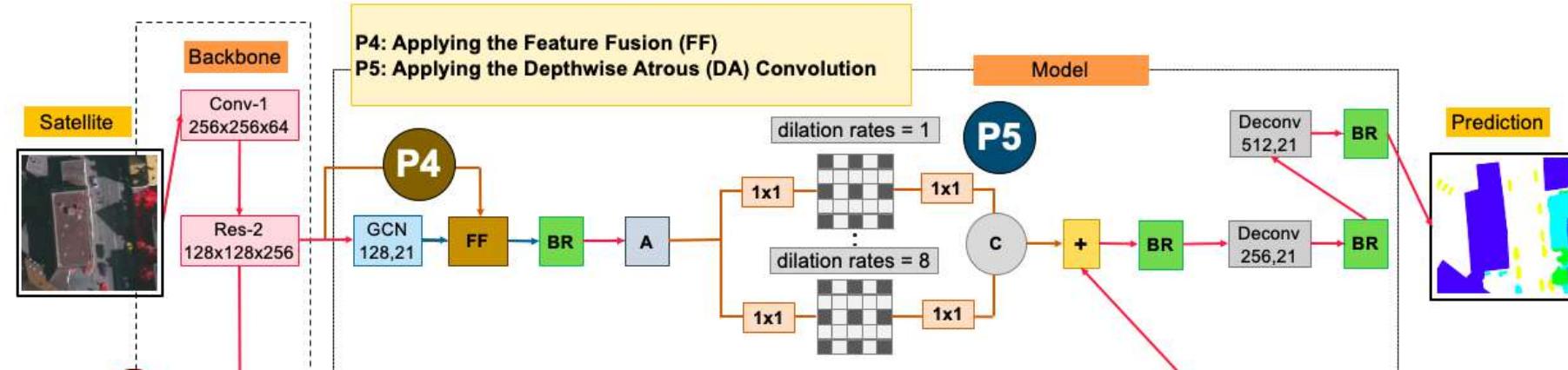
BR = Boundary Refinement Block

A = Channel Attention Block

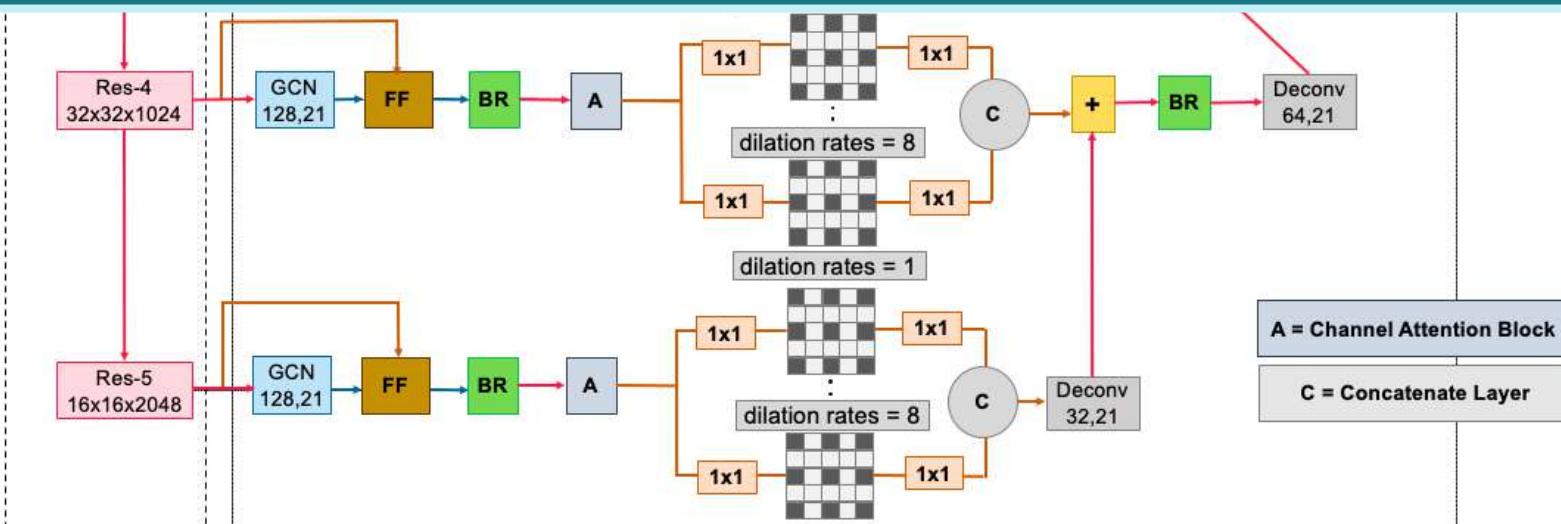
TL = Tr

FF = Fe

DA = D



The Whole of Proposed Method : Encoders Matter



# Outline | Experimental Results

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- **Experimental Results**
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Evaluation

Corpus 1  
ISPRS Vaihingen

Corpus 2  
Nan, Thailand

Corpus 3  
Isan, Thailand

Abbreviation	Description
A	Channel Attention Block
GCN	Global Convolutional Network
GCN50	Global Convolutional Network with ResNet50
GCN101	Global Convolutional Network with ResNet101
GCN152	Global Convolutional Network with ResNet52
TL	Domain-Specific Transfer Learning
FF	Feature Fusion Module
DA	Depthwise Atrous Convolution

Abbreviations on our proposed deep learning methods

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Performance Metrics

# Recap: Each Methods from Proposed

GCN = Global Conv Block

A = Channel Attention Block

TL = Transfer Learning

FF = Feature Fusion Block

DA = Depthwise Atrous

P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)

- **Experiment 1:** How it impacts modern and over-deeper backbone?
- **Experiment 2:** Chanel Attention
- **Experiment 3:** Deep CNNs with Domain Specific Transfer Learning
- **Experiment 4:** Feature Fusion
- **Experiment 5:** Depthwise Atrous Convolution
- **Three data sets:** two private corpora from Landsat-8 satellite (Nan and Isan Region) and one public benchmark from the “ISPRS Vaihingen” challenge.

1<sup>st</sup> Corpus  
Nan, Thailand (Medium Resolution Corpus)



# Evaluation

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	-	Res50	GCN	0.881	0.872	0.875
	-	Res101	GCN	0.862	0.897	0.877
	-	Res152	GCN	0.892	0.878	0.884
	-	Res152	GCN-A	0.907	0.929	0.917
	TL	Res152	GCN-A	0.921	0.918	0.918
	TL	Res152	GCN-A-FF	0.930	0.924	0.927
	TL	Res152	GCN-A-FF-DA	0.934	0.939	0.936

Result: Our proposed method yields a higher F1 Score from baseline method at 6.2%

# Evaluation

Corpus 1  
Nan, Thailand

Corpus 2  
ISPRS Vaihingen

Corpus 3  
Isan, Thailand

- Each class

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN50	0.967	0.948	0.817	0.881	0.792
	GCN101	0.976	0.929	0.685	0.929	0.785
	GCN152	0.976	0.950	0.823	0.913	0.797
	GCN152-A	0.984	0.944	0.882	0.899	0.822
	GCN152-TL-A	0.974	0.953	0.864	0.934	0.828
	GCN152-TL-A-FF	0.986	0.982	0.918	0.956	0.844
	GCN152-TL-A-FF-DA	0.989	0.957	0.934	0.949	0.868

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	-	Res50	GCN	0.881	0.872	0.875
	-	Res101	GCN	0.862	0.897	0.877
	-	Res152	GCN	0.892	0.878	0.884

- GCN50 overcame DECD ~ 0.116 % F1
- GCN152 overcame DECD ~ 1.043 % F1

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?

- Each class

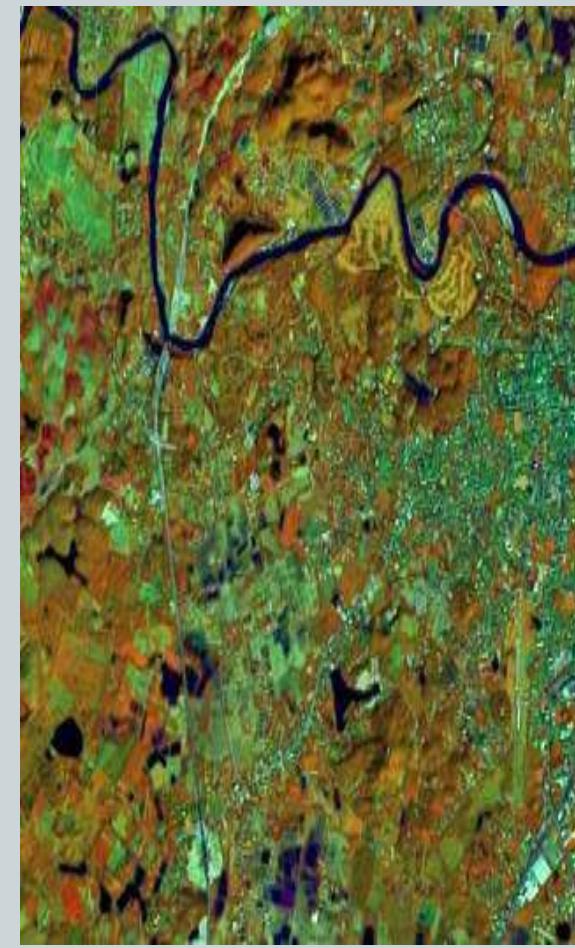
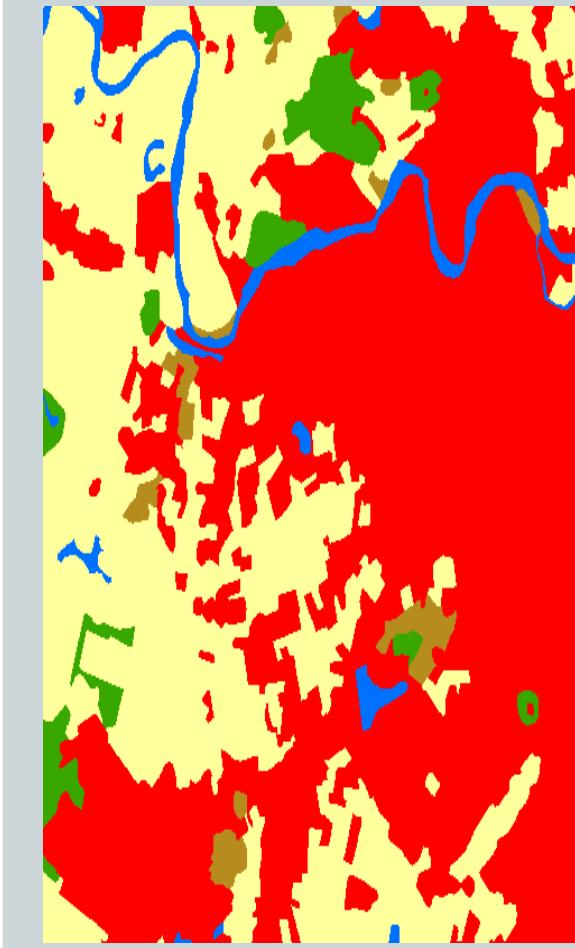
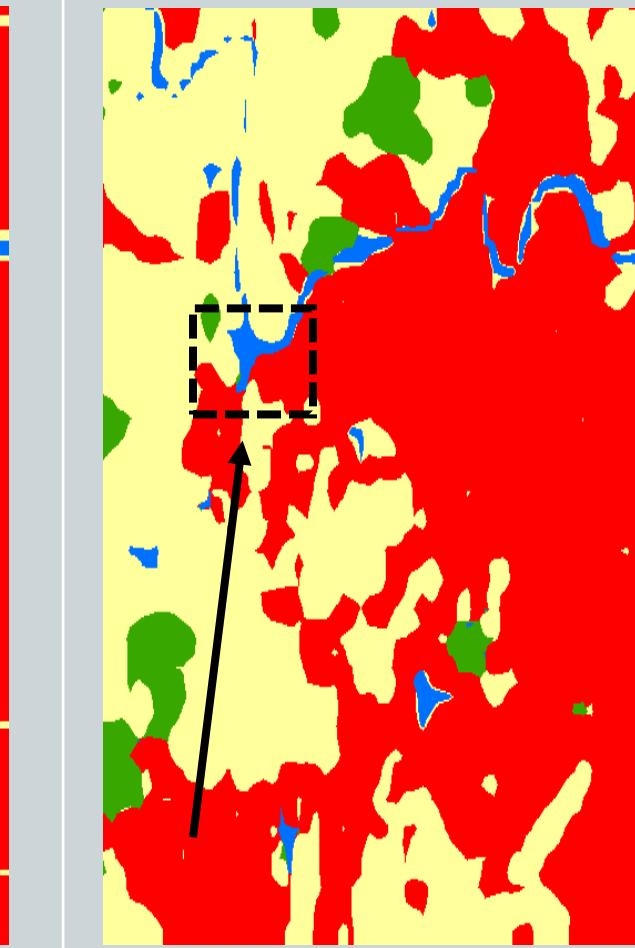
Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN50	0.967	0.948	0.817	0.881	0.792
	GCN101	0.976	0.929	0.685	0.929	0.785
	GCN152	0.976	0.950	0.823	0.913	0.797

- GCN Family won DECD **4 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

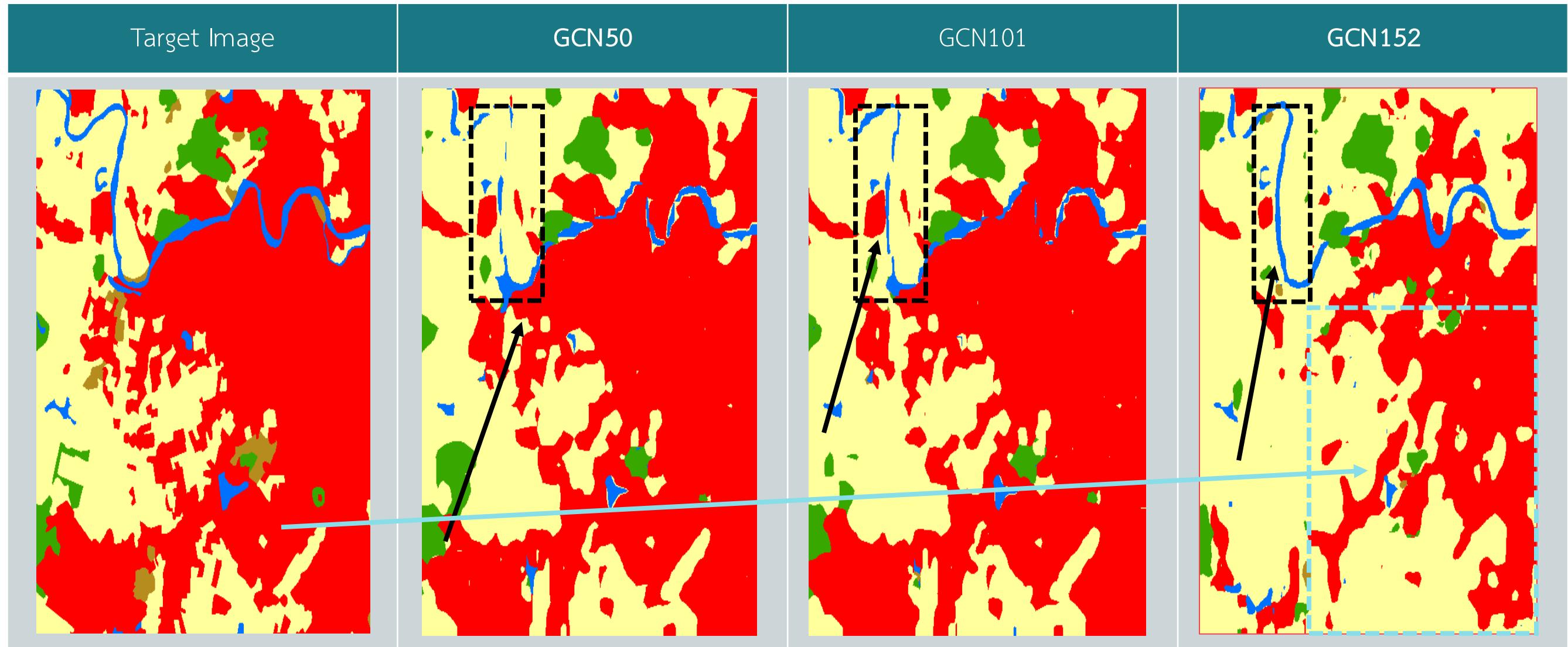
Experiment 1:  
How it impacts modern and over-deeper backbone?

Input Image	Target Image	Baseline Method [10]	GCN50
			

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?



# Evaluation

- Precision
- Recall
- F1-score

Corpus 1  
Nan, Thailand

Experiment 2:  
Chanel Attention

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	-	Res152	GCN	0.892	0.878	0.884
	-	Res152	GCN-A	0.907	0.929	0.917

- GCN152-A overcame DECD ~ 4.332 % F1
- GCN152-A overcame GCN152 ~ 3.288 % F1

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 2:  
Chanel Attention

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN152	0.976	0.950	0.823	0.913	0.797
	GCN152-A	0.984	0.944	0.882	0.899	0.822

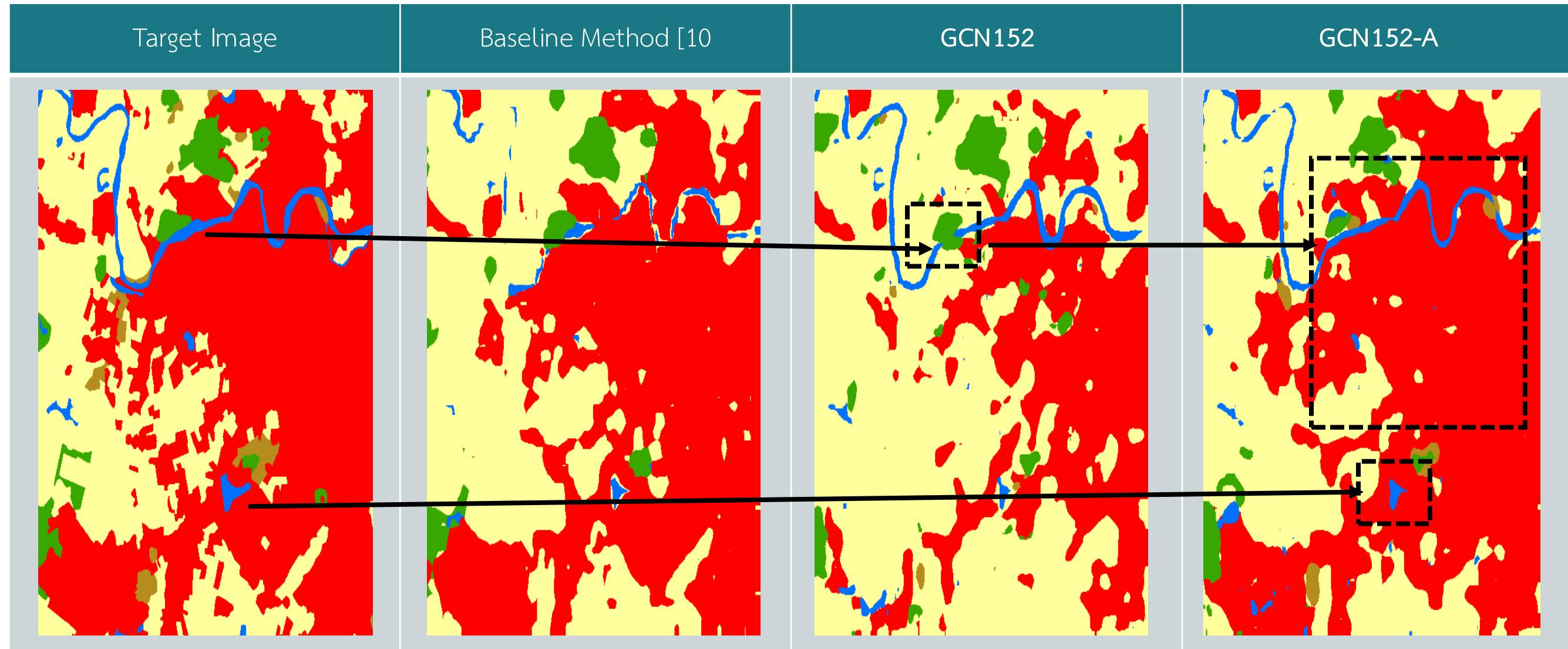
- Each class

- Our Proposed won DECD **4 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 2:  
Chanel Attention



# Evaluation

Corpus 1  
Nan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	-	Res152	GCN-A	0.907	0.929	0.917
	TL	Res152	GCN-A	0.921	0.918	0.918

- GCN152-A-TL overcame DECD ~ 4.446 % F1
- GCN152-A-TL overcame GCN152-A ~ 0.114 % F1

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN152-A	0.984	0.944	0.882	0.899	0.822
	GCN152-TL-A	0.974	0.953	0.864	0.934	0.828

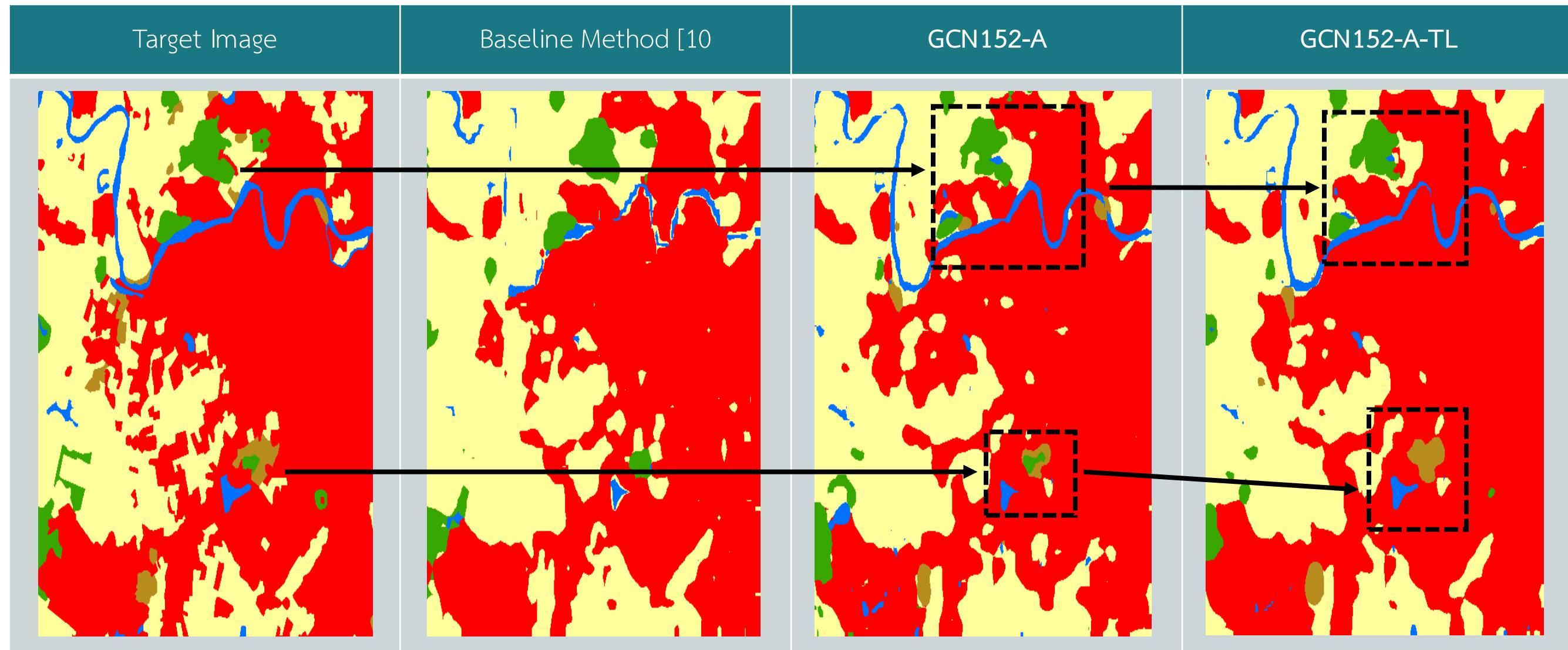
- Each class

- Our Proposed won DECD **4 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning



# Evaluation

Corpus 1  
Nan, Thailand

Experiment 4:  
Feature Fusion

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	TL	Res152	GCN-A	0.921	0.918	0.918
	TL	Res152	GCN-A-FF	0.930	0.924	0.927

- GCN152-A-TL-FF overcame DECD ~ 5.288 % F1
- GCN152-A-TL-FF overcame GCN152-A-TL ~ 0.843 % F1

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 4:  
Feature Fusion

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN152-TL-A	0.974	0.953	0.864	0.934	0.828
	GCN152-TL-A-FF	0.986	0.982	0.918	0.956	0.844

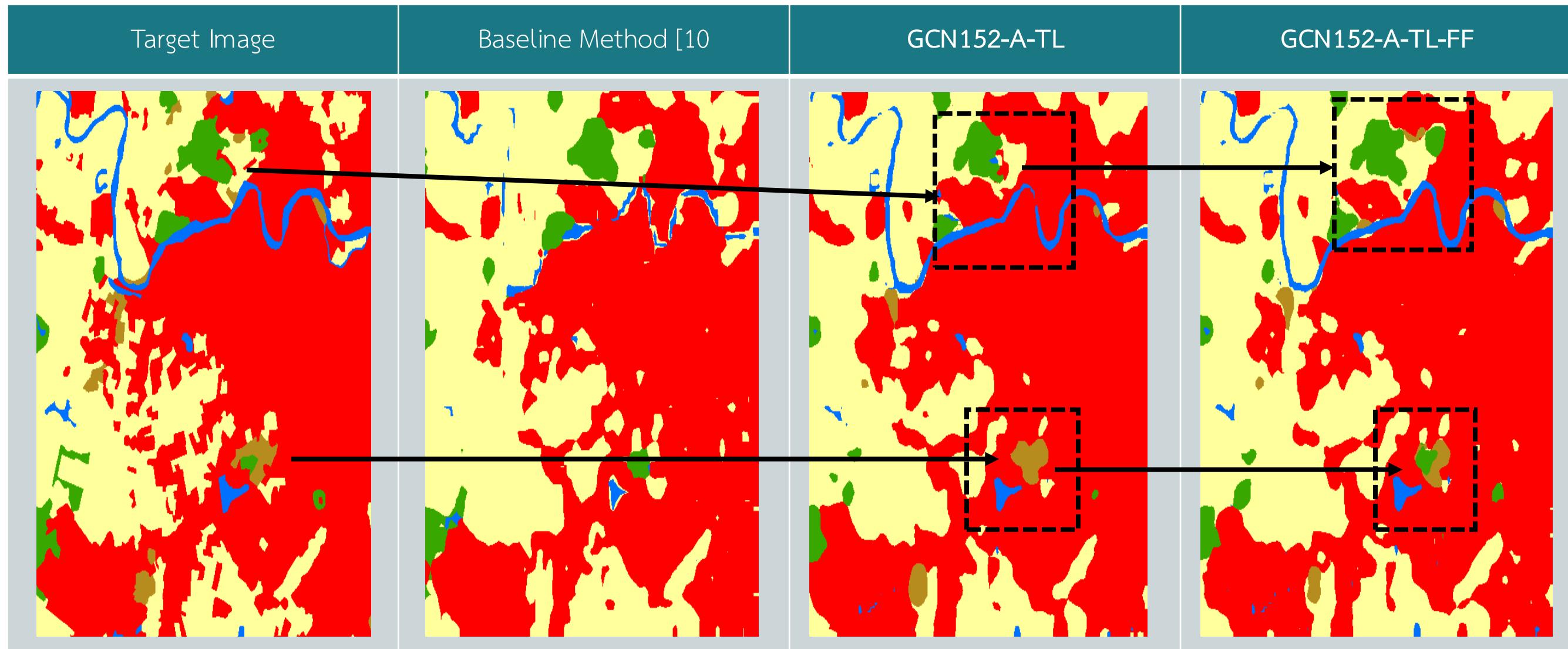
- Each class

- Our Proposed won DECD **5 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 4:  
Feature Fusion



# Evaluation

Corpus 1  
Nan, Thailand

Experiment 5:  
Depthwise Atrous Convolution

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.857	0.894	0.874
Proposed	TL	Res152	GCN-A-FF	0.930	0.924	0.927
	TL	Res152	GCN-A-FF-DA	0.934	0.939	0.936

- GCN152-A-TL-FF-DA overcame DECD ~ 6.221 % F1
- GCN152-A-TL-FF-DA overcame GCN152-A-TL-FF ~ 0.933 % F1

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 5:  
Depthwise Atrous Convolution

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.982	0.962	0.763	0.854	0.725
Proposed	GCN152-TL-A-FF	0.986	0.982	0.918	0.956	0.844
	GCN152-TL-A-FF-DA	0.989	0.957	0.934	0.949	0.868

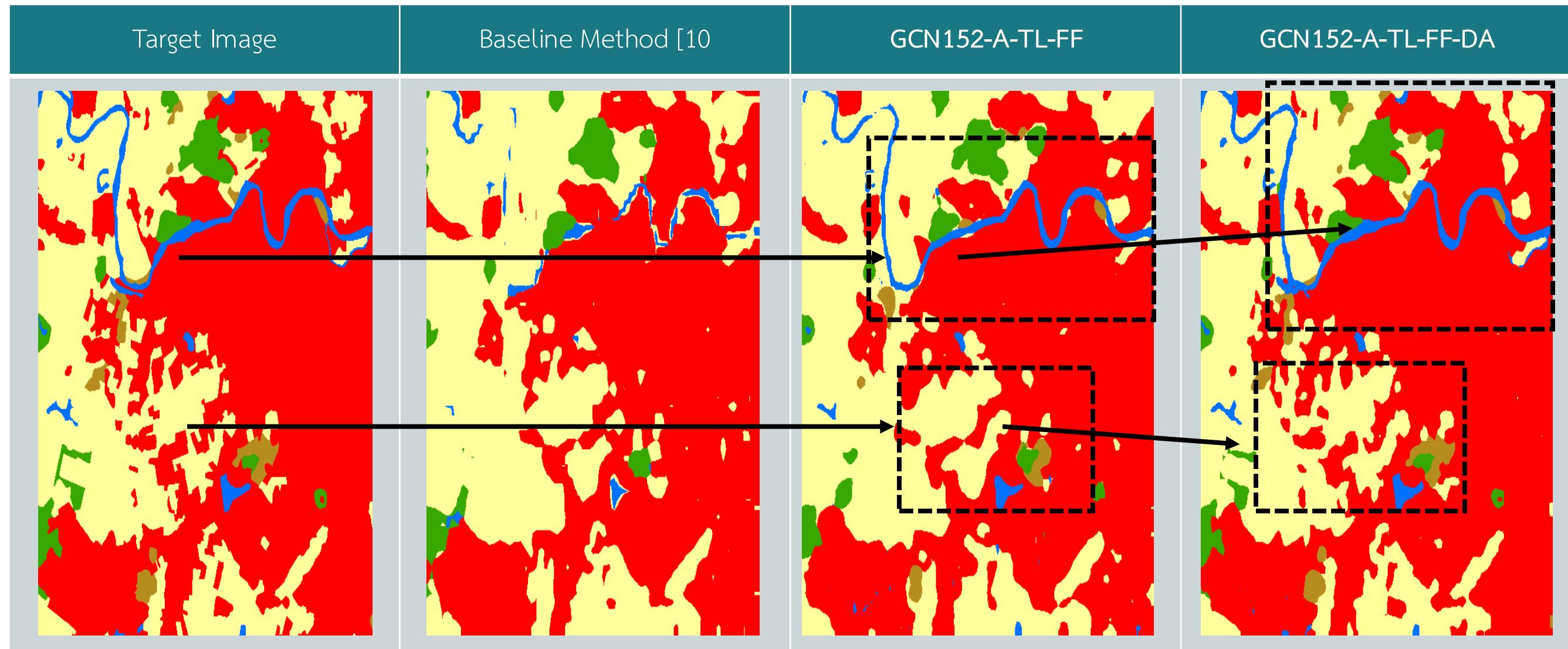
- Each class

- Our Proposed won DECD **5 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 5:  
Depthwise Atrous Convolution



# Evaluation

Corpus 1  
Nan, Thailand

Summary

Method	Model	F1 Score	Increase
Baseline	DCED	0.874	
P1	Enhanced GCN + Deeper Head Network	0.884	1.043 %
P2	+ Attention	0.917	3.288 %
P3	+ Transfer Learning	0.918	0.114 %
P4	+ Feature Fusion	0.927	0.843 %
P5	+ Depthwise Atrous Convolution	0.936	0.933%

The most impactful method:  
**Channel Attention**



2<sup>nd</sup> Corpus  
ISPRS Vaihingen

# Evaluation

- Precision
- Recall
- F1-score

	Corpus 1 Nan, Thailand		Corpus 2 ISPRS Vaihingen		Corpus 3 Isan, Thailand	
Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	-	Res50	GCN	0.872	0.852	0.858
	-	Res101	GCN	0.850	0.854	0.866
	-	Res152	GCN	0.873	0.864	0.868
	-	Res152	GCN-A	0.875	0.869	0.874
	TL	Res152	GCN-A	0.897	0.877	0.881
	TL	Res152	GCN-A-FF	0.896	0.904	0.905
	TL	Res152	GCN-A-FF-DA	0.923	0.900	0.911

Result: Our proposed method yields a higher F1 Score from baseline method at 5.7%

# Evaluation

Corpus 1  
Nan, Thailand

Corpus 2  
ISPRS Vaihingen

Corpus 3  
Isan, Thailand

- Each class

Method	Model	Imps	Building	Low veg	Tree	Car
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN50	0.876	0.873	0.857	0.953	0.803
	GCN101	0.941	0.913	0.742	0.904	0.699
	GCN152	0.810	0.963	0.895	0.912	0.806
	GCN152-A	0.886	0.928	0.811	0.895	0.820
	GCN152-TL-A	0.871	0.916	0.890	0.918	0.874
	GCN152-TL-A-FF	0.928	0.976	0.926	0.968	0.898
	GCN152-TL-A-FF-DA	0.907	0.979	0.927	0.972	0.910

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 1:  
How it impacts modern and over-deeper backbone?

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	-	Res50	GCN	0.872	0.852	0.858
	-	Res101	GCN	0.850	0.854	0.866
	-	Res152	GCN	0.873	0.864	0.868

- GCN50 overcame DECD ~ 0.386 % F1
- GCN152 overcame DECD ~ 1.366 % F1

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 1:  
How it impacts modern and over-deeper backbone?

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN50	0.876	0.873	0.857	0.953	0.803
	GCN101	0.941	0.913	0.742	0.904	0.699
	GCN152	0.810	0.963	0.895	0.912	0.806

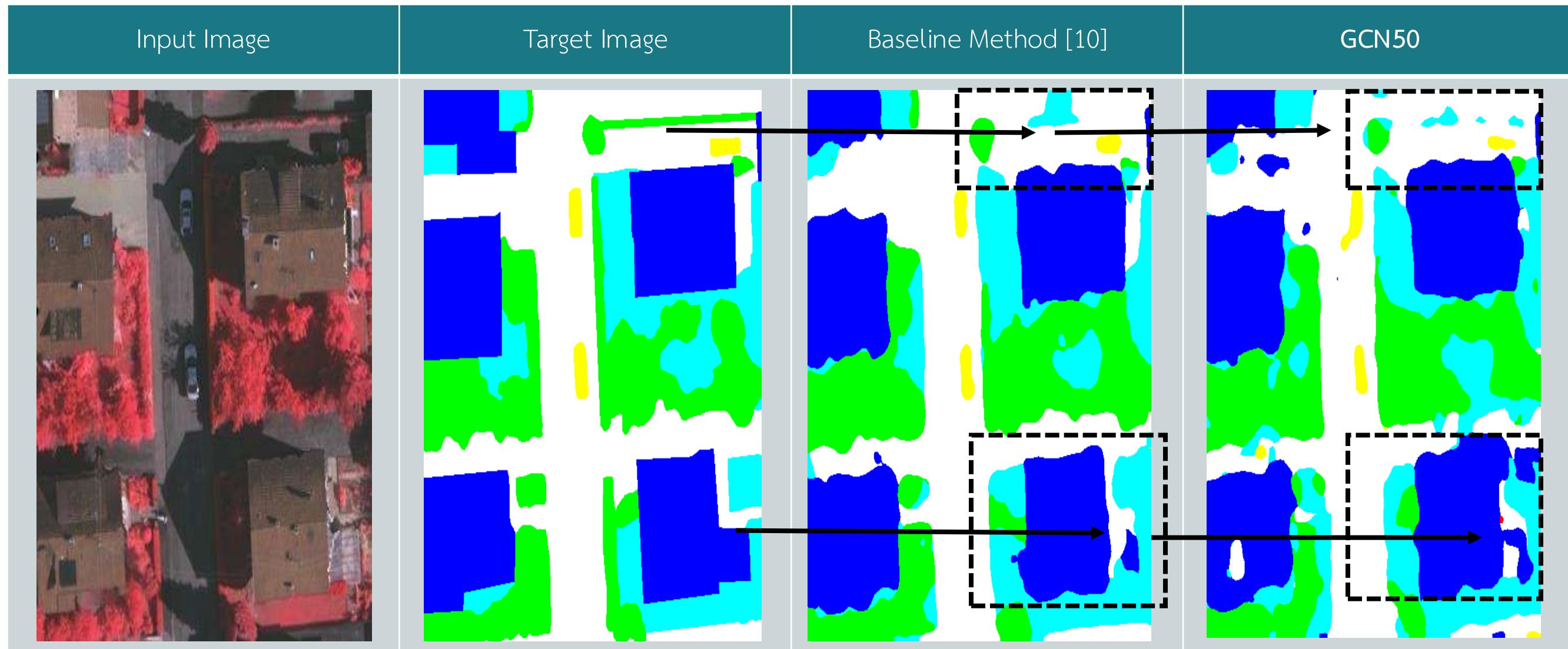
- Each class

- GCN Family won DECD **5 out of 5** classes

# Evaluation

Corpus 2  
ISPRS Vaihingen

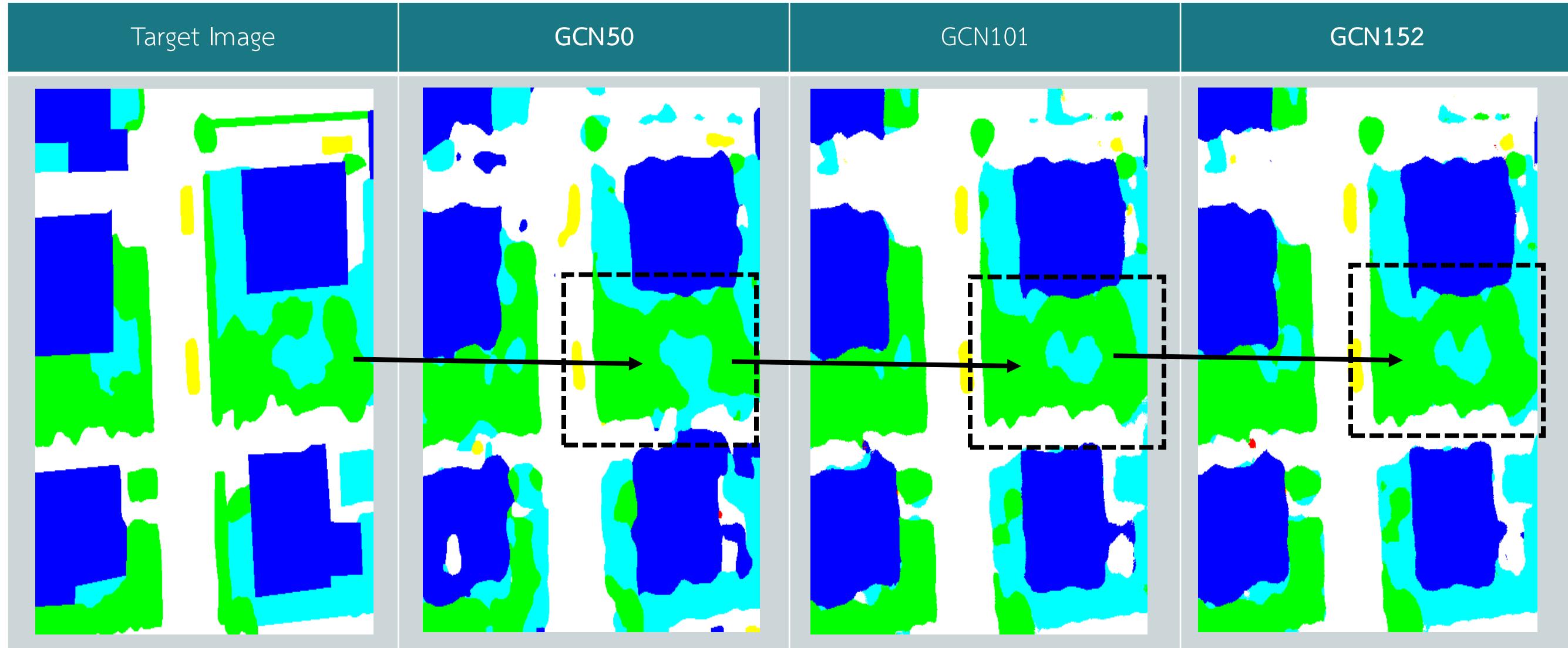
Experiment 1:  
How it impacts modern and over-deeper backbone?



# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 1:  
How it impacts modern and over-deeper backbone?



# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 2:  
Chanel Attention

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	-	Res152	GCN	0.873	0.864	0.868
	-	Res152	GCN-A	0.875	0.869	0.874

- GCN152-A overcame DECD ~ 1.916 % F1
- GCN152-A overcame GCN152 ~ 0.55 % F1

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 2:  
Chanel Attention

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN152	0.810	0.963	0.895	0.912	0.806
	GCN152-A	0.886	0.928	0.811	0.895	0.820

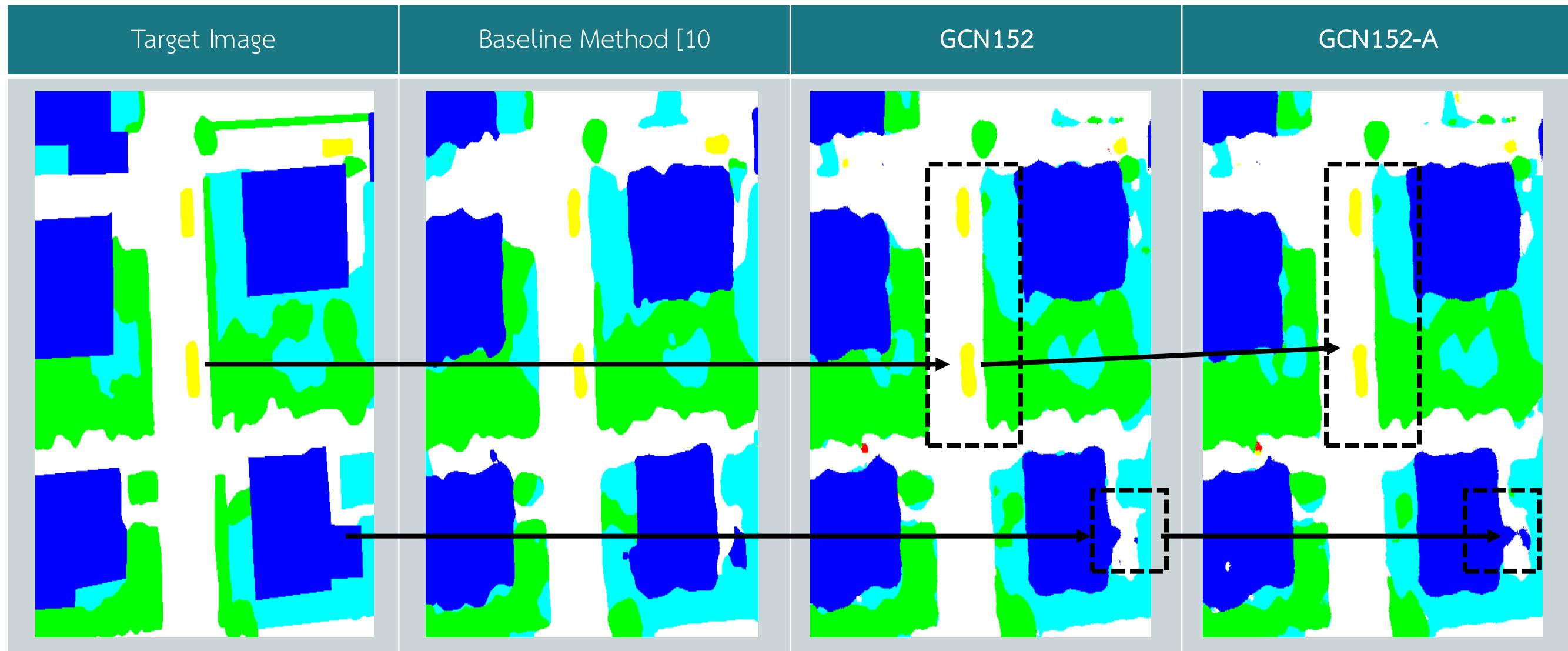
- Each class

- Our Proposed won DECD **4 out of 5** classes

# Evaluation

Corpus 1  
Nan, Thailand

Experiment 2:  
Chanel Attention



# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	-	Res152	GCN-A	0.875	0.869	0.874
	TL	Res152	GCN-A	0.897	0.877	0.881

- GCN152-A-TL overcame DECD ~ 2.642 % F1
- GCN152-A-TL overcame GCN152-A ~ 0.726 % F1

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN152-A	0.886	0.928	0.811	0.895	0.820
	GCN152-TL-A	0.871	0.916	0.890	0.918	0.874

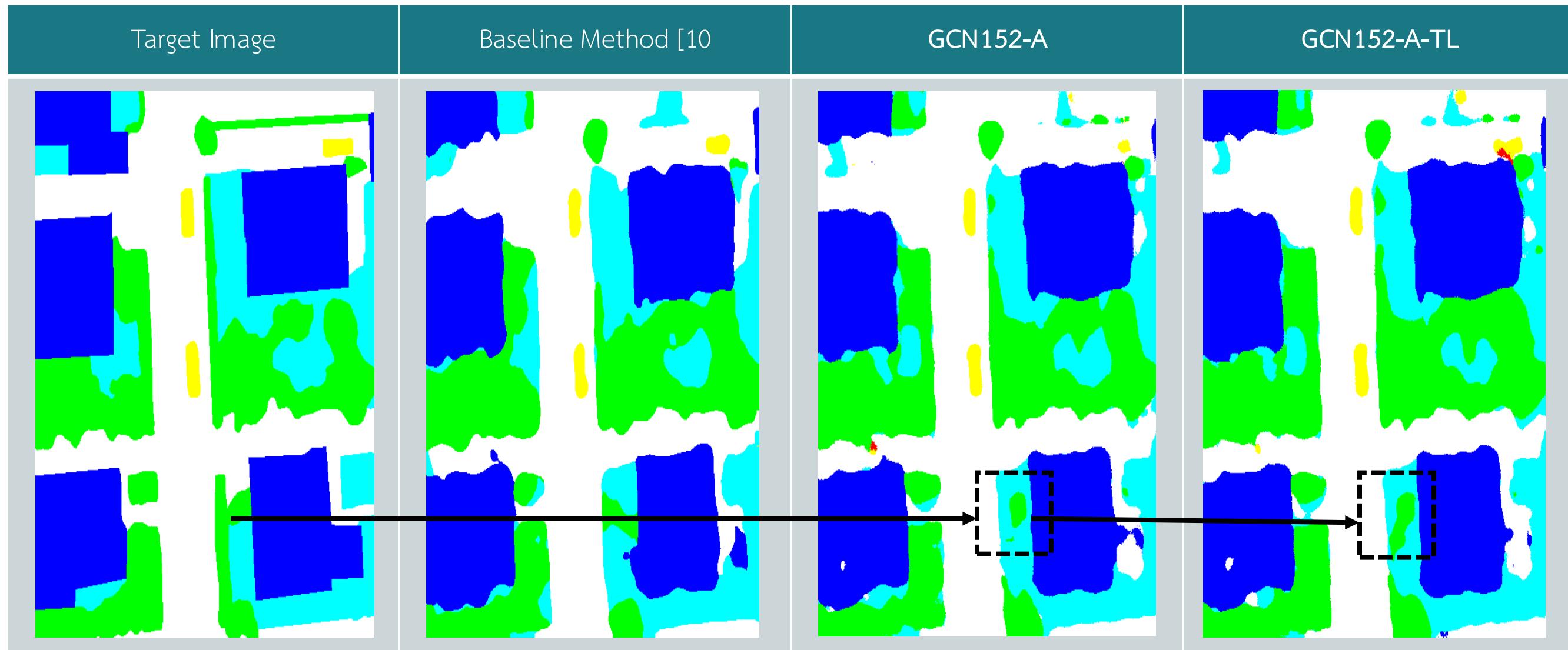
- Each class

- Our Proposed won DECD **5 out of 5** classes

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning



# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 4:  
Feature Fusion

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	TL	Res152	GCN-A	0.897	0.877	0.881
	TL	Res152	GCN-A-FF	0.896	0.904	0.905

- GCN152-A-TL-FF overcame DECD ~ 5.097 % F1
- GCN152-A-TL-FF overcame GCN152-A-TL ~ 2.455 % F1

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 4:  
Feature Fusion

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN152-TL-A	0.871	0.916	0.890	0.918	0.874
	GCN152-TL-A-FF	0.928	0.976	0.926	0.968	0.898

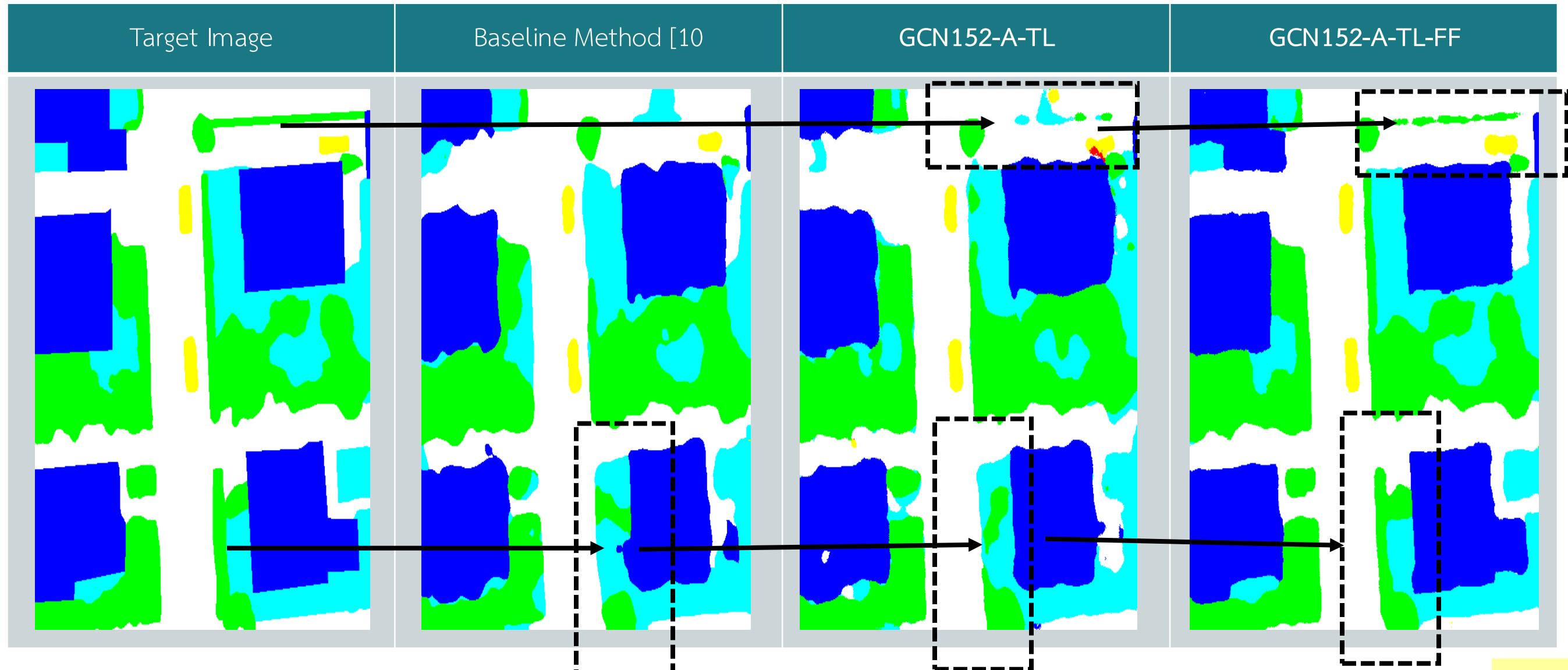
- Each class

- Our Proposed won DECD **5 out of 5** classes

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 4:  
Feature Fusion



# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 5:  
Depthwise Atrous Convolution

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.867	0.849	0.854
Proposed	TL	Res152	GCN-A-FF	0.896	0.904	0.905
	TL	Res152	GCN-A-FF-DA	0.923	0.900	0.911

- GCN152-A-TL-FF-DA overcame DECD ~ 5.67 % F1
- GCN152-A-TL-FF-DA overcame GCN152-A-TL-FF ~ 0.573 % F1

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 5:  
Depthwise Atrous Convolution

Method	Model	Agri	Forest	Misc	Urban	Water
Baseline	DCED	0.872	0.893	0.841	0.914	0.815
Proposed	GCN152-TL-A-FF	0.928	0.976	0.926	0.968	0.898
	GCN152-TL-A-FF-DA	0.907	0.979	0.927	0.972	0.910

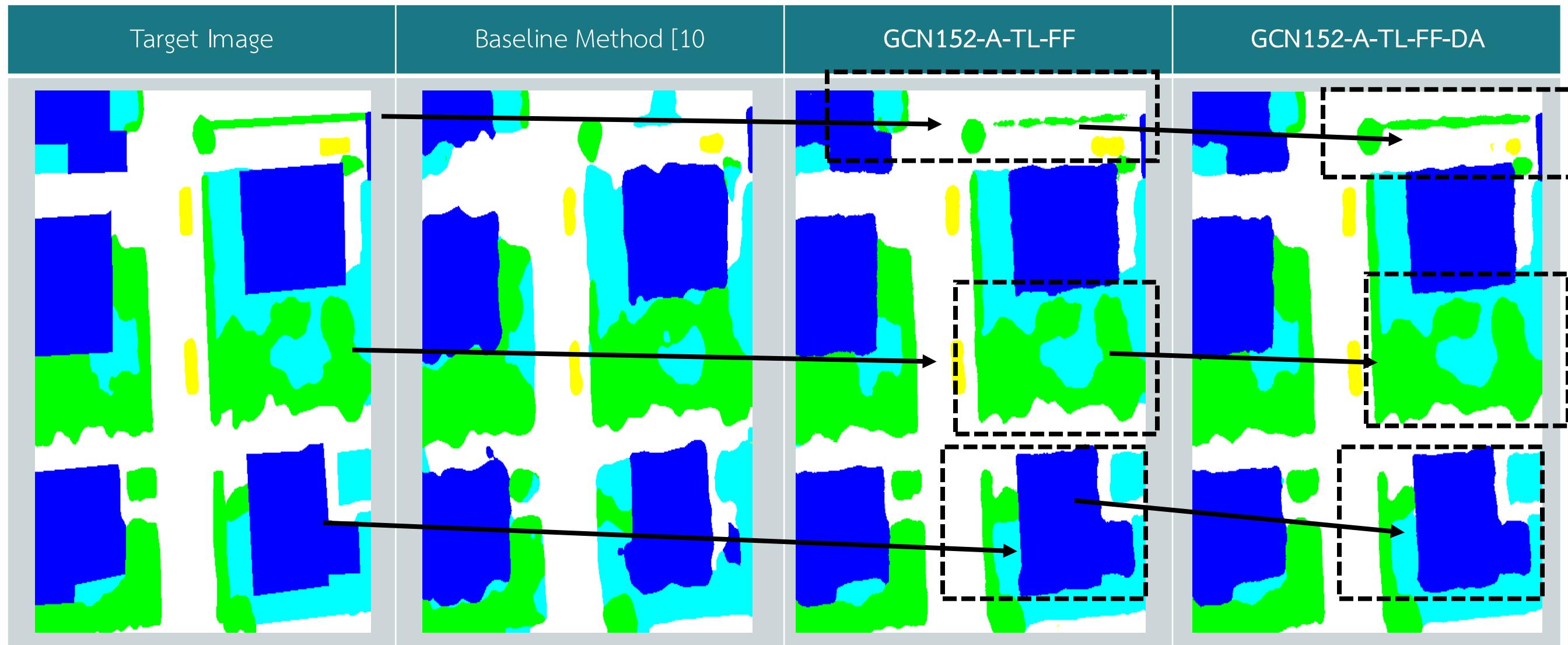
- Each class

- Our Proposed won DECD **5 out of 5** classes

# Evaluation

Corpus 2  
ISPRS Vaihingen

Experiment 5:  
Depthwise Atrous Convolution



# Evaluation

Corpus 2  
ISPRS Vaihingen

Summary

Method	Model	F1 Score	Increase
Baseline	DCED	0.854	
P1	Enhanced GCN + Deeper Head Network	0.868	1.366 %
P2	+ Attention	0.874	0.550 %
P3	+ Transfer Learning	0.881	0.726 %
P4	+ Feature Fusion	0.905	2.455 %
P5	+ Depthwise Atrous Convolution	0.911	0.573 %

The most impactful method:  
**Feature Fusion**

3<sup>rd</sup> Corpus  
Isan, Thailand (Medium Resolution Corpus)



# Evaluation

- Precision
- Recall
- F1-score

	Corpus 1 Nan, Thailand		Corpus 2 ISPRS Vaihingen		Corpus 3 Isan, Thailand	
Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	-	Res50	GCN	0.873	0.872	0.872
	-	Res101	GCN	0.865	0.884	0.874
	-	Res152	GCN	0.860	0.898	0.876
	-	Res152	GCN-A	0.865	0.891	0.877
	TL	Res152	GCN-A	0.890	0.923	0.899
	TL	Res152	GCN-A-FF	0.919	0.934	0.929
	TL	Res152	GCN-A-FF-DA	0.945	0.938	0.947

Result: Our proposed method yields a higher F1 Score from baseline method at 13.7%

# Evaluation

- Each class

Corpus 1 Nan, Thailand		Corpus 2 ISPRS Vaihingen		Corpus 3 Isan, Thailand
Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN50	0.933	0.778	0.888
	GCN101	0.837	0.815	0.862
	GCN152	0.910	0.721	0.879
	GCN152-A	0.858	0.768	0.854
	GCN152-TL-A	0.919	0.899	0.919
	GCN152-TL-A-FF	0.952	0.925	0.931
	GCN152-TL-A-FF-DA	0.969	0.948	0.938

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	-	Res50	GCN	0.873	0.872	0.872
	-	Res101	GCN	0.865	0.884	0.874
	-	Res152	GCN	0.860	0.898	0.876

- GCN50 overcame DECD ~ 6.145 % F1
- GCN152 overcame DECD ~ 6.601 % F1

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?

- Each class

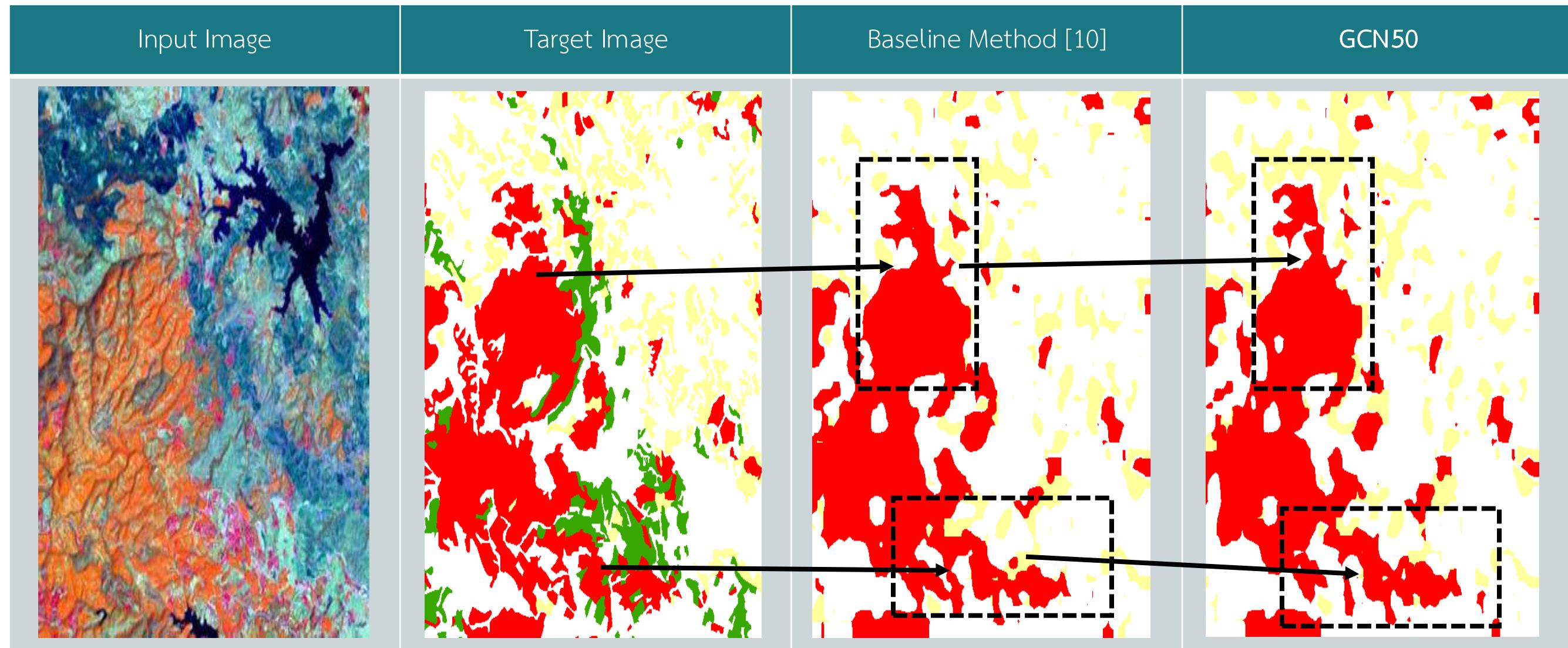
Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN50	0.933	0.778	0.888
	GCN101	0.837	0.815	0.862
	GCN152	0.910	0.721	0.879

- GCN Family won DECD **3 out of 3** classes

# Evaluation

Corpus 3  
Isan, Thailand

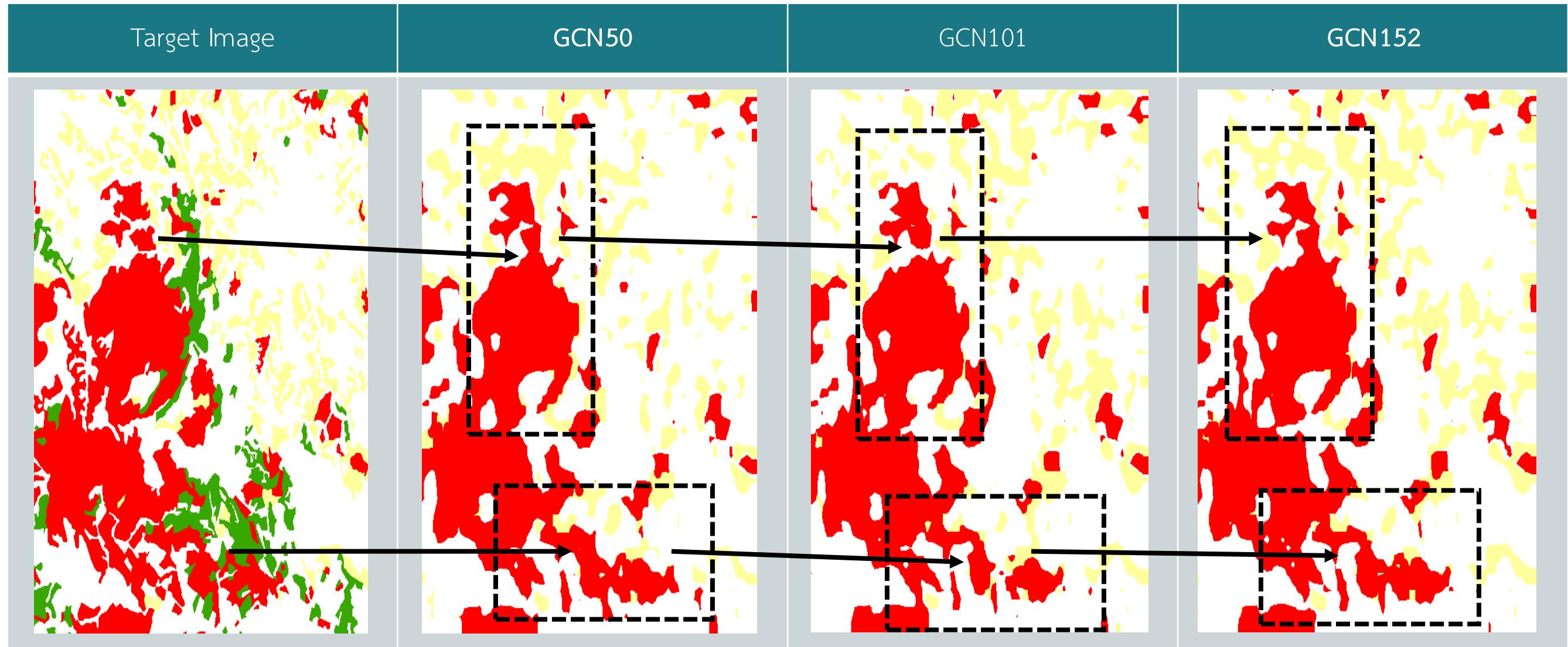
Experiment 1:  
How it impacts modern and over-deeper backbone?



# Evaluation

Corpus 3  
Isan, Thailand

Experiment 1:  
How it impacts modern and over-deeper backbone?



# Evaluation

Corpus 3  
Isan, Thailand

Experiment 2:  
Chanel Attention

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	-	Res152	GCN	0.860	0.898	0.876
	-	Res152	GCN-A	0.865	0.891	0.877

- GCN152-A overcame DECD ~ 6.681 % F1
- GCN152-A overcame GCN152 ~ 0.081 % F1

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 2:  
Chanel Attention

Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN152	0.910	0.721	0.879
	GCN152-A	0.858	0.768	0.854

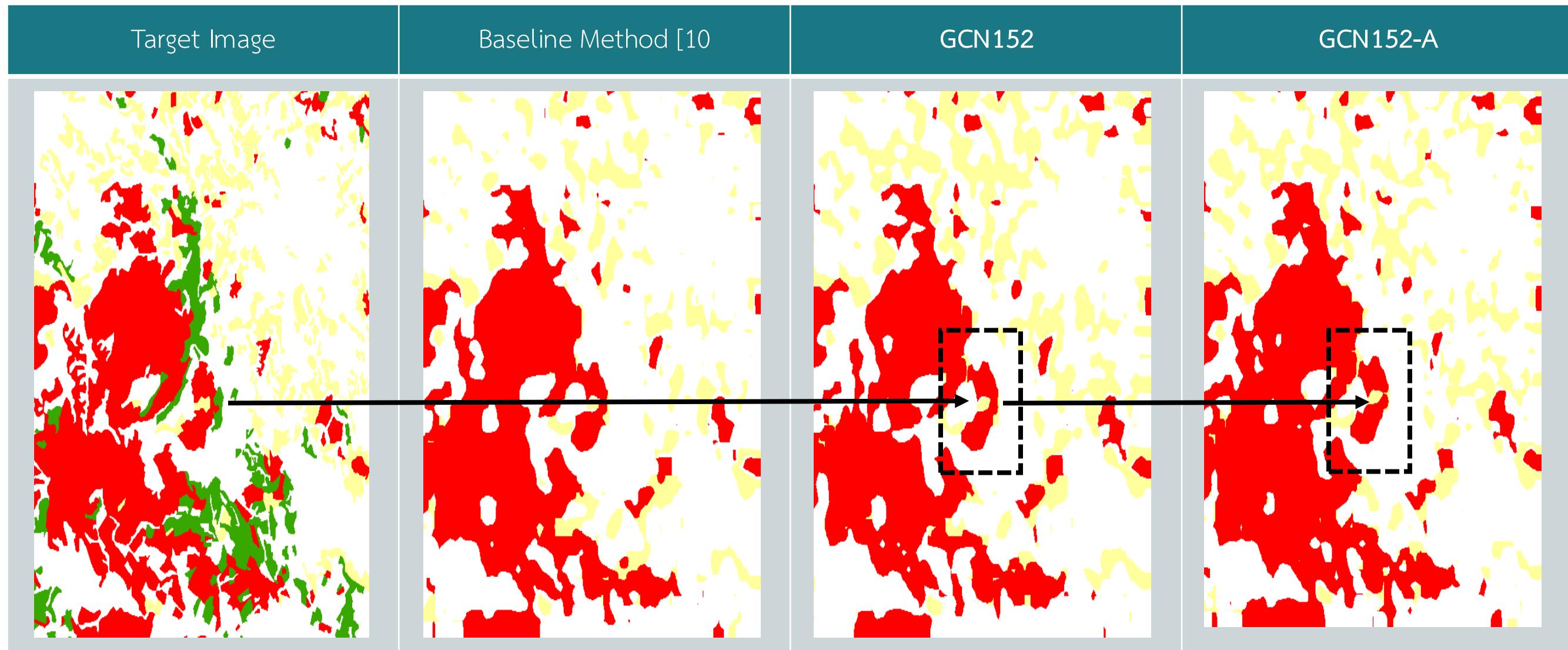
- Each class

- Our Proposed won DECD **2 out of 3** classes

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 2:  
Chanel Attention



# Evaluation

Corpus 3  
Isan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	-	Res152	GCN-A	0.865	0.891	0.877
	TL	Res152	GCN-A	0.890	0.923	0.899

- GCN152-A-TL overcame DECD ~ 8.875 % F1
- GCN152-A-TL overcame GCN152-A ~ 2.194 % F1

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning

- Each class

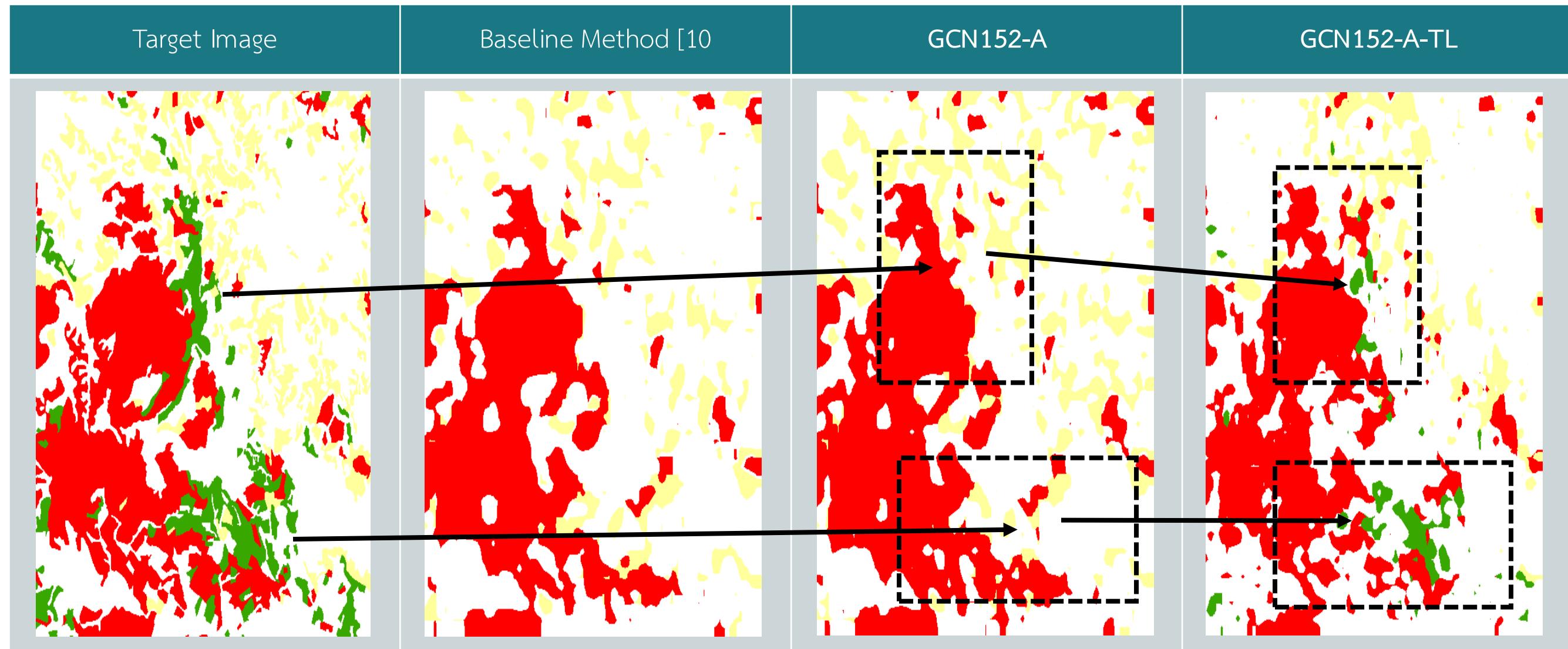
Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN152-A	0.858	0.768	0.854
	GCN152-TL-A	0.919	0.899	0.919

- Our Proposed won DECD 3 out of 3 classes

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 3:  
Deep CNNs with Domain Specific Transfer Learning



# Evaluation

Corpus 3  
Isan, Thailand

Experiment 4:  
Feature Fusion

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	TL	Res152	GCN-A	0.890	0.923	0.899
	TL	Res152	GCN-A-FF	0.919	0.934	0.929

- GCN152-A-TL-FF overcame DECD ~ 11.829 % F1
- GCN152-A-TL-FF overcame GCN152-A-TL ~ 2.954 % F1

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 4:  
Feature Fusion

Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN152-TL-A	0.919	0.899	0.919
	GCN152-TL-A-FF	0.952	0.925	0.931

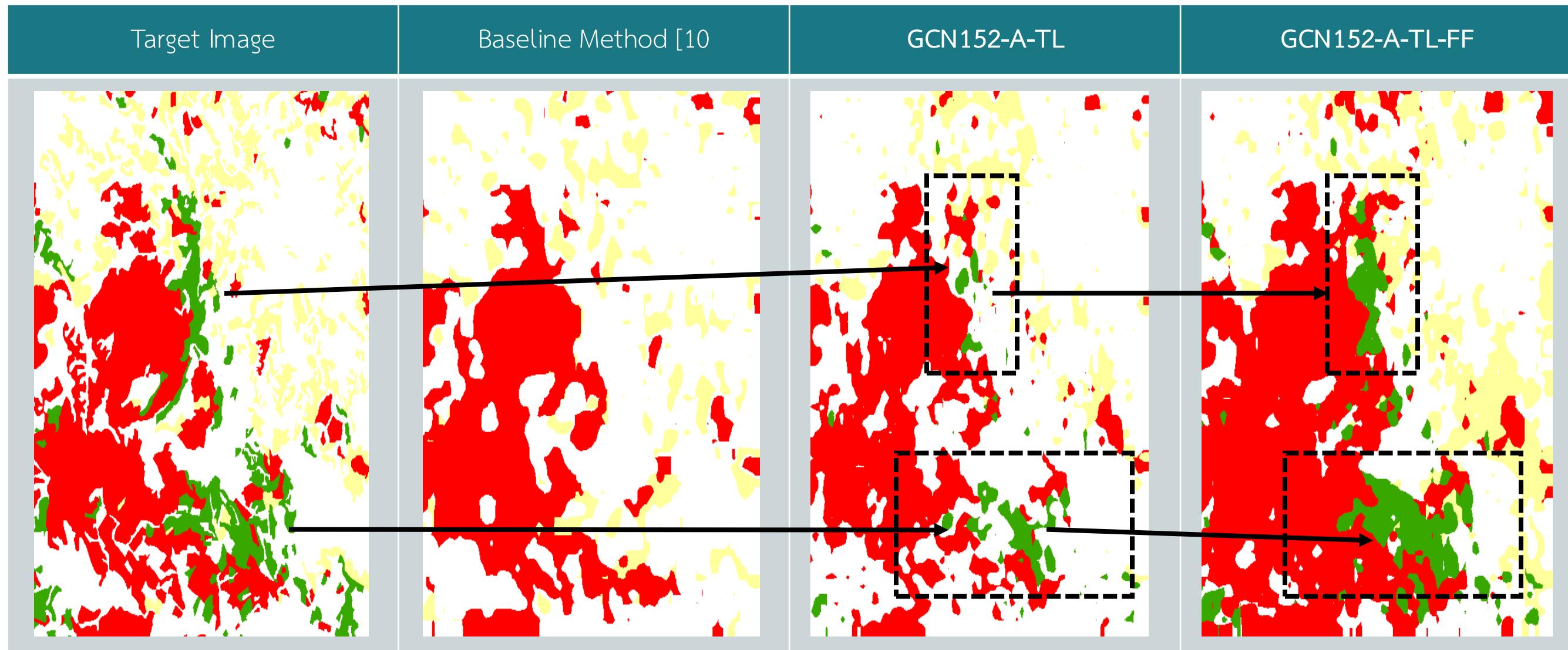
- Each class

- Our Proposed won DECD 3 out of 3 classes

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 4:  
Feature Fusion



# Evaluation

Corpus 3  
Isan, Thailand

Experiment 5:  
Depthwise Atrous Convolution

- Precision
- Recall
- F1-score

Method	Pretrained	Backbone	Model	Precision	Recall	F1 Score
Baseline	-	-	DCED	0.861	0.782	0.810
Proposed	TL	Res152	GCN-A-FF	0.919	0.934	0.929
	TL	Res152	GCN-A-FF-DA	0.945	0.938	0.947

- GCN152-A-TL-FF-DA overcame DECD ~ 13.701 % F1
- GCN152-A-TL-FF-DA overcame GCN152-A-TL-FF ~ 1.872 % F1

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 5:  
Depthwise Atrous Convolution

Method	Model	Corn	Pineapple	Pararubber
Baseline	DCED	0.905	0.815	0.820
Proposed	GCN152-TL-A-FF	0.952	0.925	0.931
	GCN152-TL-A-FF-DA	0.969	0.948	0.938

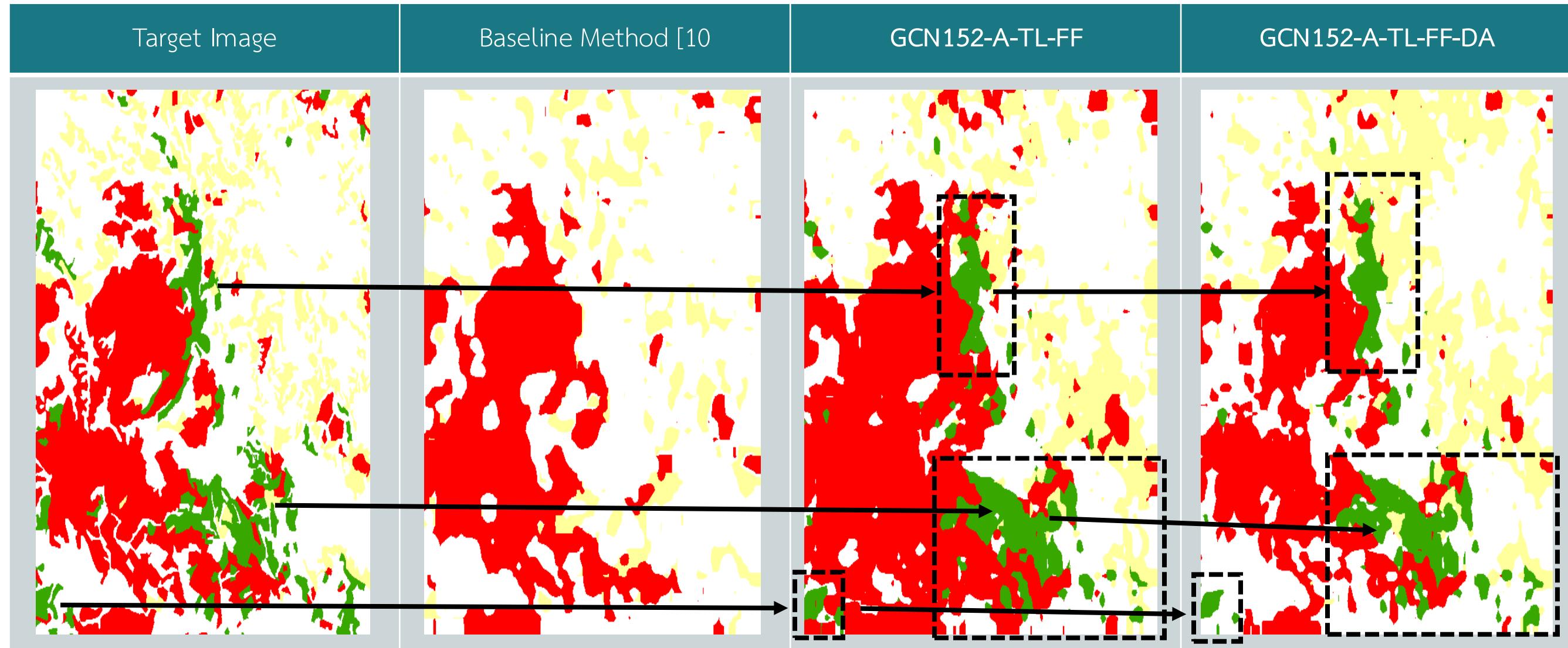
- Each class

- Our Proposed won DECD 3 out of 3 classes

# Evaluation

Corpus 3  
Isan, Thailand

Experiment 5:  
Depthwise Atrous Convolution



# Evaluation

Corpus 3  
Isan, Thailand

Summary

Method	Model	F1 Score	Increase
Baseline	DCED	0.810	
P1	Enhanced GCN + Deeper Head Network	0.876	6.601 %
P2	+ Attention	0.877	0.081 %
P3	+ Transfer Learning	0.899	2.194 %
P4	+ Feature Fusion	0.929	2.954 %
P5	+ Depthwise Atrous Convolution	0.947	1.872 %

The most impactful method:  
**Feature Fusion**  
And Transfer Learning  
(from Nan Corpus)

# Recap: The Results (Summary)

GCN = Global Conv Block

A = Channel Attention Block

TL = Transfer Learning

FF = Feature Fusion Block

DA = Depthwise Atrous

P1  
Backbone

P2  
Attention  
(A)

P3  
Transfer Learning  
(TL)

P4  
Feature Fusion  
(FF)

P5  
Depthwise Atrous  
(DA)

- **Corpus 1:** Nan Province (Medium Resolution Corpus)
  - GCN152-A-TL-FF-DA overcame DECD ~ **6.221 % F1**
  - The most impactful method: **Channel Attention**
- **Corpus 2:** ISPRS Vaihingen (Very-High Resolution Corpus)
  - GCN152-A-TL-FF-DA overcame DECD ~ **5.67 % F1**
  - The most impactful method: **Feature Fusion**
- **Corpus 3: Isan Region** (Medium Resolution Corpus)
  - GCN152-A-TL-FF-DA overcame DECD ~ **13.701 % F1**
  - The most impactful method: Feature Fusion and **Transfer Learning from Nan**

# Outline | Objective and Procedure

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- **Objectives and Procedure**
- Conclusions
- Publication and Reference

# Objective

## Objective of research

---

The objectives of this research are as follows:

1. To propose a **new deep learning architecture** to segment **multi-objects** from **aerial and satellite images (remote sensing corpora)**
2. To **explore the effectiveness** of proposing new deep learning techniques for semantic segmentation particularly on remote sensing corpora

# Objective

## Scope of research

---

1. Evaluate the proposed new deep learning on **ISPRS Vaihingen corpus** (a city district of Stuttgart, Germany) and **GISTDA corpora** (GISTDA Nan province and Isan zone corpora) with Encoder-Decoder baseline model
  - Nan province corpora have **five classes**: agriculture, forest, miscellaneous, urban, and water
  - Isan zone corpora have **three classes**: corn, pineapple, and rubber tree
2. Evaluate the proposed deep learning on reliable measurements such as **Precision, Recall, and F1-score**

# Procedure

## Procedure

Research Planning	S1/2017	S2/2017	S1/2018	S2/2018	S1/2019	S2/2019
1. Research modern deep learning techniques						
2. Research deep learning on remote sensing images						
3. Literature review						
4. Request and collect data sets from ISPRS corpus and private corpus (GISTDA)						
5. Design and implement the proposed and baseline deep learning.						
6. Conclude and prepare for 1 <sup>st</sup> ISI journal						
7. Write and thesis proposal examination						
8. Evaluate and improve my new deep learning architecture						
9. Conclude and prepare for 2 <sup>nd</sup> ISI journal						
10. Write and defend the dissertation						

# Outline | Conclusions

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- **Conclusions**
- Publication and Reference

# Conclusions

Title: Semantic Segmentation on Remotely Sensed Images Using Deep Convolutional Encoder-Decoder Neural Network

---

- **What:** Semantic Segmentation on Remotely Sensed Corpora
- **Why:** The previous methods suffer from accuracy performance
- **How:** Deep Convolutional Encoder-Decoder Neural Network
- **Proposed Methods (What's New):**
  - (1) Varying Backbones (2) Channel Attention (3) Domain-specific Transfer Learning (4) Feature Fusion (5) Depthwise Atrous Conv
- **Result:**
  - The results demonstrate that the “GCN152-TL-A-FF-DA” model significantly exceeds all baselines. It is the victor in all data sets and exceeds more than 90% of F1: 0.9114, 0.9362, and 0.9111 of the Landsat-8w3c, Landsat-8w5c, and ISPRS Vaihingen.
  - Moreover, it reaches an accuracy surpassing 90% in almost all classes.
- **Future Plan:**
  - Efficient Uncertainty Estimation for Semantic Segmentation (Aleatoric and Epistemic) | Explainable AI

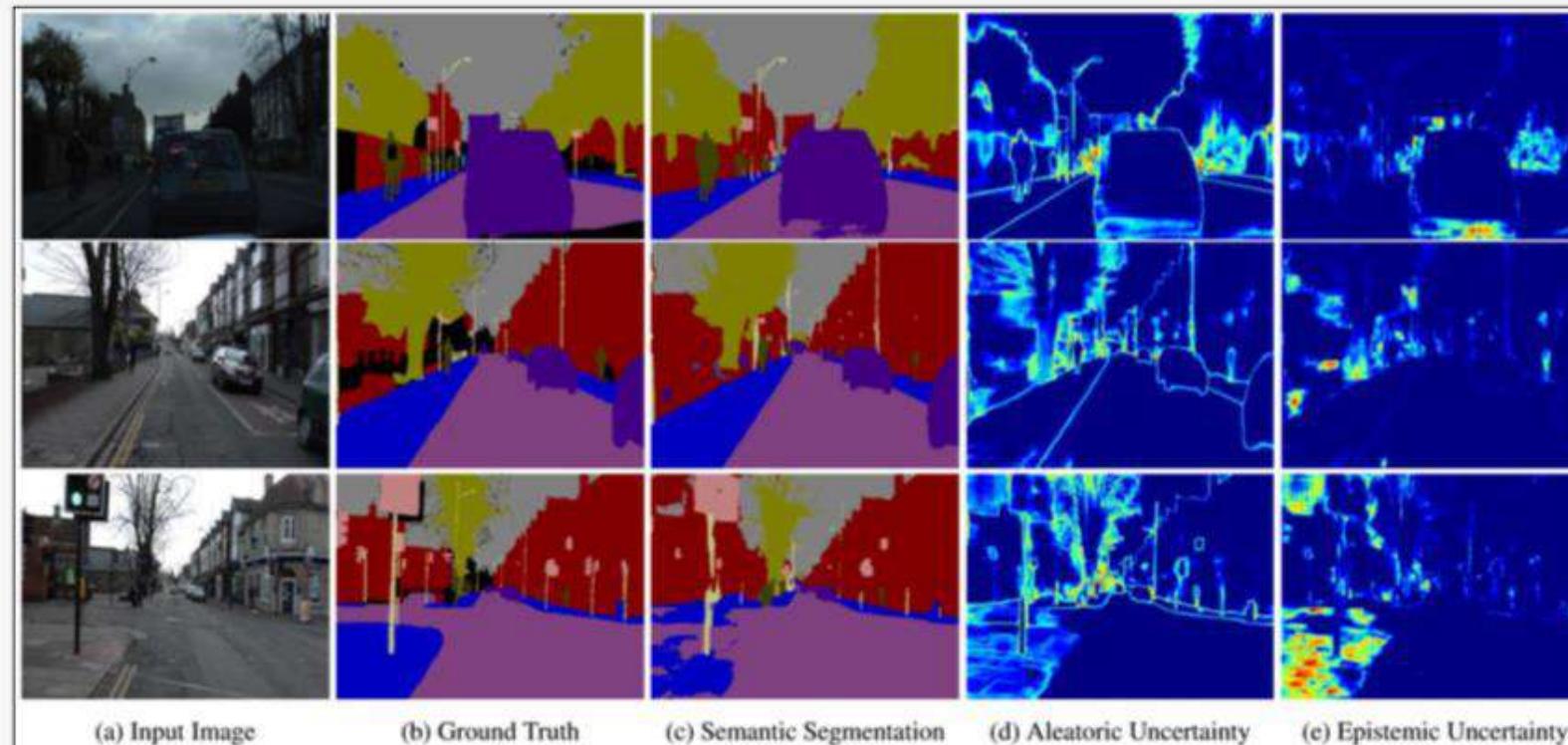
# Efficient Uncertainty Estimation for Semantic Segmentation (Aleatoric and Epistemic)

**Future Plan:** Huang, Po-Yu, et al. "Efficient uncertainty estimation for semantic segmentation in videos." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

What kind of uncertainty can we model?

*Epistemic* uncertainty is *modeling* uncertainty

*Aleatoric* uncertainty is *sensing* uncertainty



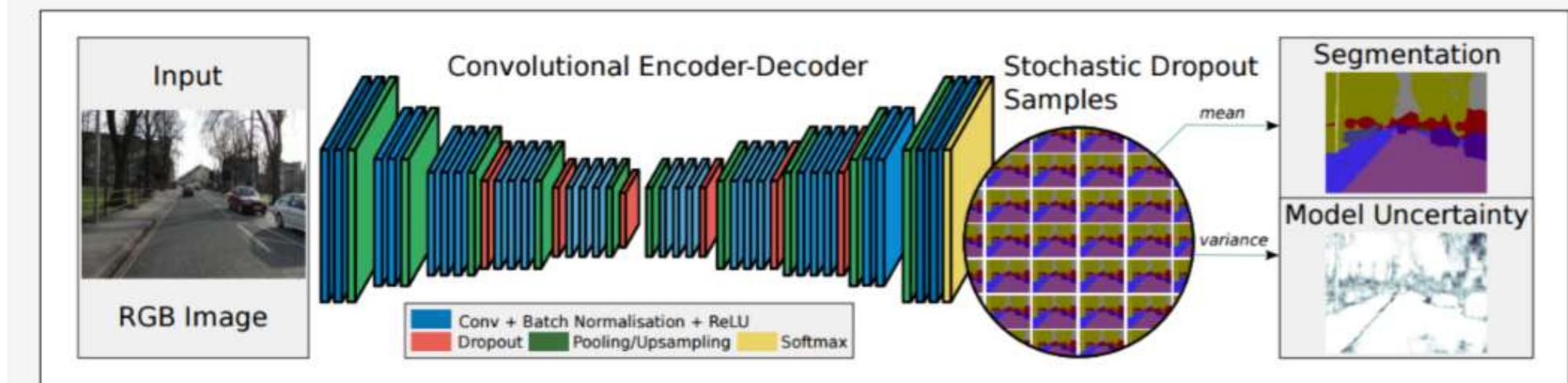
# Efficient Uncertainty Estimation for Semantic Segmentation (Aleatoric and Epistemic)

**Future Plan:** Huang, Po-Yu, et al. "Efficient uncertainty estimation for semantic segmentation in videos." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

## Modeling Epistemic Uncertainty with Bayesian Deep Learning

We can **model epistemic uncertainty** in deep learning models using Monte Carlo **dropout sampling** at test time.

Dropout sampling can be interpreted as **sampling from a distribution over models**.



Alex Kendall, Vijay Badrinarayanan and Roberto Cipolla **Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding**. arXiv preprint arXiv:1511.02680, 2015.

# Efficient Uncertainty Estimation for Semantic Segmentation (Aleatoric and Epistemic)

**Future Plan:** Huang, Po-Yu, et al. "Efficient uncertainty estimation for semantic segmentation in videos." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

Modeling Aleatoric Uncertainty with Probabilistic Deep Learning		
	Deep Learning	Probabilistic Deep Learning
Model	$[\hat{y}] = f(x)$	$[\hat{y}, \hat{\sigma}^2] = f(x)$
Regression	$Loss = \ y - \hat{y}\ ^2$	$Loss = \frac{\ y - \hat{y}\ ^2}{2\hat{\sigma}^2} + \log \hat{\sigma}^2$
Classification	$Loss = SoftmaxCrossEntropy(\hat{y}_t)$	$\hat{y}_t = \hat{y} + \epsilon_t \quad \epsilon_t \sim N(0, \hat{\sigma}^2)$ $Loss = \frac{1}{T} \sum_t SoftmaxCrossEntropy(\hat{y}_t)$

# Outline | Conclusions

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- **Publication** and Reference

# 1<sup>st</sup> Publication (Q1-Tier1, ISI Journal, 2019)

Title: Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning, 2019

- Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning. *Remote Sens.* **2019**, *11*, 83.

The screenshot shows the MDPI Remote Sensing journal website. The article title is "Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning". The authors listed are Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peerapon Vateekul. The article was received on December 5, 2018, revised on December 25, 2018, accepted on January 1, 2019, and published on January 4, 2019. It has 1212 views, 860 downloads, and 5 citations. The abstract discusses the challenges of semantic segmentation in remote sensing and introduces an enhanced global convolutional network with channel attention and domain-specific transfer learning. The journal's impact factor is 4.118.

The screenshot shows the MDPI Remote Sensing journal website. The journal logo features a satellite icon and the text "remote sensing". To the right is a large yellow circular badge with the text "IMPACT FACTOR 4.118". The URL https://www.mdpi.com/2072-4292/11/1/83 is highlighted in a red box.

• <https://www.mdpi.com/2072-4292/11/1/83>

Remote Sens. 2019, 11(1), 83; <https://doi.org/10.3390/rs11010083>

Received: 5 December 2018 / Revised: 25 December 2018 / Accepted: 1 January 2019 / Published: 4 January 2019

# 2<sup>nd</sup> Publication (Q1-Tier1, ISI Journal, 2020)

Title: Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution, 2020

- Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution. *Remote Sens.* **2020**, *12*, 1233.

The screenshot shows the MDPI website interface. At the top, there's a navigation bar with links for Safari, File, Edit, View, History, Bookmarks, Window, Help, and a search bar. Below the navigation is the MDPI logo and a search bar labeled "Search for Articles: Title / Keyword". The main content area displays the article title: "Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution". The article is categorized under "Open Access" and "Article". It was written by Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peempon Vateekul. The article is associated with the journal "remote sensing" and the volume "Volume 12 / Issue 8". The URL is https://doi.org/10.3390/rs12081233. The page also includes sections for "Article Overview", "Article Versions", "Export Article", and "Related Info Links". At the bottom, there are buttons for "View Full-Text", "Download PDF", "Browse Figures", "Review Reports", and "Cite This Paper".

The screenshot shows the journal cover for "remote sensing". The cover features a yellow circular badge with the text "IMPACT FACTOR 4.118". The journal title "remote sensing" is displayed in large green letters next to a small icon of a satellite and a globe. The background of the cover is white. To the left of the cover, there's a link to the journal's homepage: "Journals / Remote Sensing / Volume 12 / Issue 8".

- <https://www.mdpi.com/2072-4292/12/8/1233>

Remote Sens. 2020, 12(8), 1233; <https://doi.org/10.3390/rs12081233>

# Outline | Publication and Reference

- Introduction
- Related Theory
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and **Reference**

# Reference (based-modern deep learning)

- [1] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890).
- [2] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 11-19).
- [3] Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2018). Large Kernel Matters--Improve Semantic Segmentation by Global Convolutional Network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4353-4361).
- [4] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).
- [5] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 325-341).

# Reference (based-deep learning on Remote Sensing)

- [6] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
- [7] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481-2495.
- [8] Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE international conference on computer vision (pp. 1520-1528).
- [9] Wang, H., Wang, Y., Zhang, Q., Xiang, S., & Pan, C. (2017). Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sensing*, 9(5), 446.
- [10] Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., & Pan, C. (2018). Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145, 78-95.
- etc.



# Q&A



Chula  
Chulalongkorn University

---

Teerapong Panboonyuen (Kao)

ธีรพงศ์ ปานบุญยืน (เก้า)

<https://kaopanboonyuen.github.io/>



# Appendix

# 1<sup>st</sup> Publication (Q1-Tier1, ISI Journal, 2019)

Title: Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning, 2019

- Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning. *Remote Sens.* **2019**, *11*, 83.

The screenshot shows the MDPI Remote Sensing journal website. The article title is "Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning". The authors listed are Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peerapon Vateekul. The article was received on December 5, 2018, revised on December 25, 2018, accepted on January 1, 2019, and published on January 4, 2019. It has 1212 views, 860 downloads, and 5 citations. The abstract discusses the challenges of semantic segmentation in remote sensing and introduces an enhanced global convolutional network with channel attention and domain-specific transfer learning. The journal's impact factor is 4.118.

The screenshot shows the MDPI Remote Sensing journal website. The journal logo features a satellite icon and the word "remote sensing". To the right is a large yellow circular badge with the text "IMPACT FACTOR 4.118". The URL https://www.mdpi.com/2072-4292/11/1/83 is highlighted in a red box.

• <https://www.mdpi.com/2072-4292/11/1/83>

Remote Sens. 2019, 11(1), 83; <https://doi.org/10.3390/rs11010083>

Received: 5 December 2018 / Revised: 25 December 2018 / Accepted: 1 January 2019 / Published: 4 January 2019

# 2<sup>nd</sup> Publication (Q1-Tier1, ISI Journal, 2020)

Title: Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution, 2020

- Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution. *Remote Sens.* **2020**, *12*, 1233.

The screenshot shows the MDPI website interface. At the top, there's a navigation bar with links for Safari, File, Edit, View, History, Bookmarks, Window, Help, and a search bar. Below the navigation is the MDPI logo and a main menu with links for Journals, Information, Author Services, Initiatives, About, Sign In / Sign Up, and Submit. The main content area displays the article details. The title of the article is "Semantic Labeling in Remote Sensing Corpora Using Feature Fusion-Based Enhanced Global Convolutional Network with High-Resolution Representations and Depthwise Atrous Convolution". It was published in the journal "remote sensing" (Volume 12, Issue 8). The authors listed are Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peempon Vateekul. The article is marked as Open Access and Article. On the left side, there's an "Article Menu" with sections for Article Overview, Article Versions, Export Article, and Related Info Links. At the bottom, there are buttons for View Full-Text, Download PDF, Browse Figures, Review Reports, and Cite This Paper.

The screenshot shows the journal logo for "remote sensing" and a large yellow circular badge with the text "IMPACT FACTOR 4.509". Below the journal logo, there's a link to the article: "https://www.mdpi.com/2072-4292/12/8/1233".

Remote Sens. 2020, 12(8), 1233; <https://doi.org/10.3390/rs12081233>

# Detail of All Corpora (1) Support Value

**ISPRS**

		Support	
Class	0	778361	Impervious Surfaces
Class	1	920658	Buildings
Class	2	332791	Low Vegetation
Class	3	393875	Tree
Class	4	32939	Car

**NAN**

		Support	
Class	0	443917	Agriculture
Class	1	1022641	Forest
Class	2	20039	Miscellaneous
Class	3	61773	Urban
Class	4	7086	Water

**ISAN**

LC129048

		Support	
Class	0	333917	Miscellaneous
Class	1	892808	Para Rubber
Class	2	32180	Pine Apple
Class	3	523588	Corn

# Detail of All Corpora (2) Training Size

## Public Data Set: 2D Semantic Labeling - Vaihingen

- Training Set: 512x512 (210 Images)
- Validation Set: 512x512 (30 Images)
- Testing Set: 512x512 (30 Images)

## Private Data Set: GISTDA Nan Province Corpus

- Training Set: 512x512 (1,770 Images)
- Validation Set: 512x512 (49 Images)
- Testing Set: 512x512 (100 Images)

## Private Data Set: GISTDA ISAN zone Corpus

- Training Set: 512x512 (2,115 Images)
- Validation Set: 512x512 (49 Images)
- Testing Set: 512x512 (100 Images)

NAN		Support	
Class	0	443917	Agriculture
Class	1	1022641	Forest
Class	2	20039	Miscellaneous
Class	3	61773	Urban
Class	4	7086	Water

ISAN	LC129048	Support	
Class	0	333917	Miscellaneous
Class	1	892808	Para Rubber
Class	2	32180	Pine Apple
Class	3	523588	Corn

# ResNet - Architecture

**ResNet-50**  
PRETRAINED MODEL

P1

Satellite



ResNet 50

ResNet 101

ResNet 152

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

# Problem Solve: Unbalanced Class



Zerphed commented on May 30, 2017

+ 😊 ...

@JeffKo427 Thanks! This is in fact what I am using right now. The losses seem quite bit, but I guess that was to be expected:

```
def weighted_pixelwise_crossentropy(class_weights):

    def loss(y_true, y_pred):
        epsilon = _to_tensor(_EPSILON, y_pred.dtype.base_dtype)
        y_pred = tf.clip_by_value(y_pred, epsilon, 1. - epsilon)
        return - tf.reduce_sum(tf.multiply(y_true * tf.log(y_pred), class_weights))

    return loss
```



5

# How does mean image subtraction work?

2 Answers

active    oldest

votes

In deep learning, there are in fact different practices as to how to subtract the mean image.

7

## Subtract mean image

The first way is to subtract mean image as @lejlot described. But there is an issue if your dataset images are not the same size. You need to make sure all dataset images are in the same size before using this method (e.g., resize original image and crop patch of same size from original image). It is used in original ResNet paper, see [reference here](#).



## Subtract the per-channel mean

The second way is to subtract per-channel mean from the original image, which is more popular. In this way, you do not need to resize or crop the original image. You can just calculate the per-channel mean from the training set. This is used widely in deep learning, e.g, Caffe: [here](#) and [here](#). Keras: [here](#). PyTorch: [here](#). (PyTorch also divide the per-channel value by standard deviation.)

[share](#) [improve this answer](#)

edited Dec 6 '17 at 1:18

answered Dec 5 '17 at 9:49



jdhao

5,760 • 2 • 36 • 60

[add a comment](#)

# Other Layers

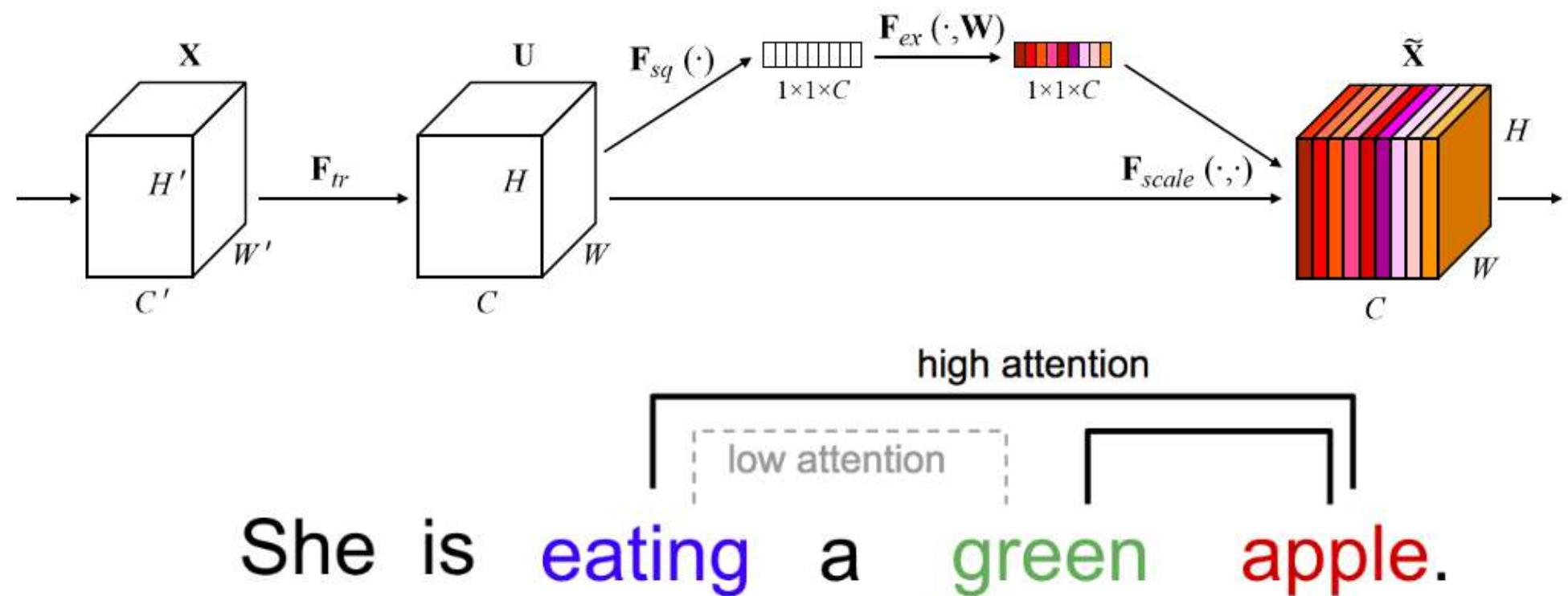
- **Interpolation Layer:** Interpolation layer
  - performs resizing operation along the spatial dimension.
  - In our network, we use bilinear interpolation.
- **Elementwise Layer:** Elementwise layer
  - performs elementwise operations on two or more previous layers, in which the feature maps must be of the same number of channels and the same size.
  - There are three kinds of elementwise operations:
    - product, sum, max.
    - In our network, we use sum operation.

# Related Theory

Attention

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- **(4) Channel Attention**
- (5) Feature Fusion
- (6) Design CNNs
- (7) Depthwise Atrous

- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features



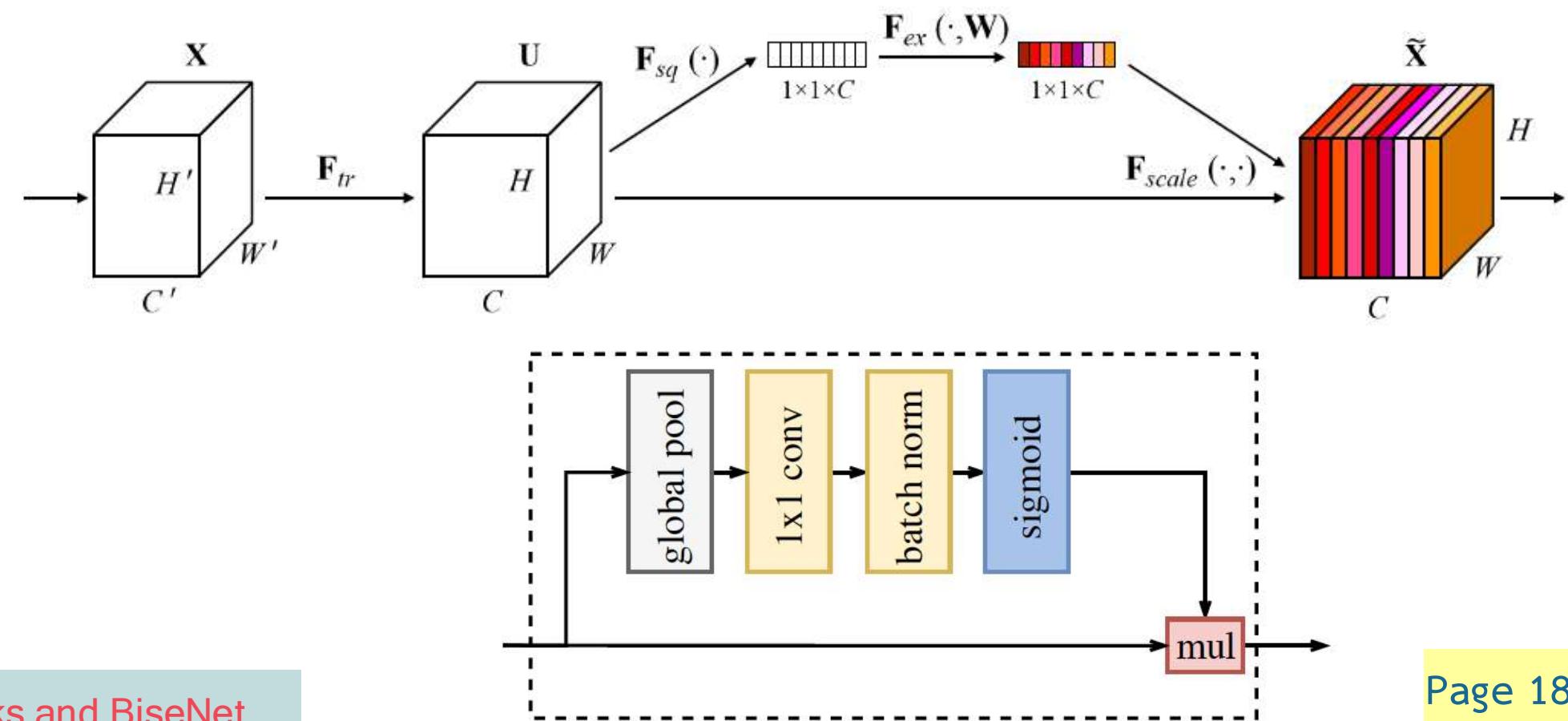
One word “attends” to other words in the same sentence differently.

# Related Theory

Attention

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- **(4) Channel Attention**
- (5) Feature Fusion
- (6) Design CNNs
- (7) Depthwise Atrous

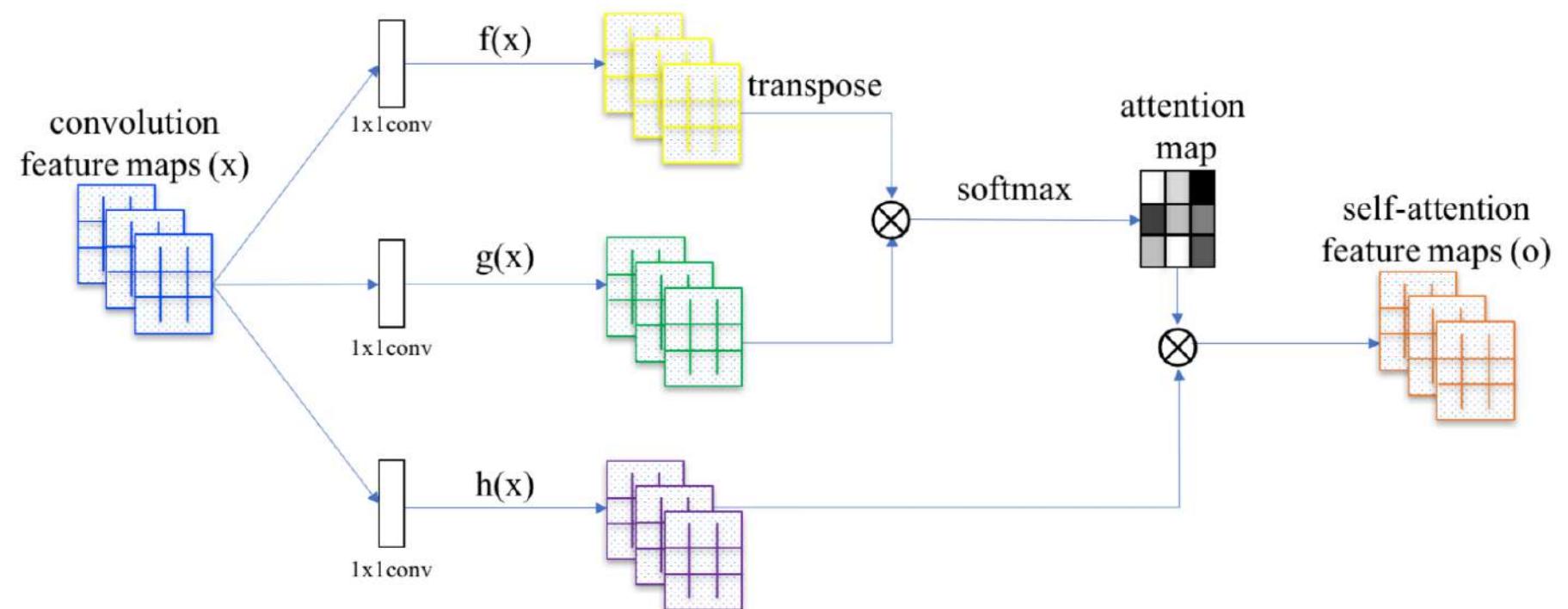
- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features



# Related Theory

Attention

- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - **(4) Channel Attention**
  - (5) Feature Fusion
  - (6) Design CNNs
  - (7) Depthwise Atrous
- Attention is helpful to focus on what we want
  - We utilize channel attention to select the important features



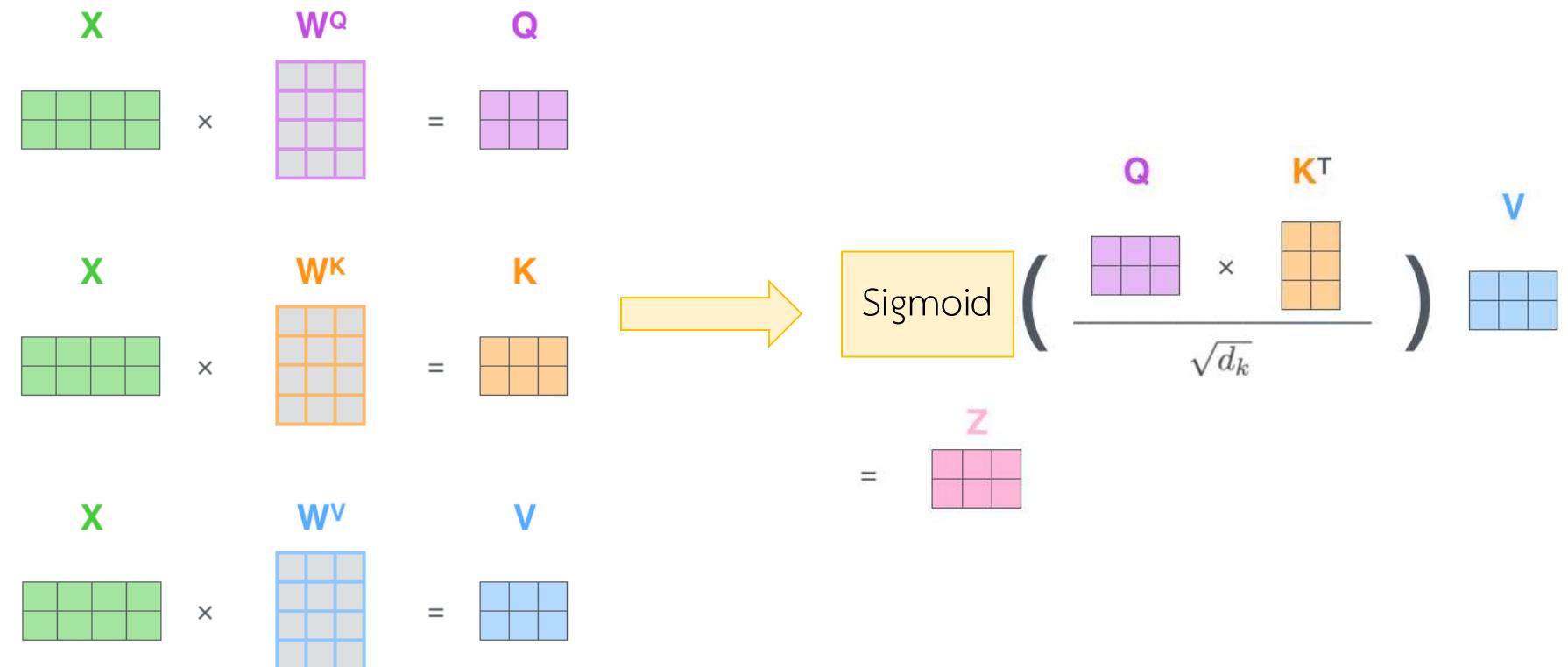
# Related Theory

Attention

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs
- (7) Depthwise Atrous

## Matrix Calculation of Self-Attention

- The first step is to calculate the Query, Key, and Value matrices.
- We do that by packing our embeddings into a matrix  $X$ , and multiplying it by the weight matrices we've trained ( $W^Q$ ,  $W^K$ ,  $W^V$ ).



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs
- (7) Depthwise Atrous

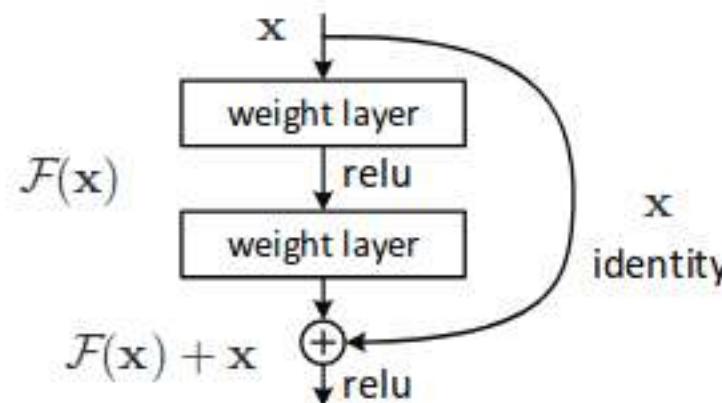


Figure 2. Residual learning: a building block.

1. The identity shortcuts ( $x$ ) can be directly used when the input and output are of the same dimensions.

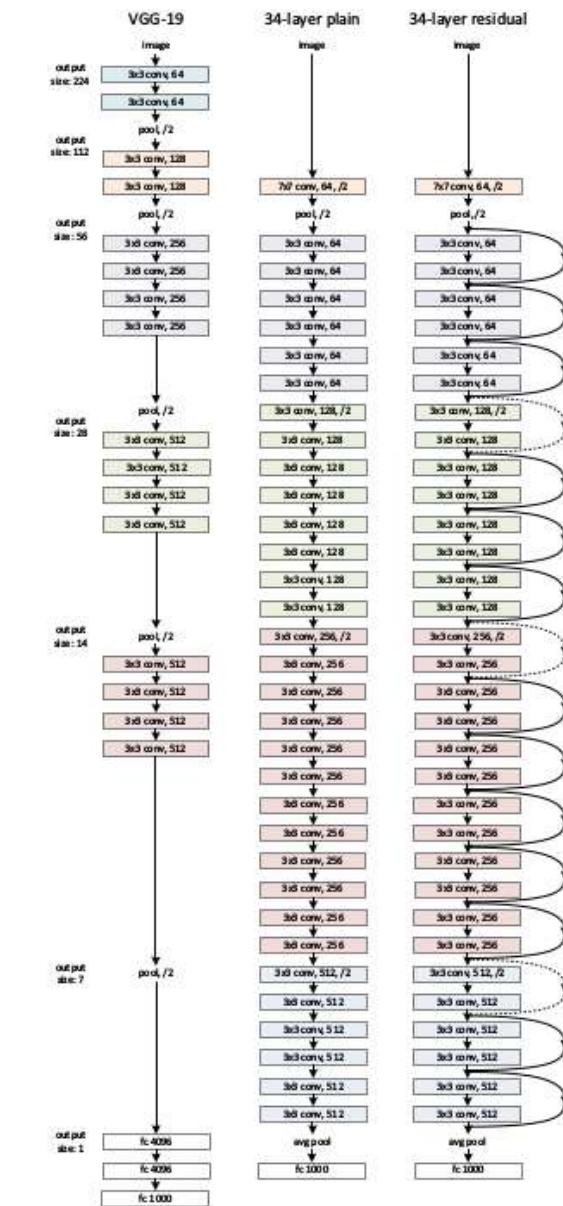
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}. \quad (1)$$

Residual block function when input and output dimensions are same

2. When the dimensions change, A) The shortcut still performs identity mapping, with extra zero entries padded with the increased dimension. B) The projection shortcut is used to match the dimension (done by  $1 \times 1$  conv) using the following formula

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s \mathbf{x}. \quad (2)$$

Residual block function when the input and output dimensions are not same.



# Evaluation

## 4.4. Evaluation metrics

To assess the quantitative performance, two overall benchmark metrics are used, i.e., *F1 score* (F1) and *intersection over union* (IoU). F1 is defined as

$$F1 = 2 \frac{\text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}}, \text{Pre} = \frac{tp}{tp + fp}, \text{Rec} = \frac{tp}{tp + fn}. \quad (9)$$

```
>>> from sklearn.metrics import f1_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> f1_score(y_true, y_pred, average='macro')
0.26...
>>> f1_score(y_true, y_pred, average='micro')
0.33...
>>> f1_score(y_true, y_pred, average='weighted')
0.26...
>>> f1_score(y_true, y_pred, average=None)
array([0.8, 0., 0.])
```

Here,  $tp, fp$  and  $fn$  are the number of true positives, false positives and false negatives, respectively.

IoU is defined as:

$$\text{IoU}(\mathcal{P}_m, \mathcal{P}_{gt}) = \frac{|\mathcal{P}_m \cap \mathcal{P}_{gt}|}{|\mathcal{P}_m \cup \mathcal{P}_{gt}|}, \quad (10)$$

where  $\mathcal{P}_{gt}$  is the set of ground truth pixels and  $\mathcal{P}_m$  is the set of prediction pixels, ‘ $\cap$ ’ and ‘ $\cup$ ’ denote *intersection* and *union* operations,

# Evaluation

```
>>> from sklearn.metrics import f1_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> f1_score(y_true, y_pred, average='macro')
0.26...
>>> f1_score(y_true, y_pred, average='micro')
0.33...
>>> f1_score(y_true, y_pred, average='weighted')
0.26...
>>> f1_score(y_true, y_pred, average=None)
array([0.8, 0., 0.])
```

## 1. Micro-average Method

In Micro-average method, you sum up the individual true positives, false positives, and false negatives of the system for different sets and the apply them to get the statistics. For example, for a set of data, the system's

True positive (TP1) = 12  
False positive (FP1) = 9  
False negative (FN1) = 3

Then precision (P1) and recall (R1) will be  $P1 = \frac{TP1}{TP1+FP1}$  and  $R1 = \frac{TP1}{TP1+FN1}$

and for a different set of data, the system's

True positive (TP2) = 50  
False positive (FP2) = 23  
False negative (FN2) = 9

Then precision (P2) and recall (R2) will be 68.49 and 84.75

Now, the average precision and recall of the system using the Micro-average method is

$$\text{Micro-average of precision} = \frac{TP1+TP2}{TP1+TP2+FP1+FP2} = \frac{12+50}{12+50+9+23} = 65.96$$

$$\text{Micro-average of recall} = \frac{TP1+TP2}{TP1+TP2+FN1+FN2} = \frac{12+50}{12+50+3+9} = 83.78$$

The Micro-average F-Score will be simply the harmonic mean of these two figures.

# Evaluation

```
>>> from sklearn.metrics import f1_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> f1_score(y_true, y_pred, average='macro')
0.26...
>>> f1_score(y_true, y_pred, average='micro')
0.33...
>>> f1_score(y_true, y_pred, average='weighted')
0.26...
>>> f1_score(y_true, y_pred, average=None)
array([0.8, 0., 0.])
```

## 2. Macro-average Method

The method is straight forward. Just take the average of the precision and recall of the system on different sets. For example, the macro-average precision and recall of the system for the given example is

$$\text{Macro-average precision} = \frac{P_1+P_2}{2} = \frac{57.14+68.49}{2} = 62.82$$

$$\text{Macro-average recall} = \frac{R_1+R_2}{2} = \frac{80+84.75}{2} = 82.25$$

The Macro-average F-Score will be simply the harmonic mean of these two figures.

Suitability Macro-average method can be used when you want to know how the system performs overall across the sets of data. You should not come up with any specific decision with this average.

On the other hand, micro-average can be a useful measure when your dataset varies in size.

share improve this answer follow

edited Nov 9 '17 at 14:44



jaggi

103 ● 1

answered Dec 30 '16 at 9:53



Rahul Reddy Vemireddy

605 ● 4 ● 6

# Related Theory

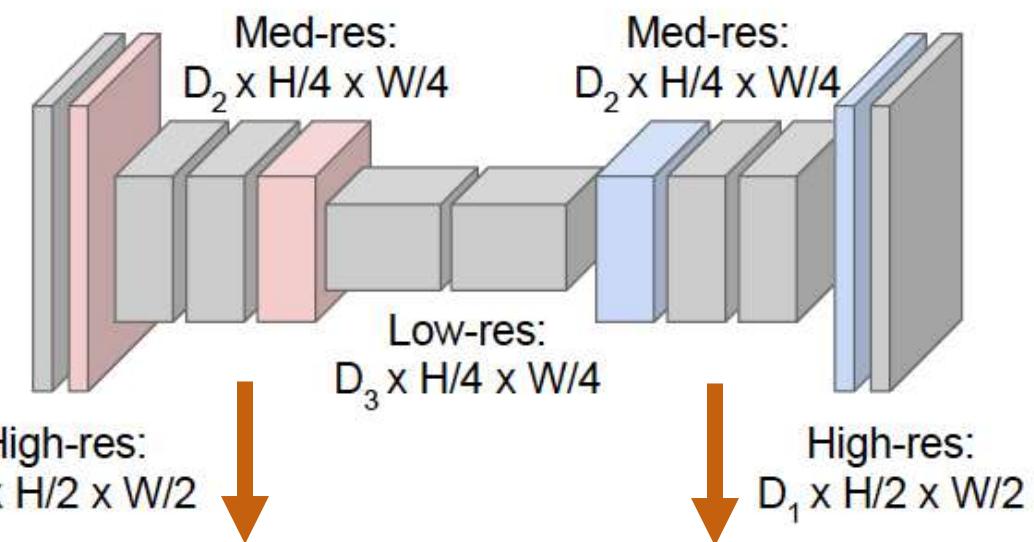
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs

**Downsampling:**  
Pooling, strided convolution



Input:  
 $3 \times H \times W$

Design network as a bunch of convolutional layers, with  
**downsampling** and **upsampling** inside the network!



Encoder Network

Decoder Network

Deep Encoder-Decoder Network (DCED)

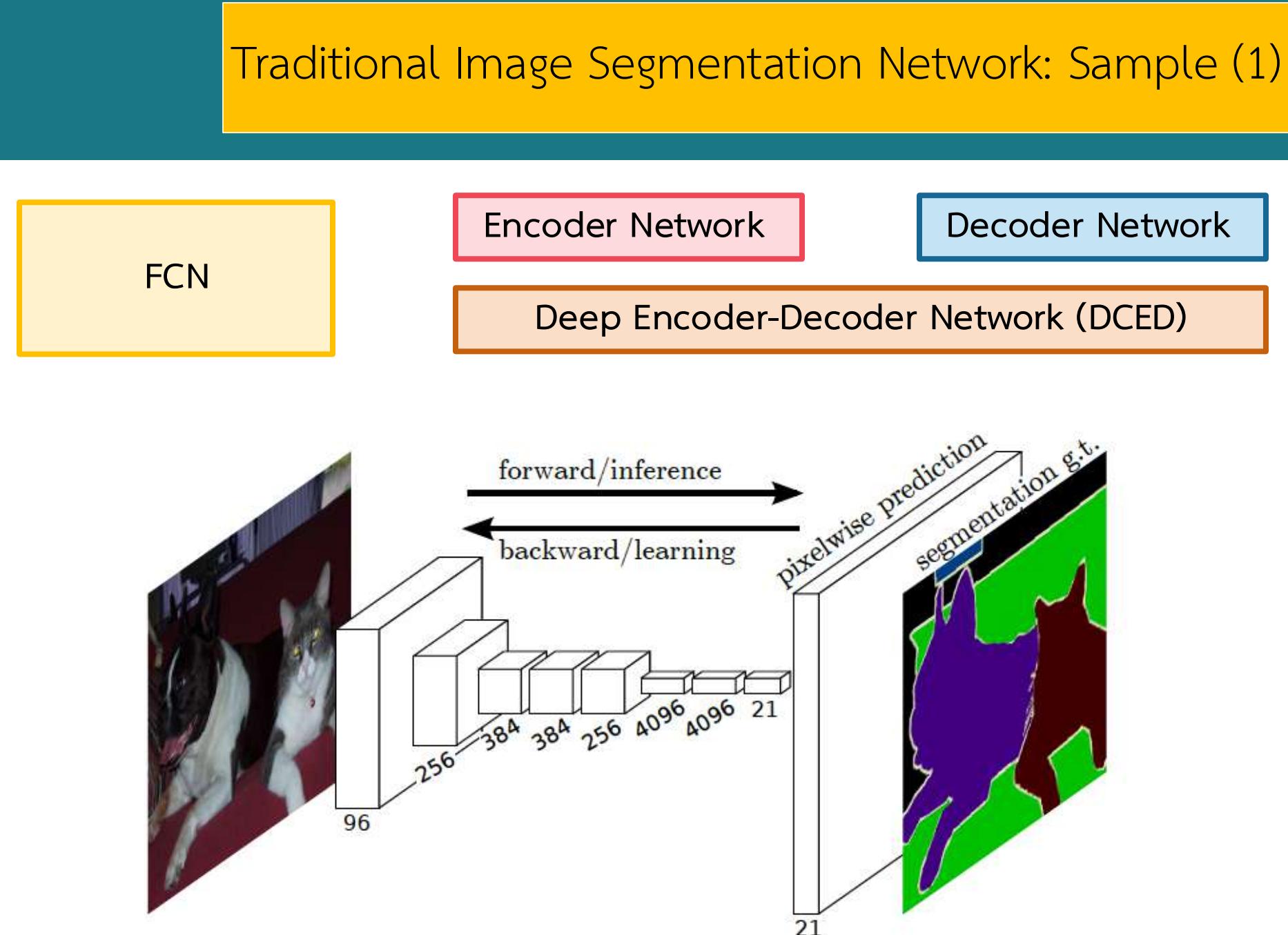
**Upsampling:**  
Unpooling or strided transpose convolution



Predictions:  
 $H \times W$

# Related Theory

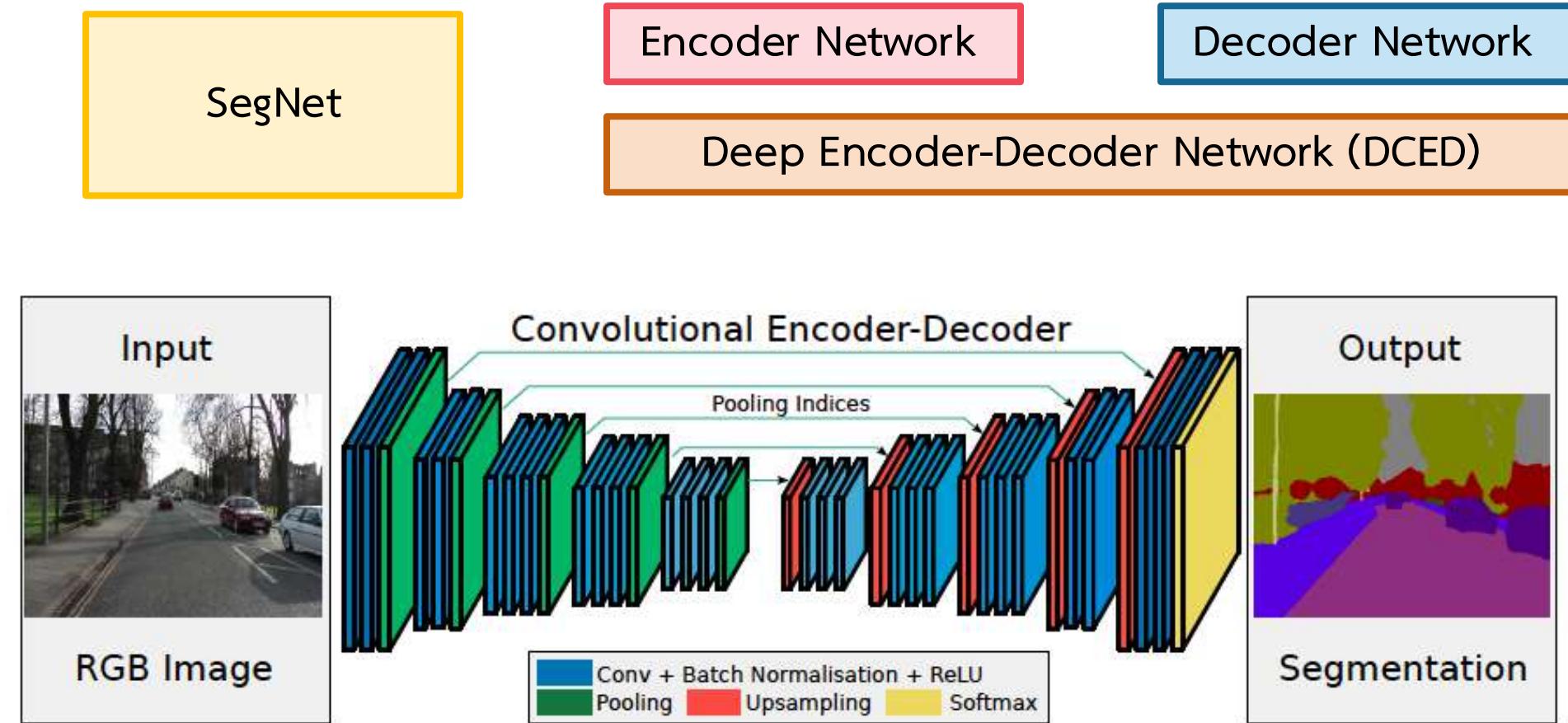
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs

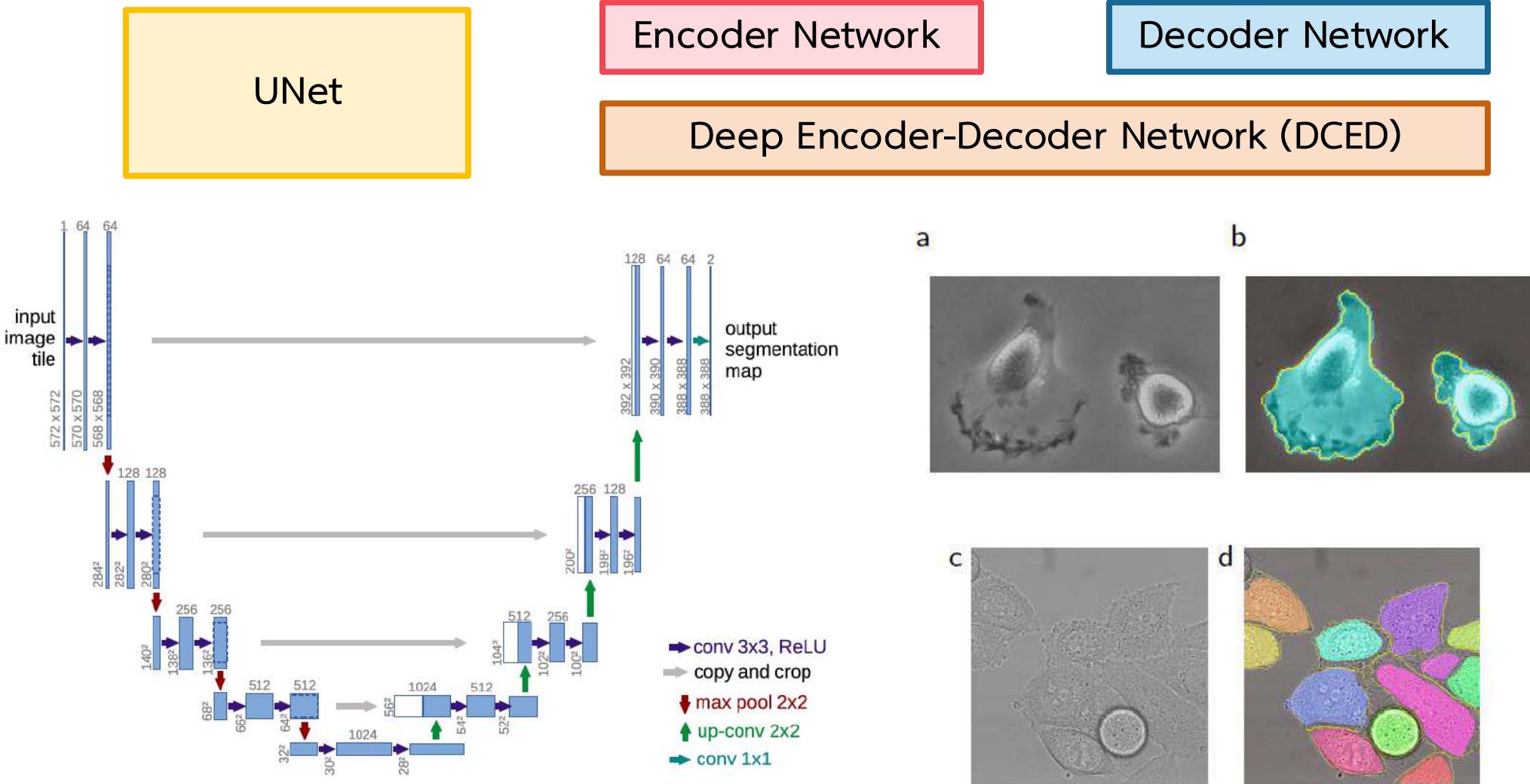
## Traditional Image Segmentation Network: Sample (2)



# Related Theory

## Traditional Image Segmentation Network: Sample (3)

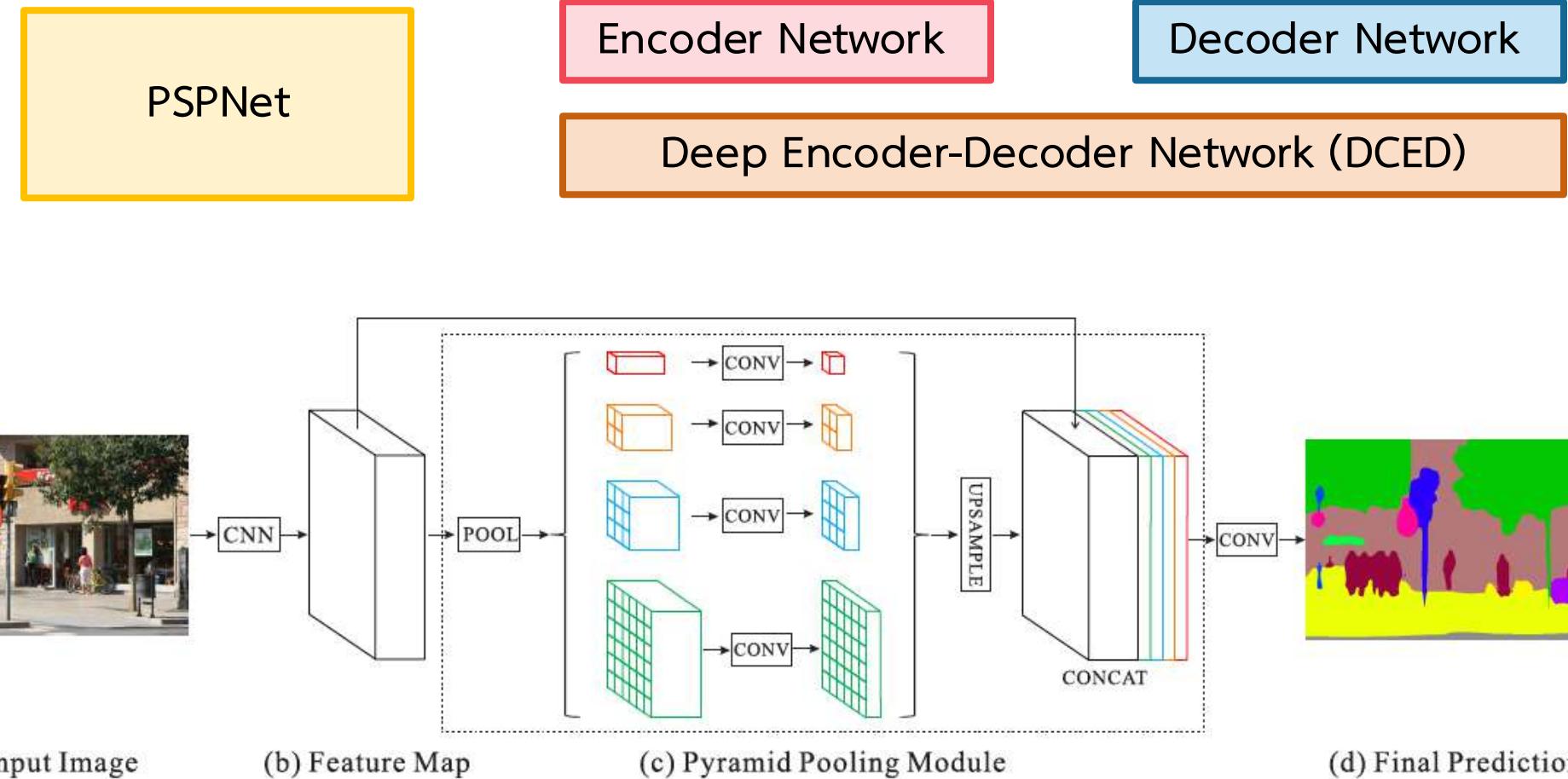
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs



# Related Theory

## Traditional Image Segmentation Network: Sample (4)

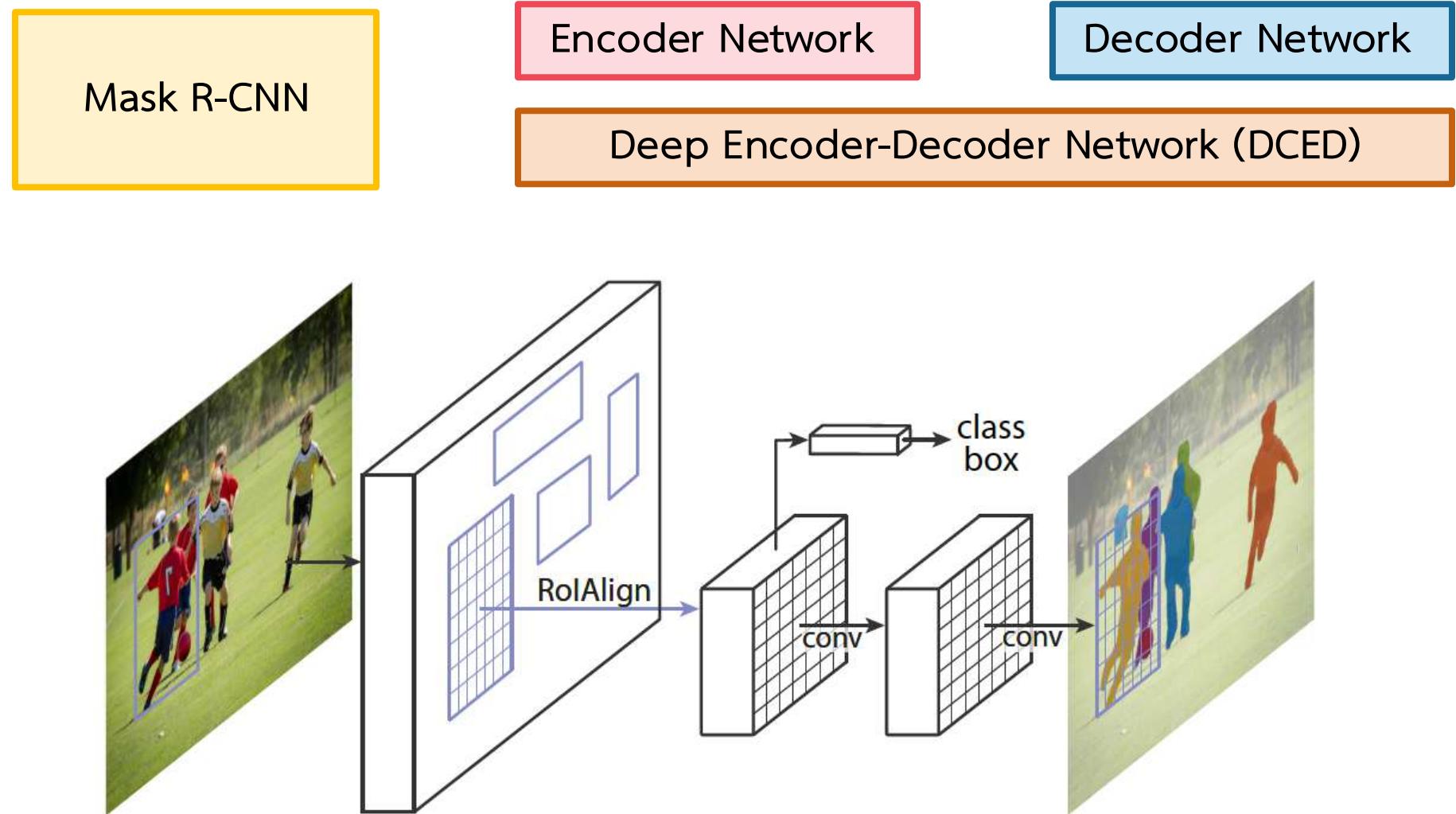
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs

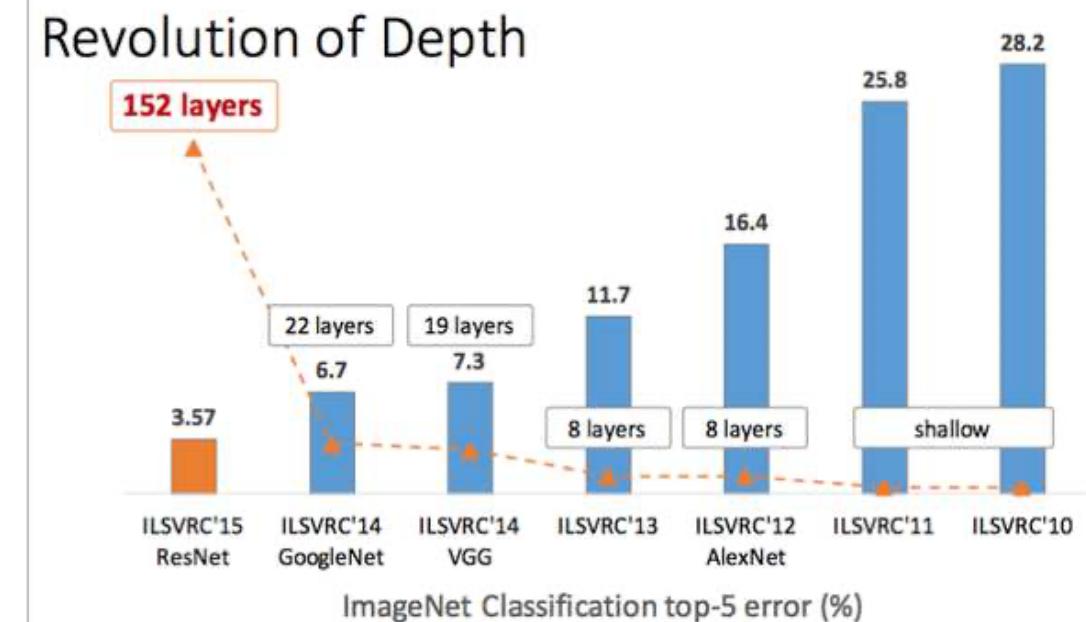
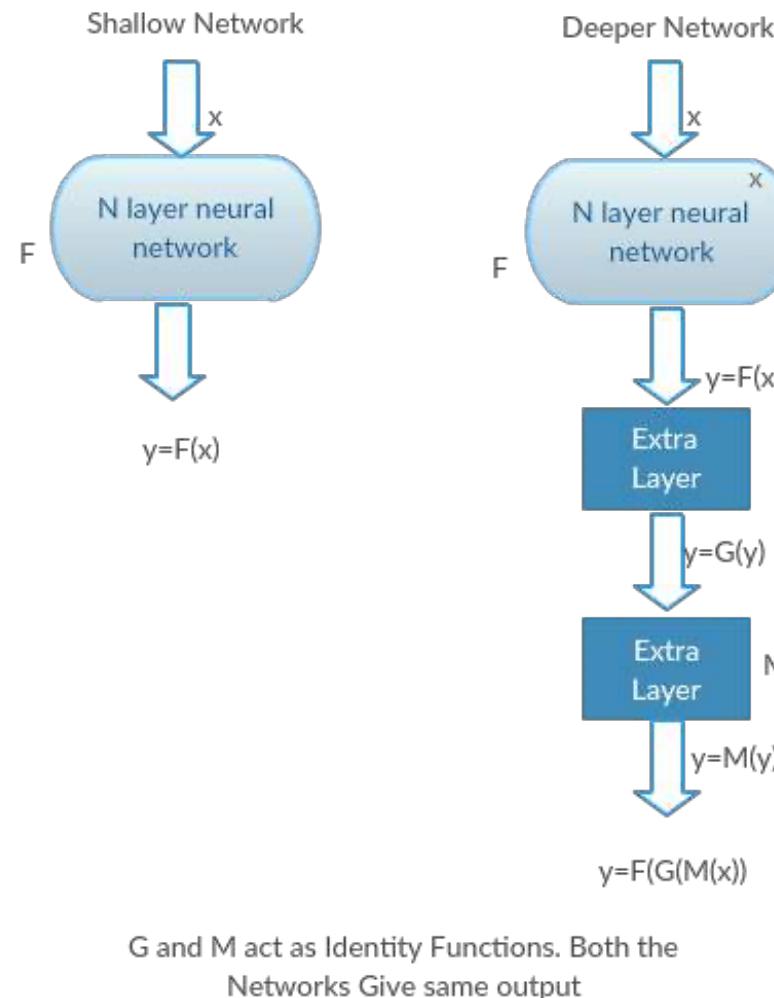
## Traditional Image Segmentation Network: Sample (5)



# Related Theory

ResNet (Microsoft)

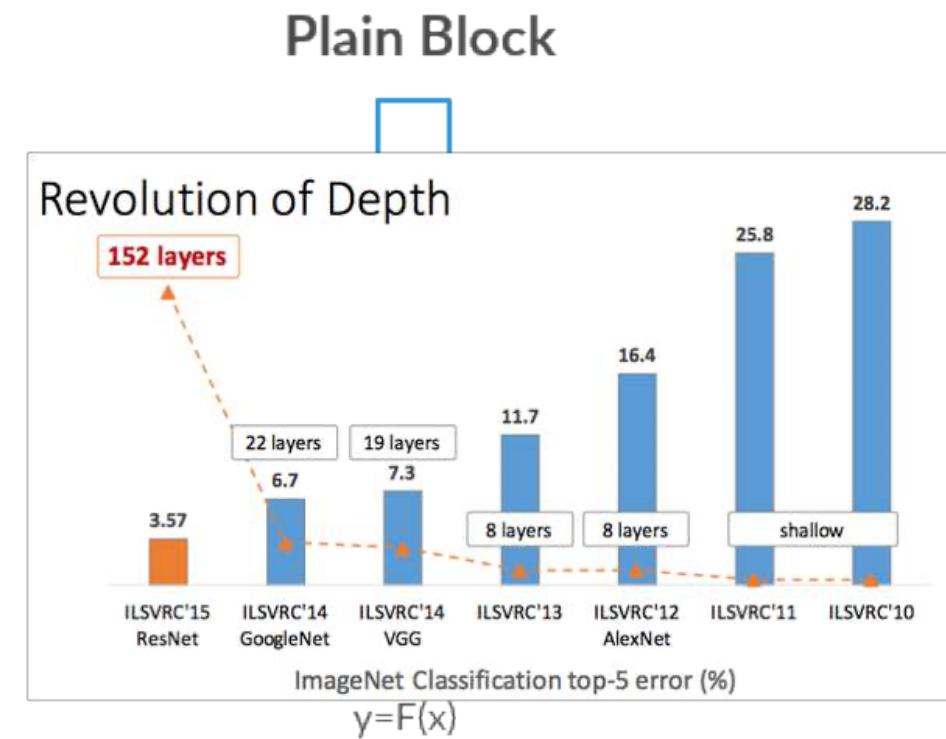
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs



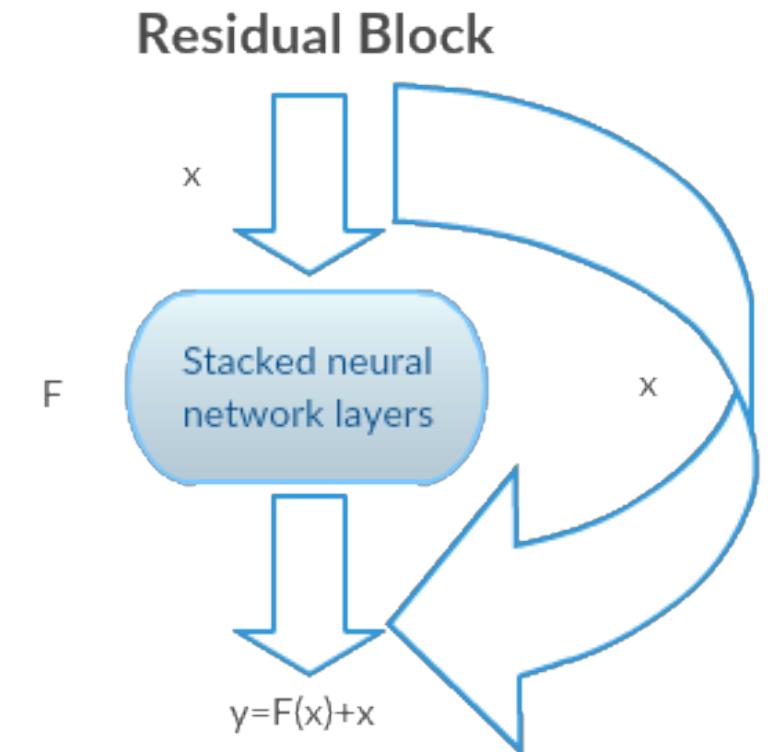
# Related Theory

ResNet (Microsoft)

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Design CNNs



Hard to get  $F(x)=x$  and make  $y=x$   
an identity mapping



Easy to get  $F(x)=0$  and make  $y=x$   
an identity mapping

Encoder Network (VGG (Residual) Style)

# Related Theory

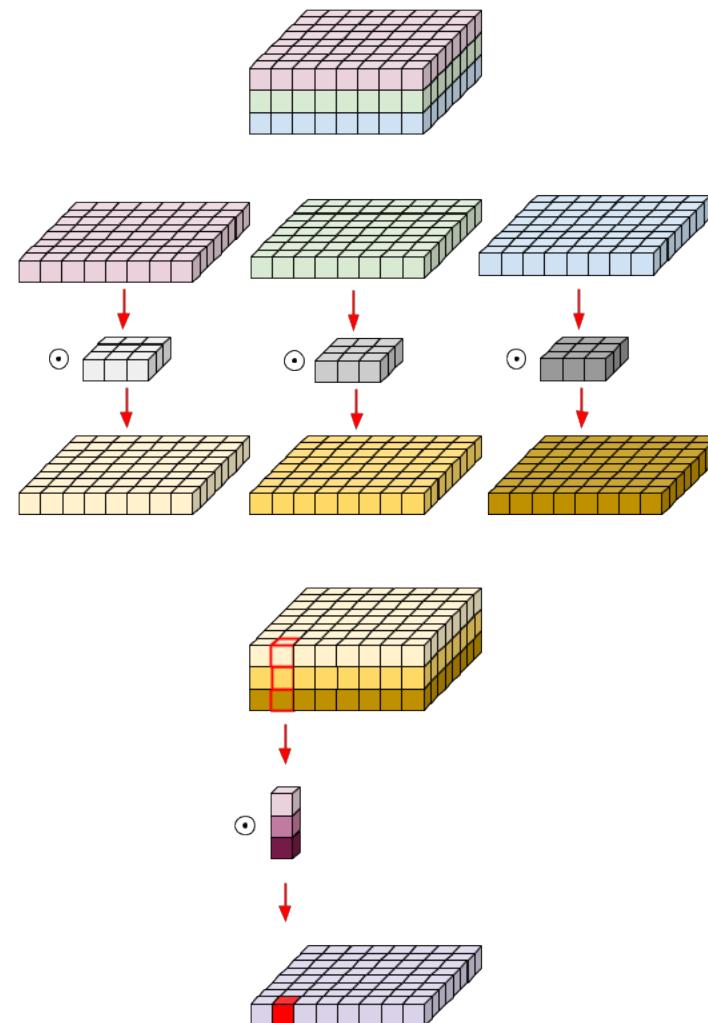
ResNet (Microsoft)

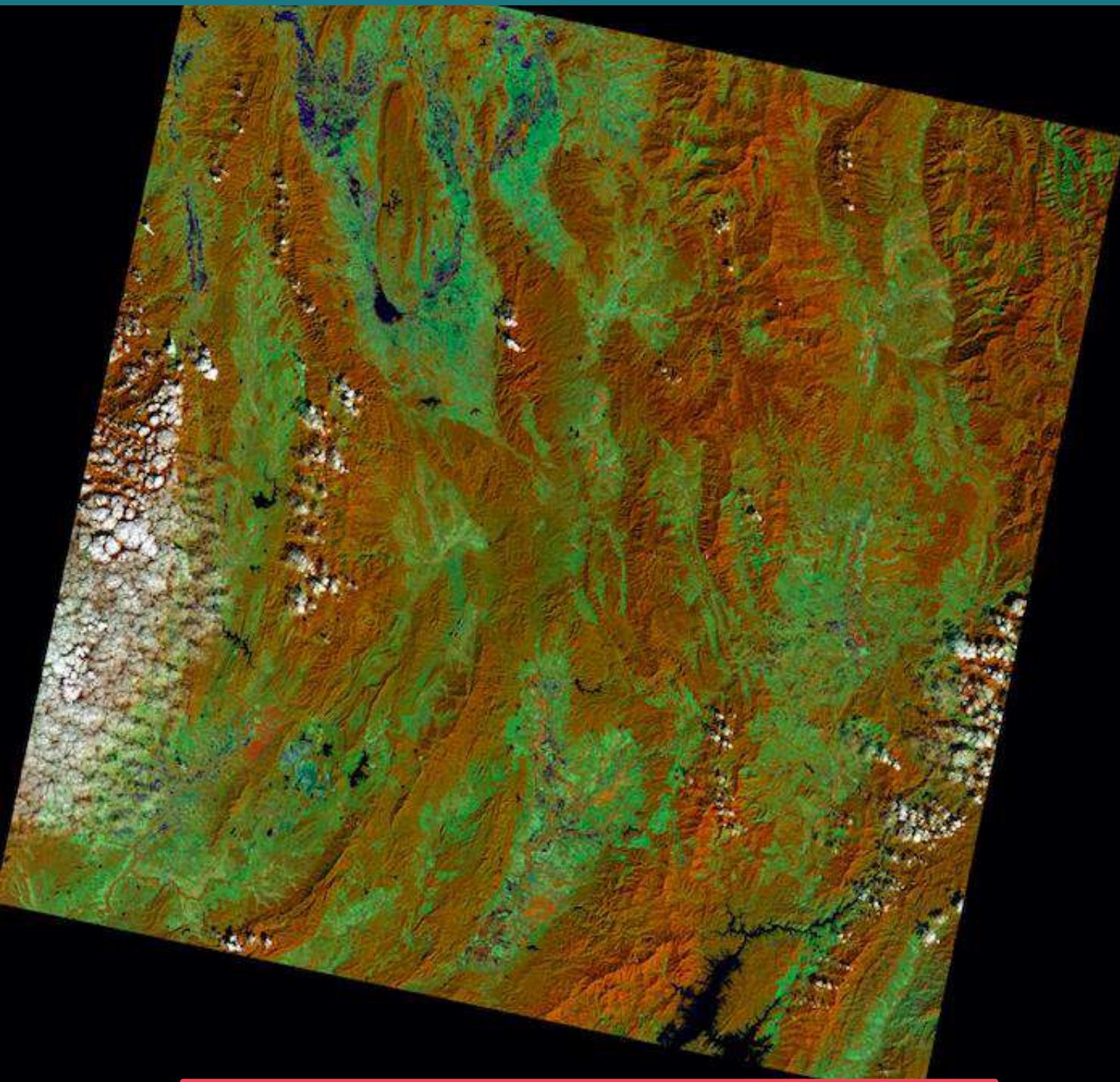
- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - (4) Channel Attention
  - (5) Feature Fusion
  - (6) Design CNNs
- More layers is better
    - but because of the vanishing gradient problem
    - model weights of the first layers can not be updated correctly through the backpropagation of the error gradient
    - the chain rule multiplies error gradient values lower than one and then, when the gradient error comes to the first layers, its value goes to zero
  - Objective of Resnet is preserve the gradient
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

# Related Theory

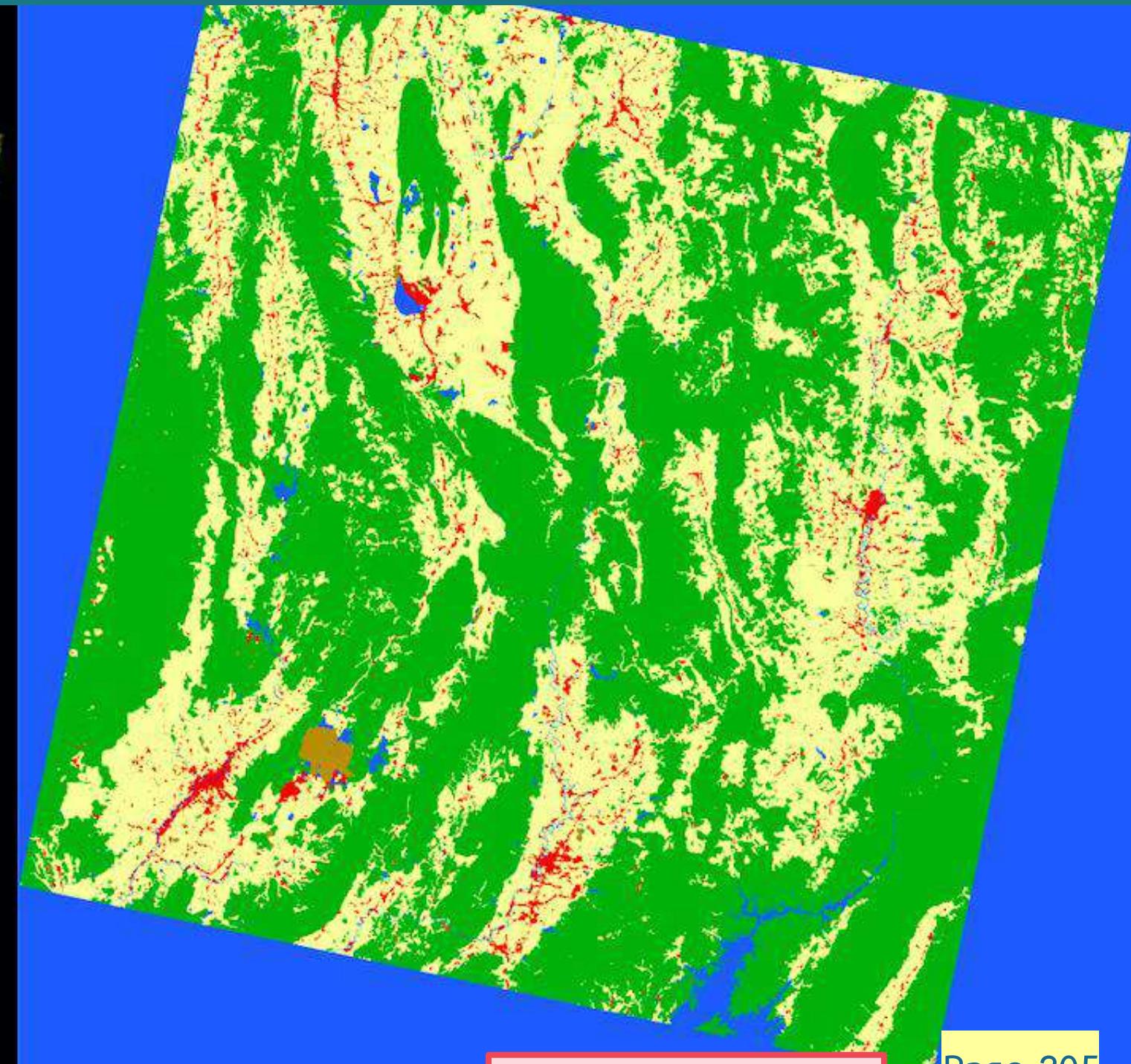
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs

## Depth-wise Separable Convolution

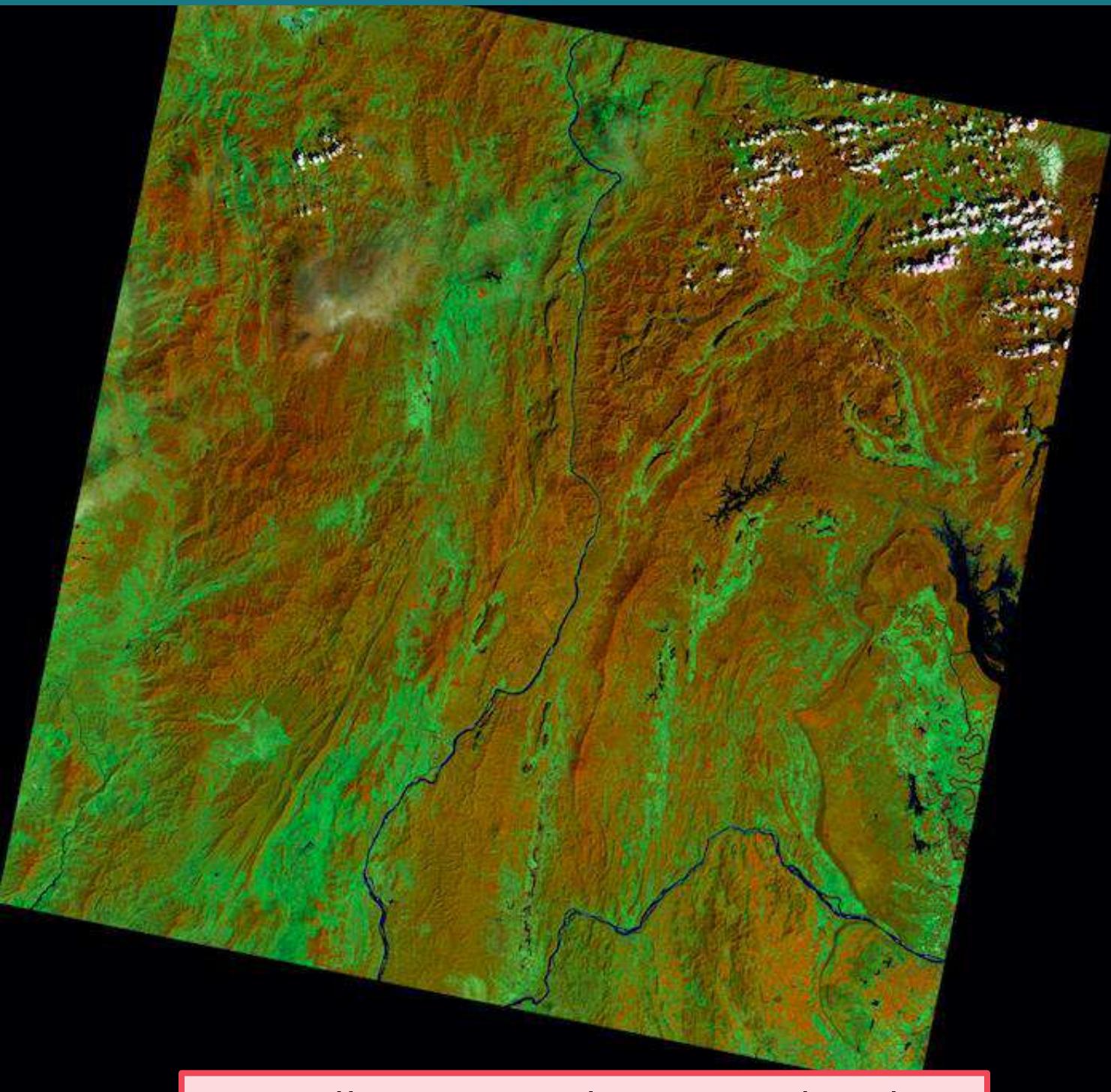




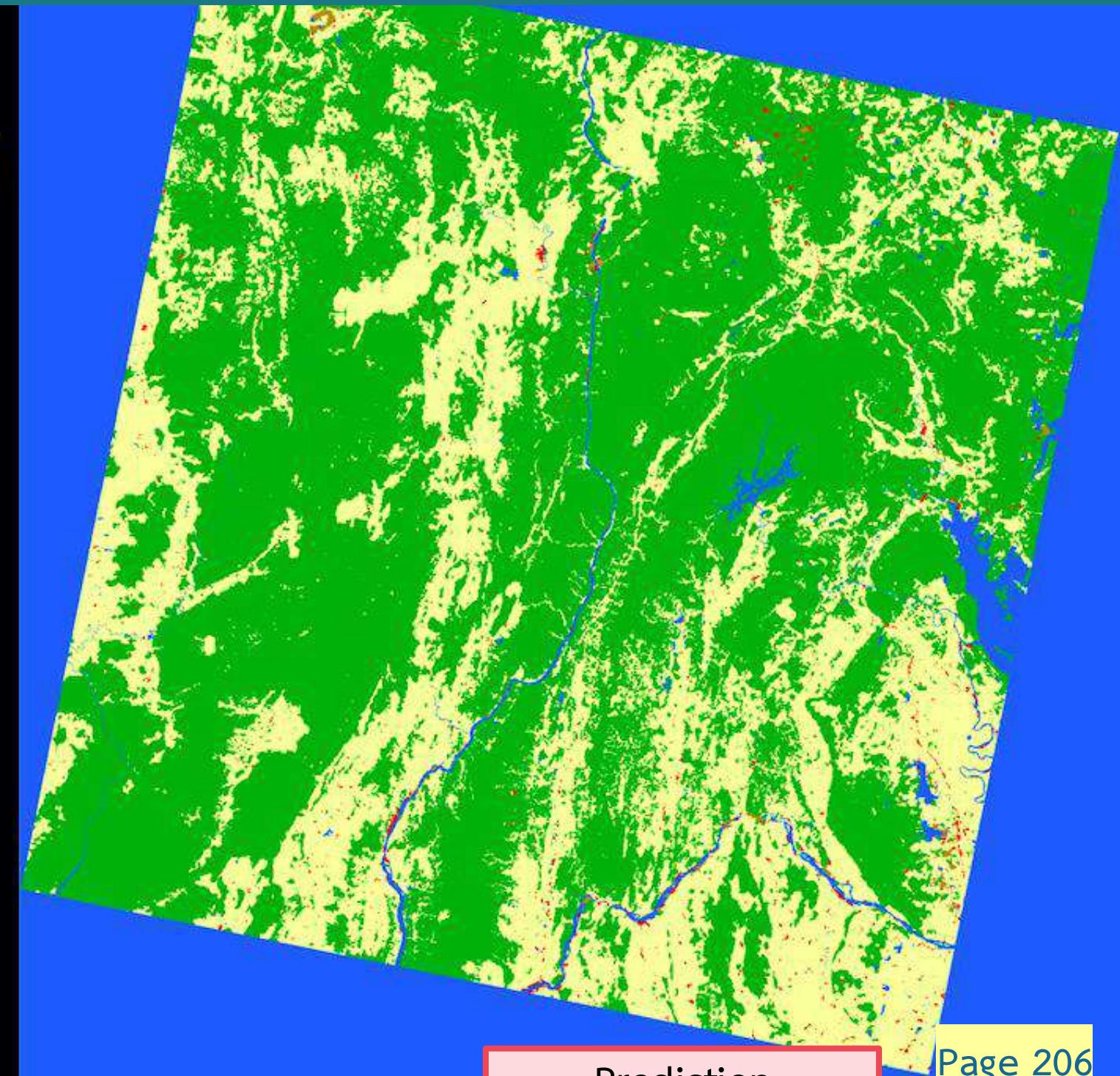
Satellite Image without ground truth



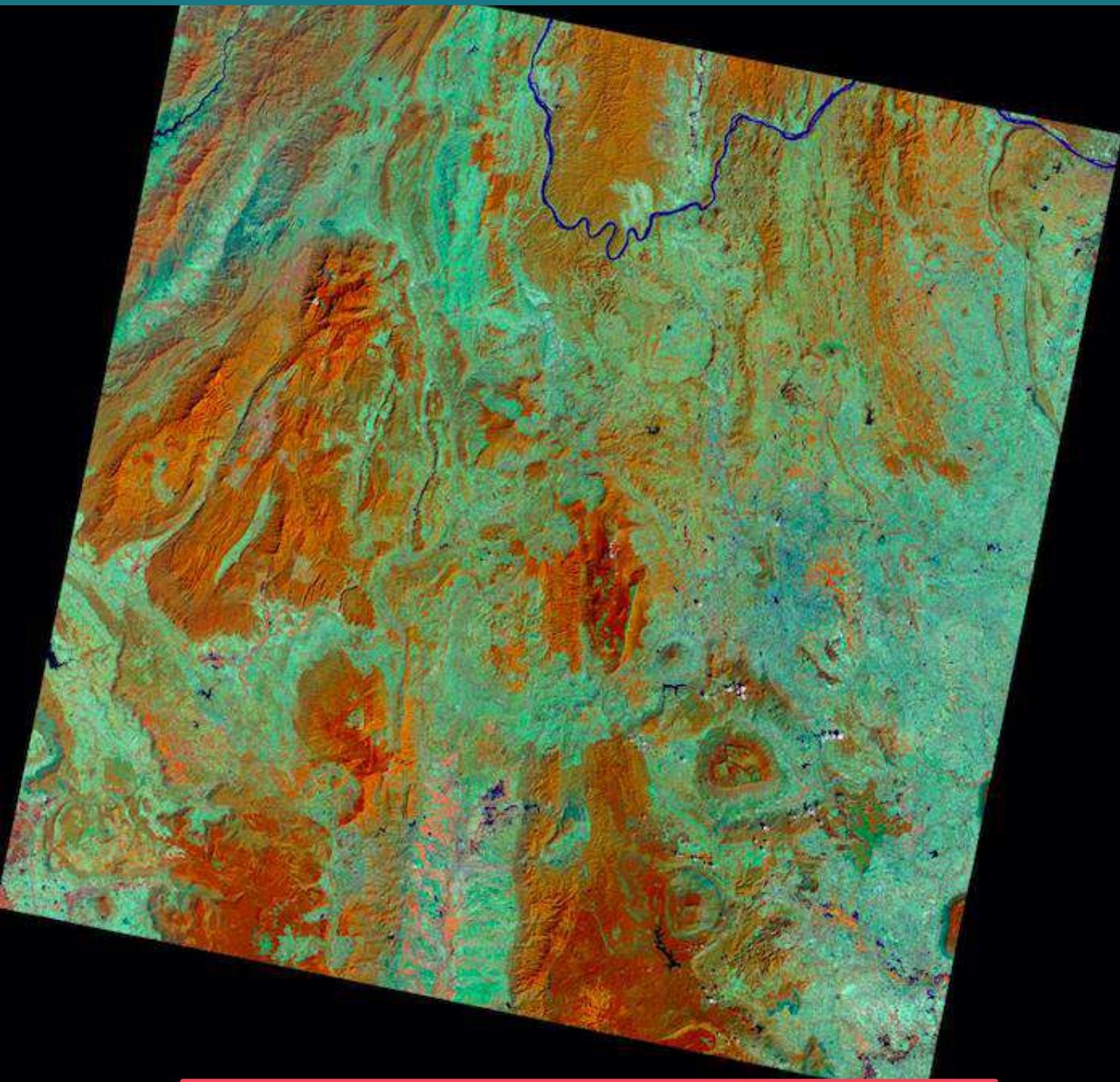
Prediction



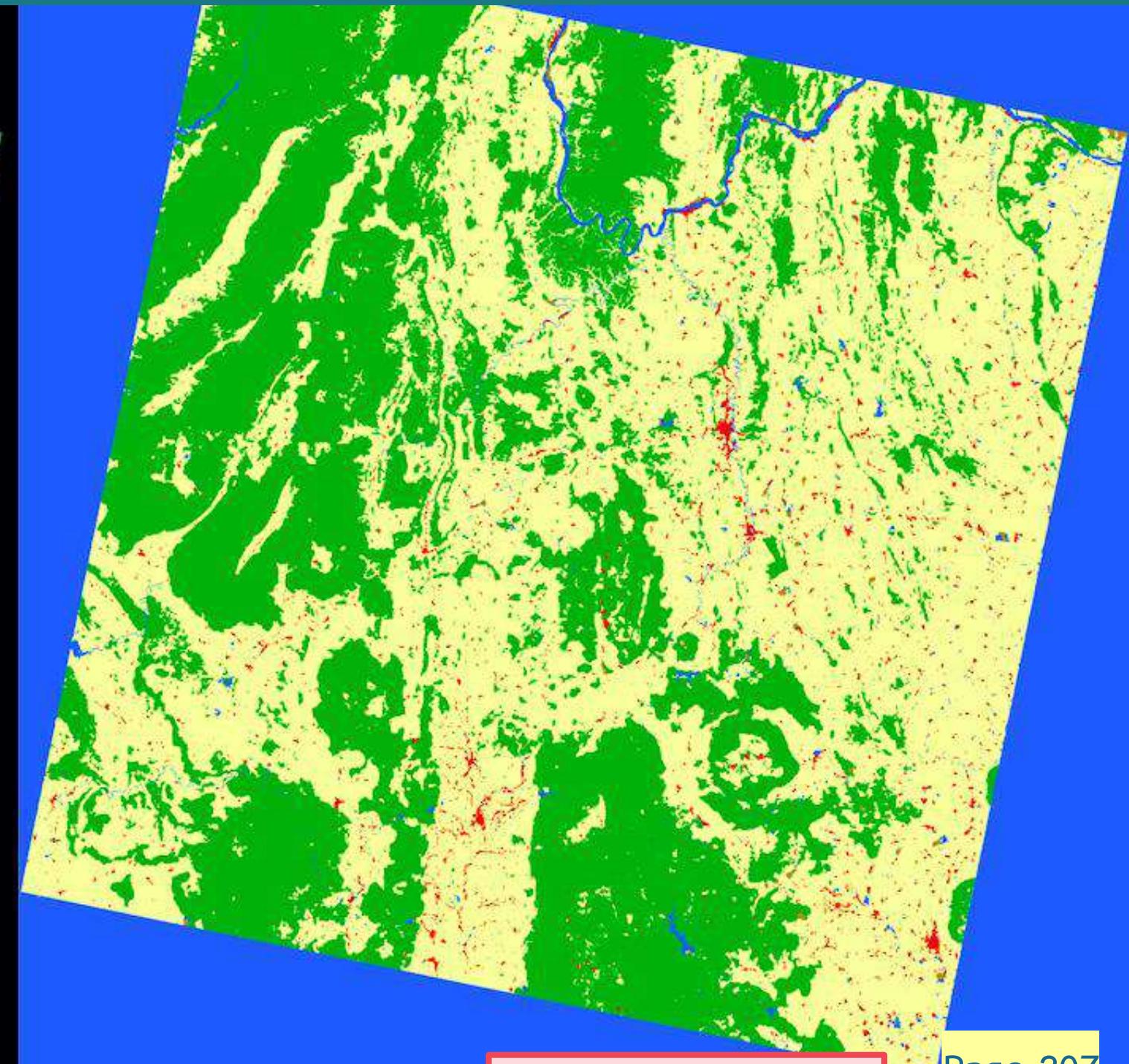
Satellite Image without ground truth



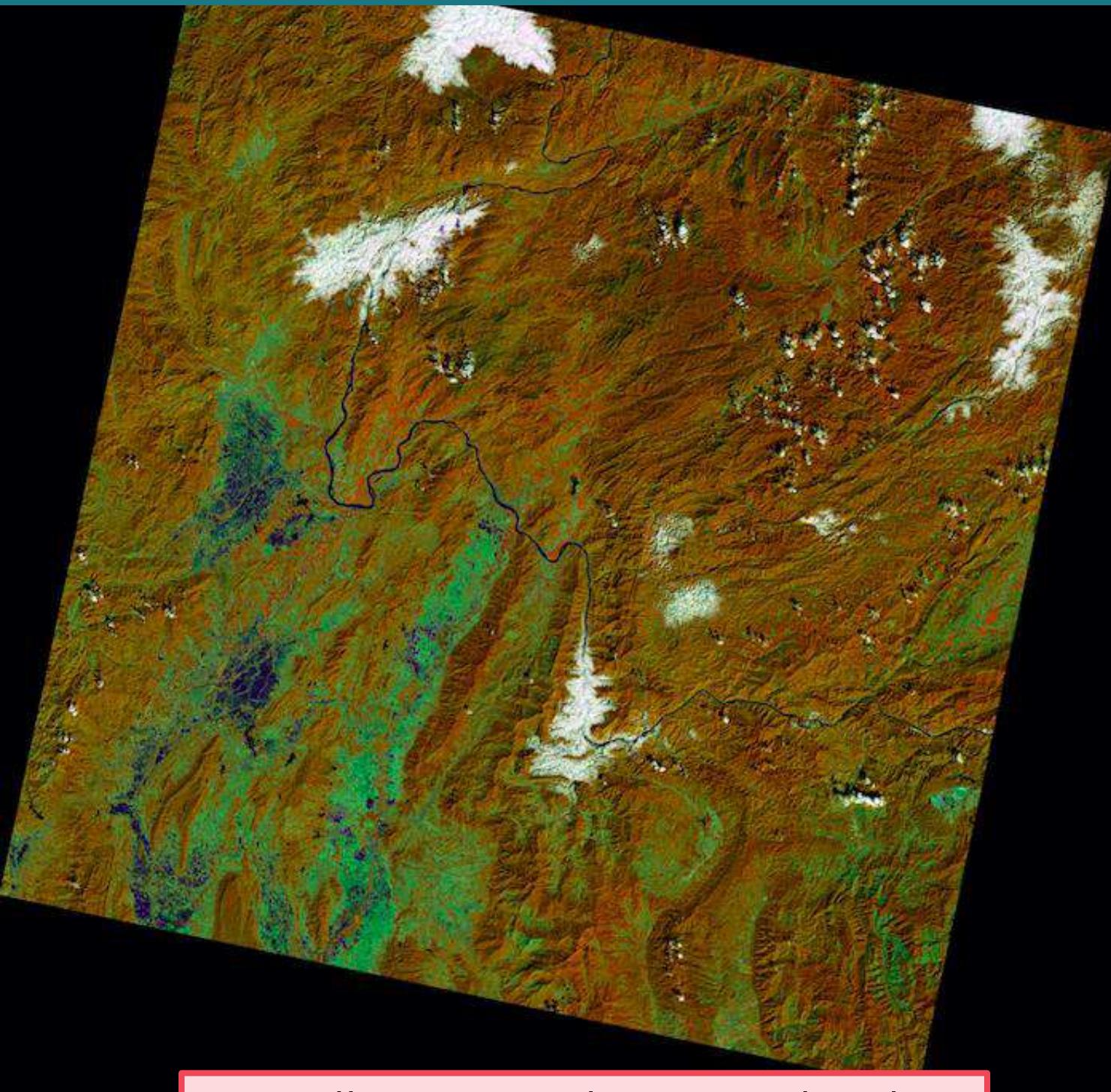
Prediction



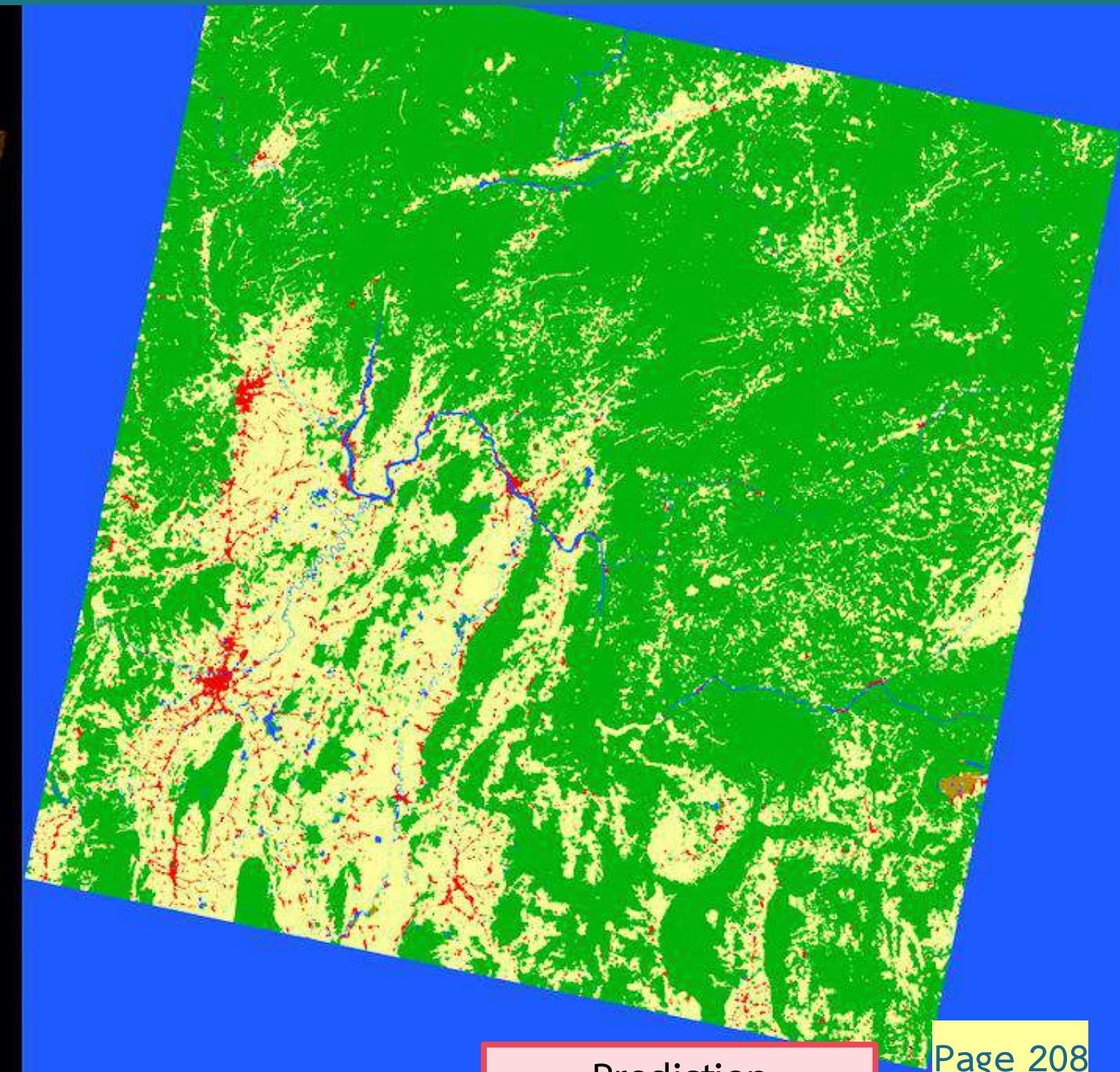
Satellite Image without ground truth



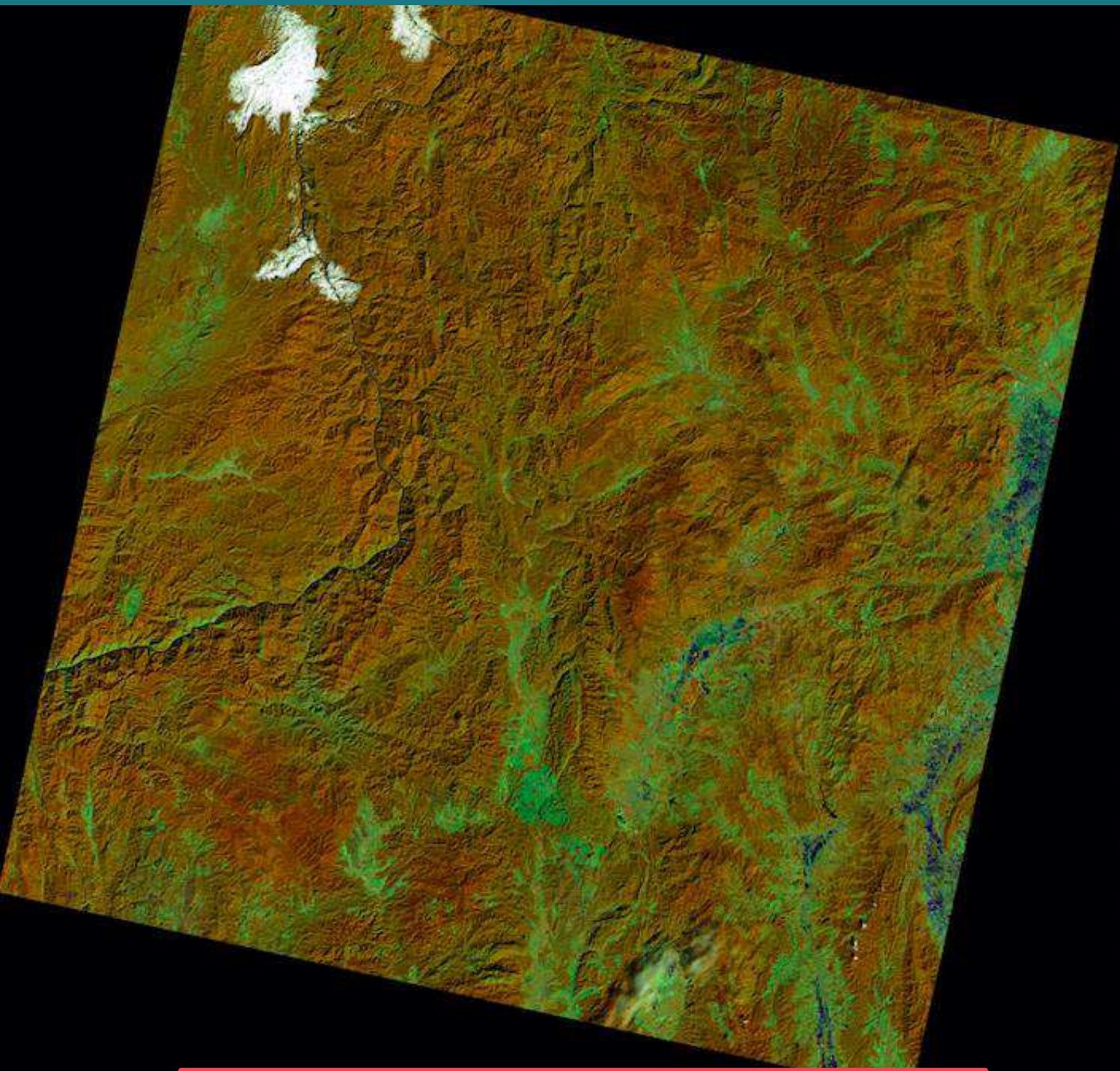
Prediction



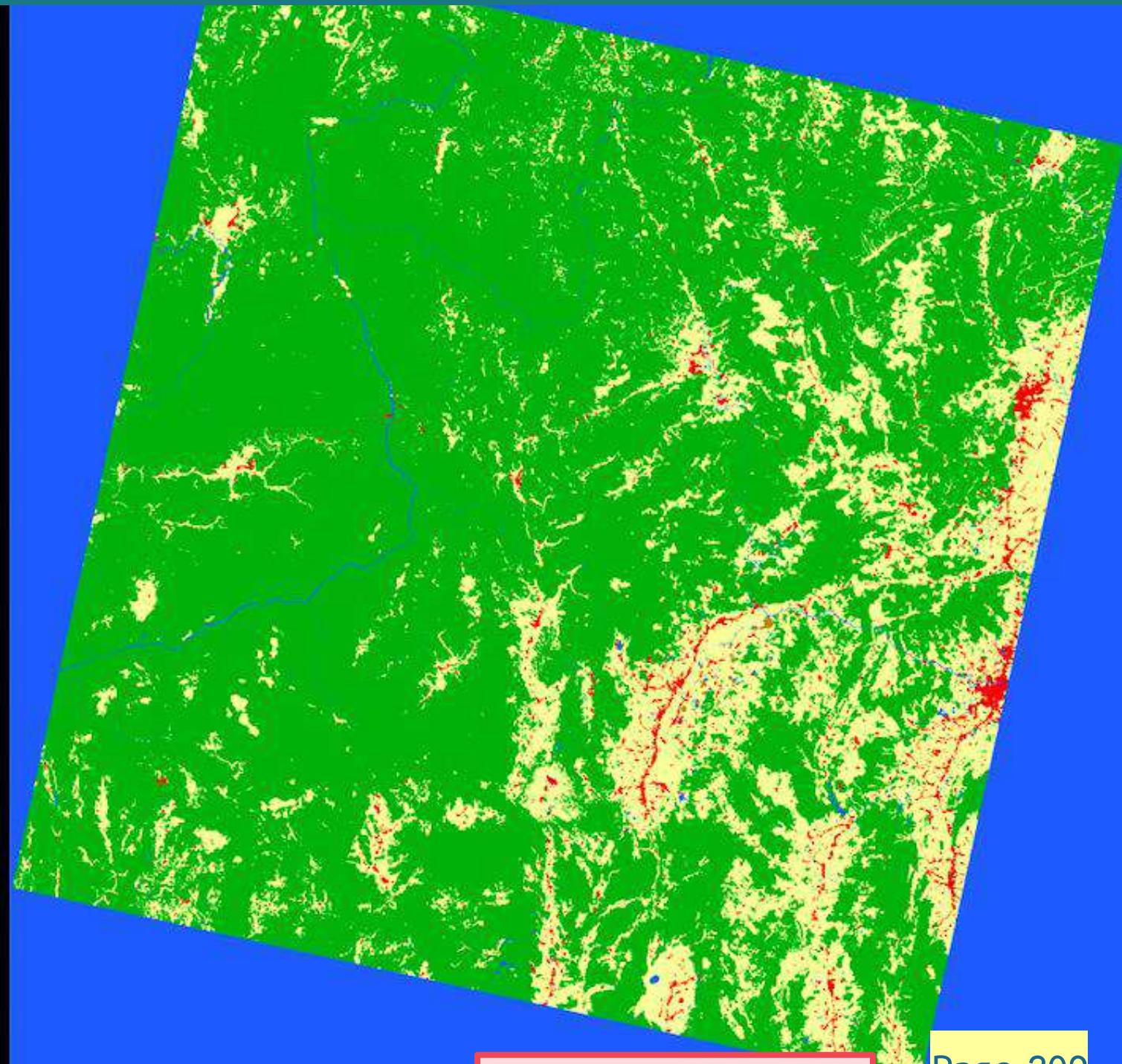
Satellite Image without ground truth



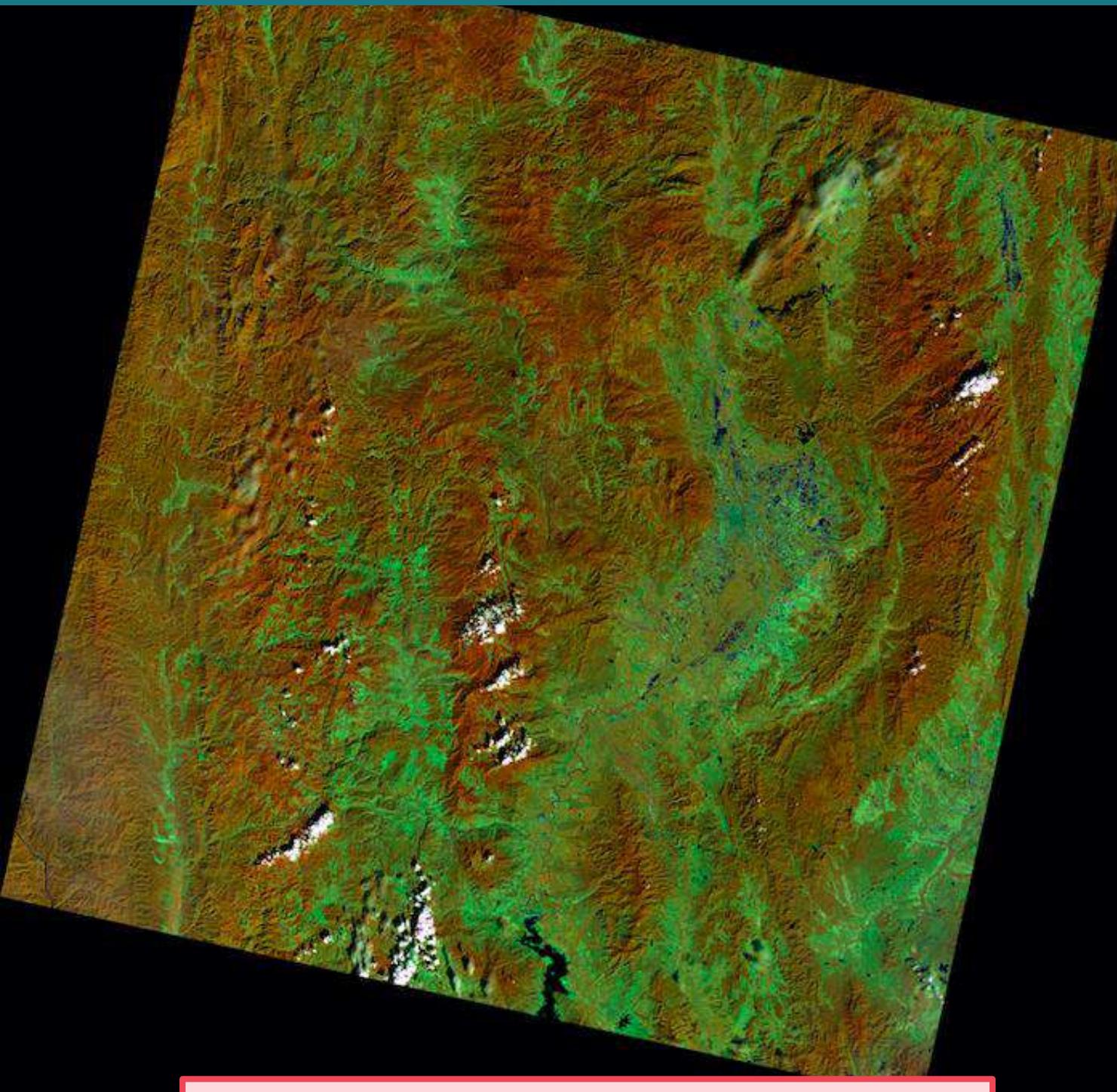
Prediction



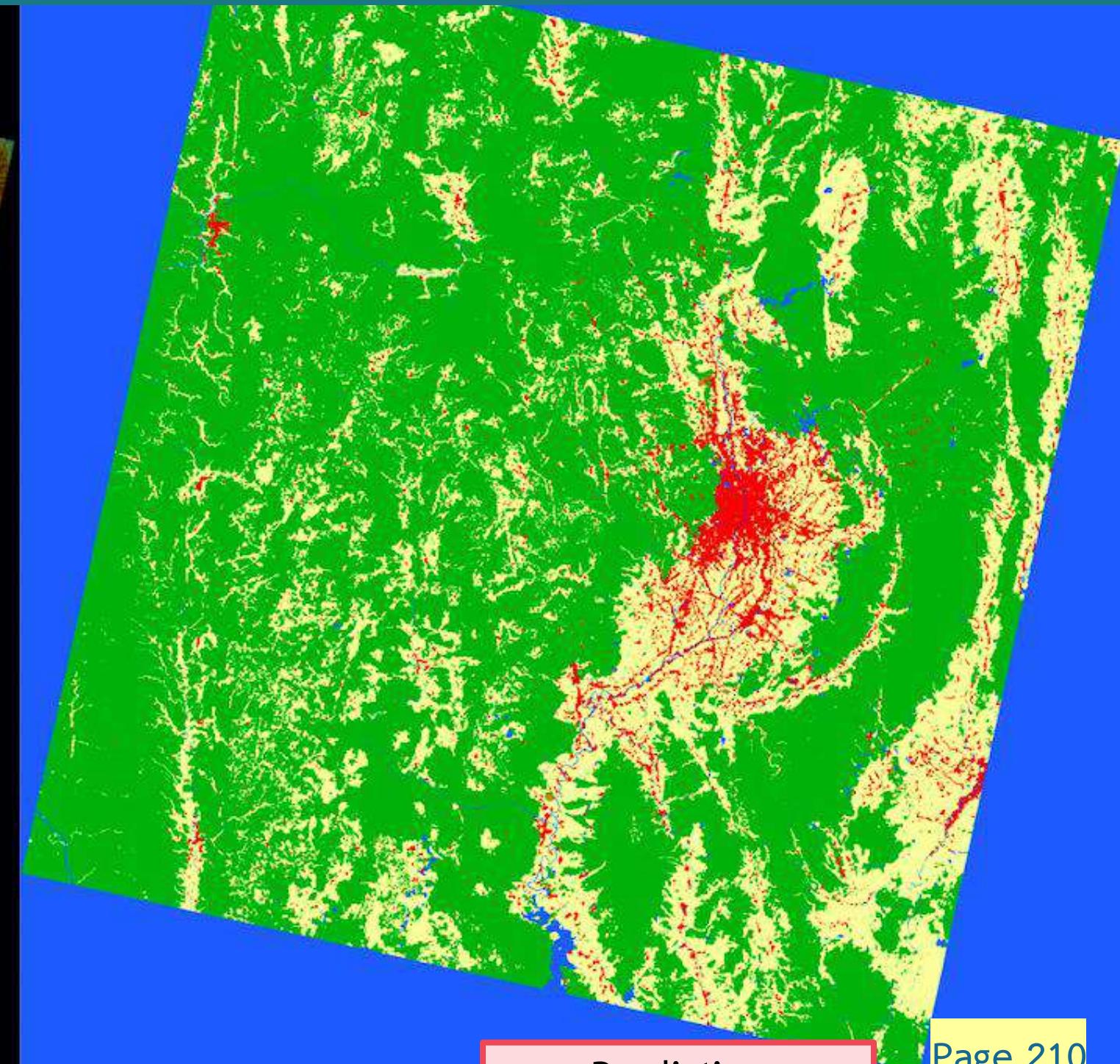
Satellite Image without ground truth



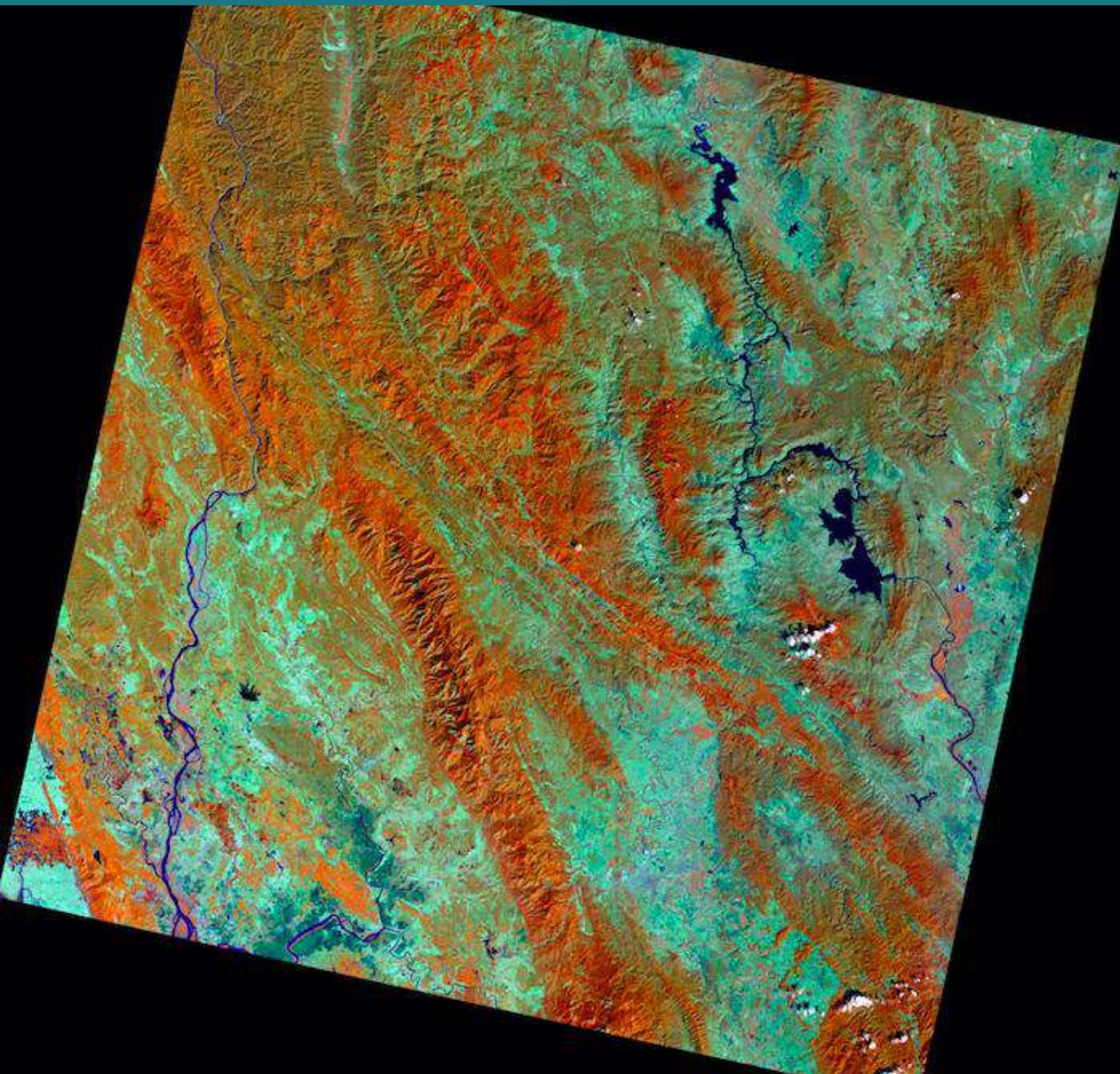
Prediction



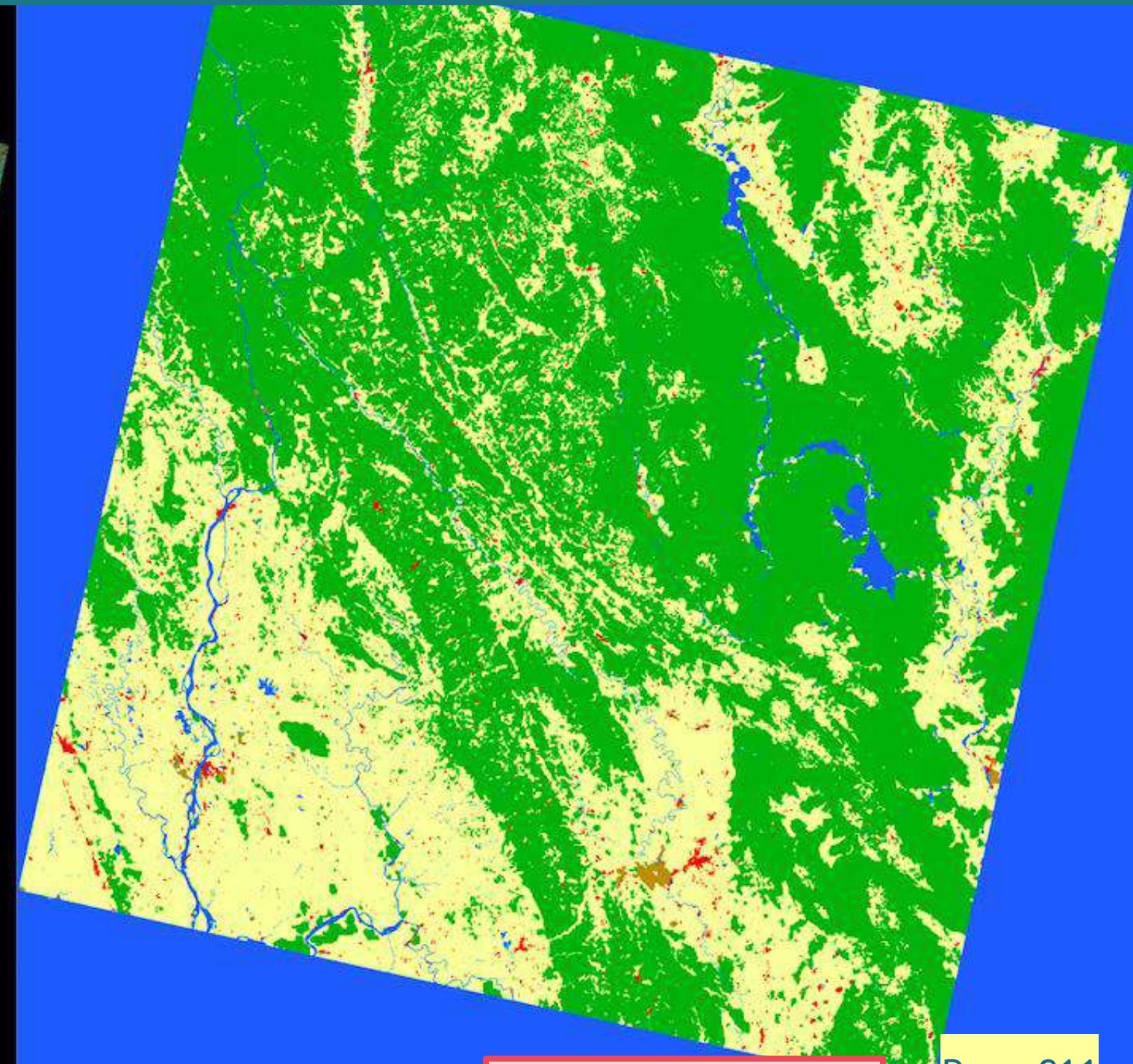
Satellite Image without ground truth



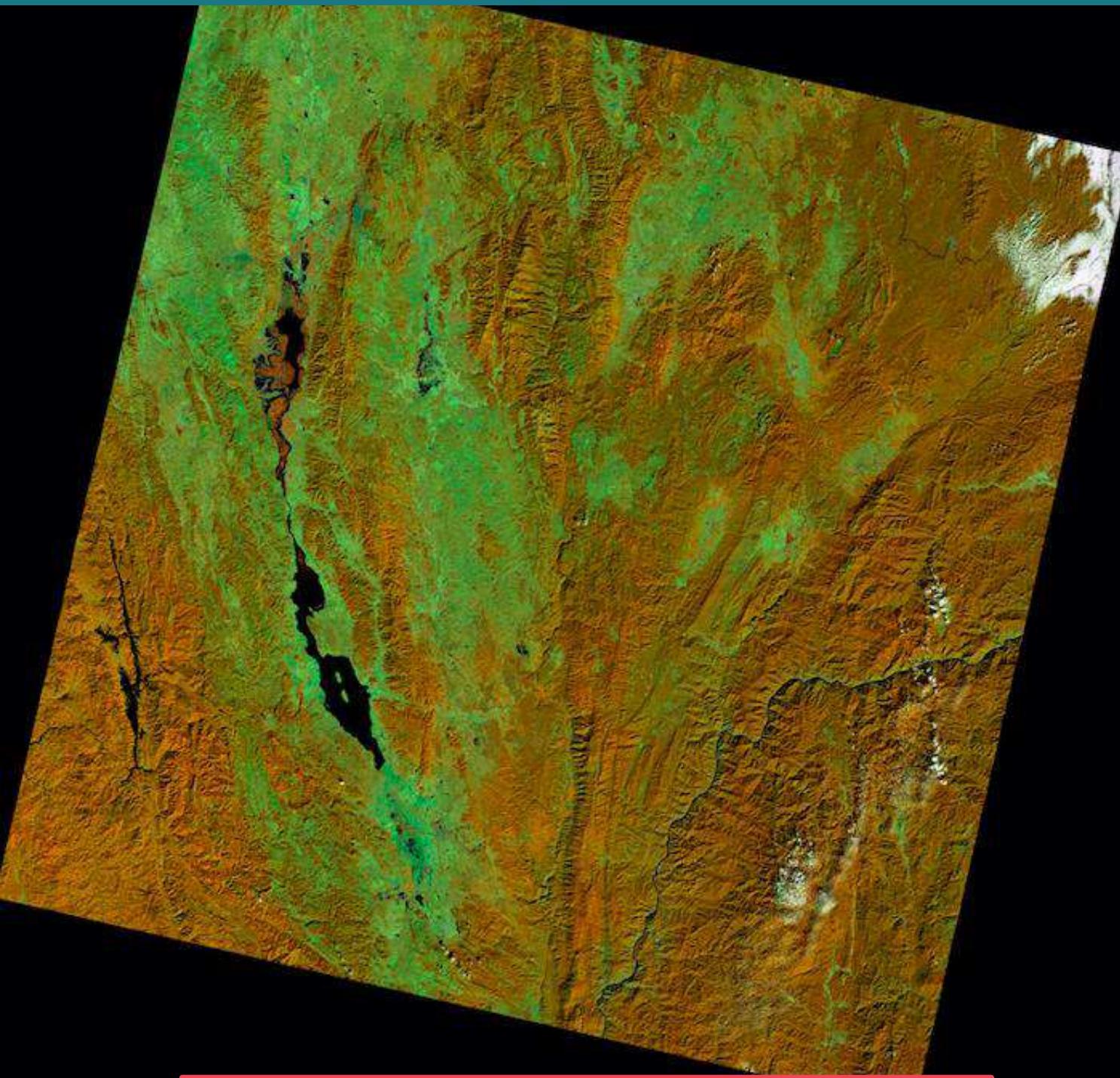
Prediction



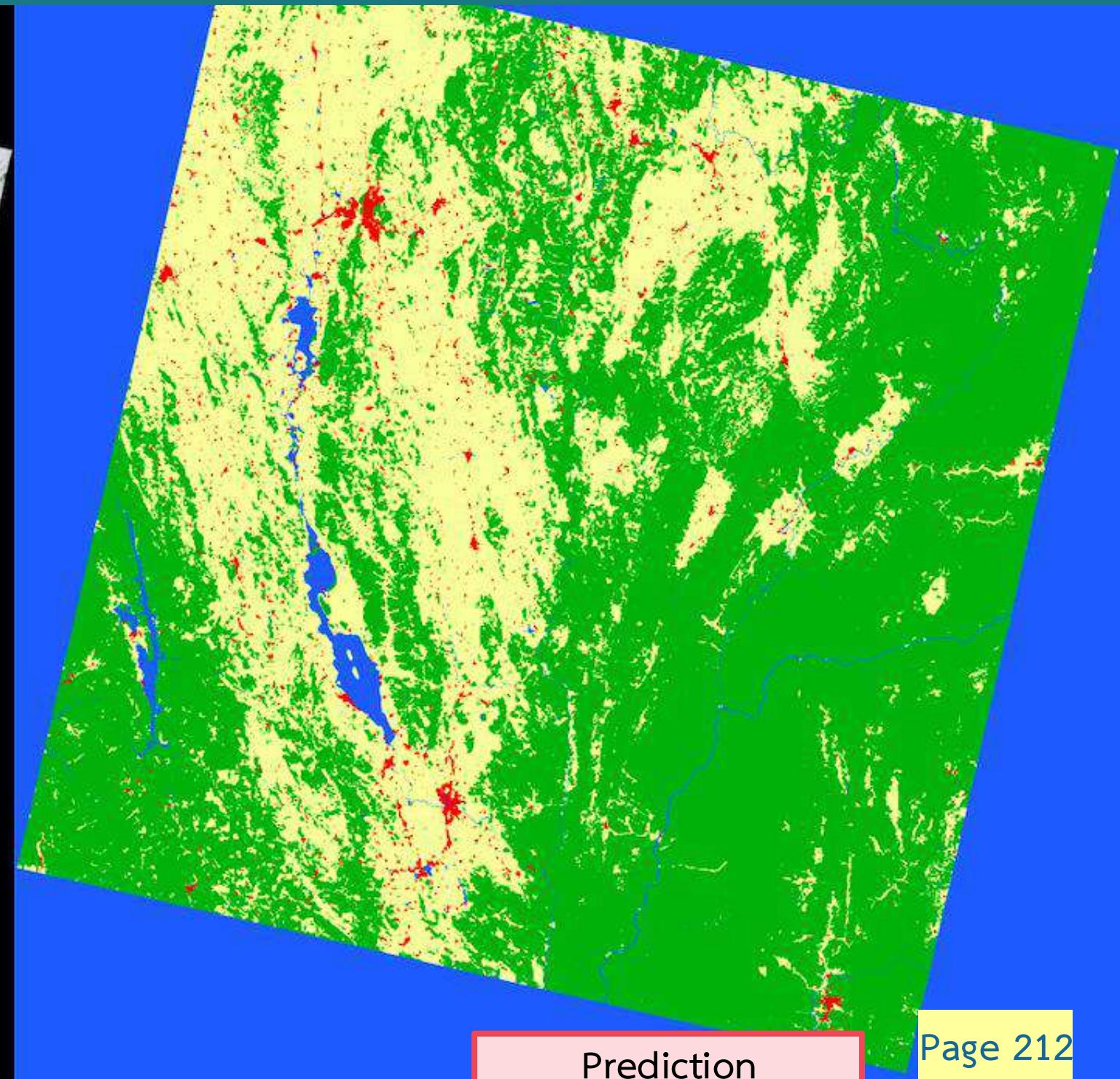
Satellite Image without ground truth



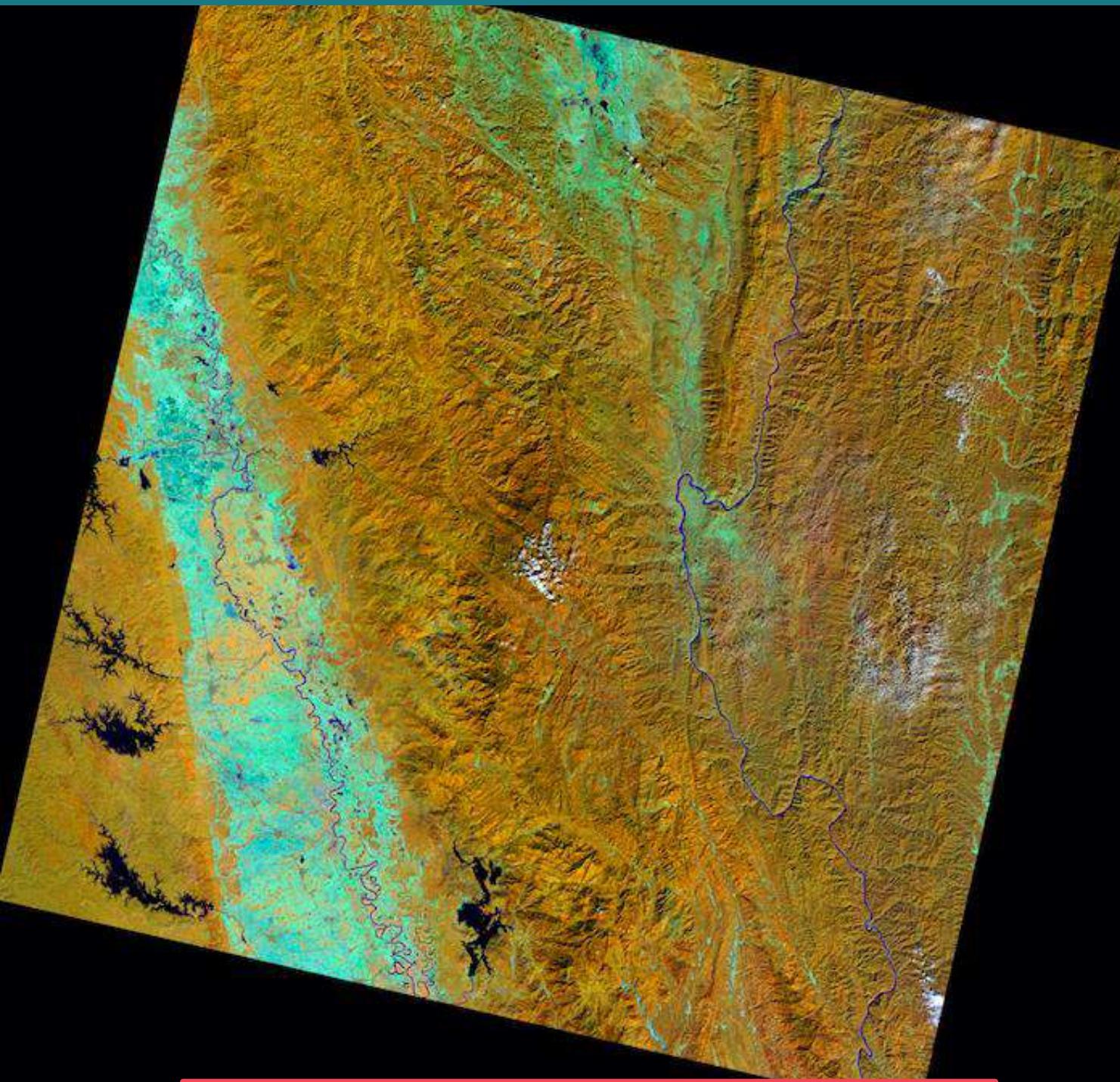
Prediction



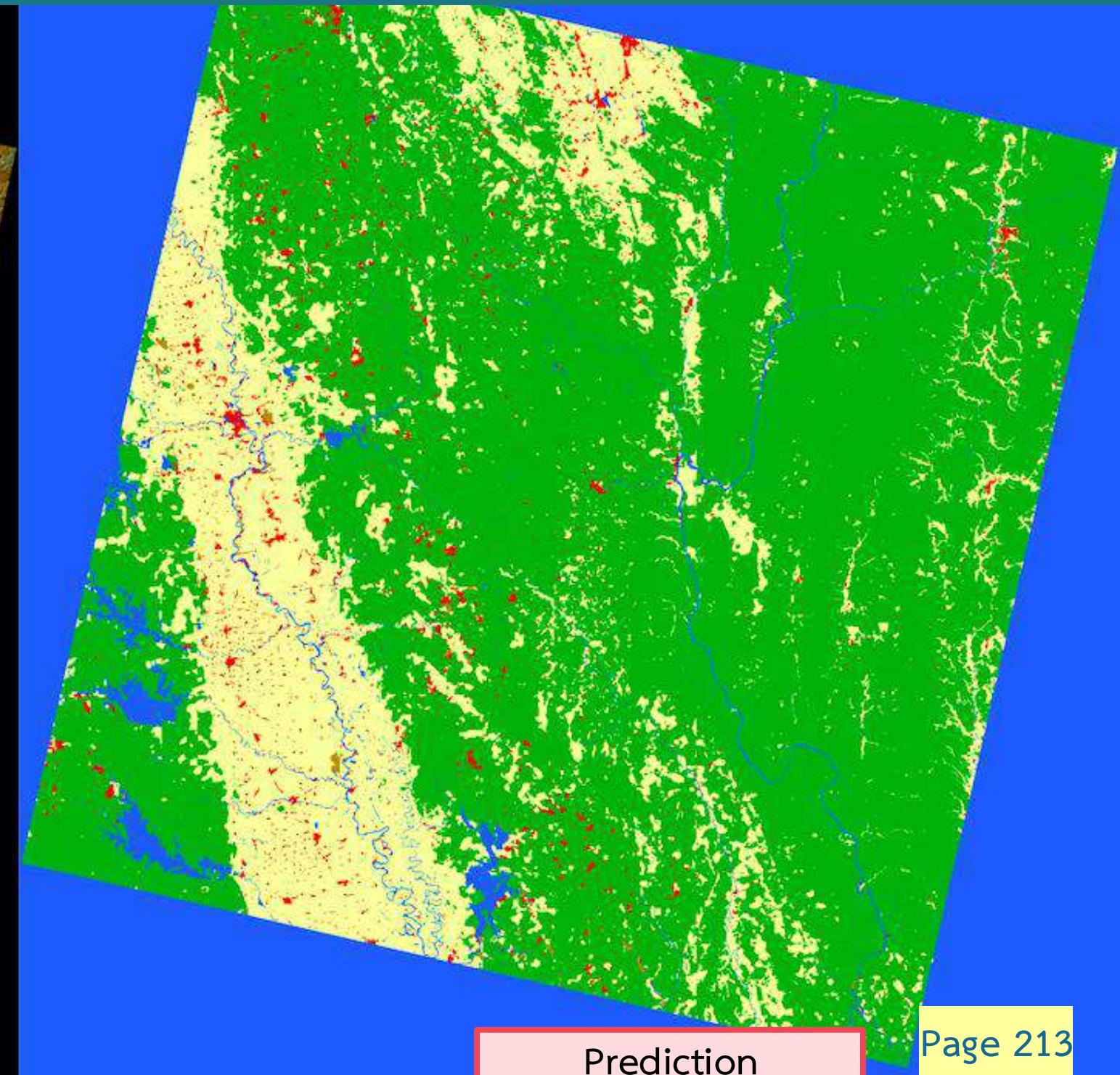
Satellite Image without ground truth



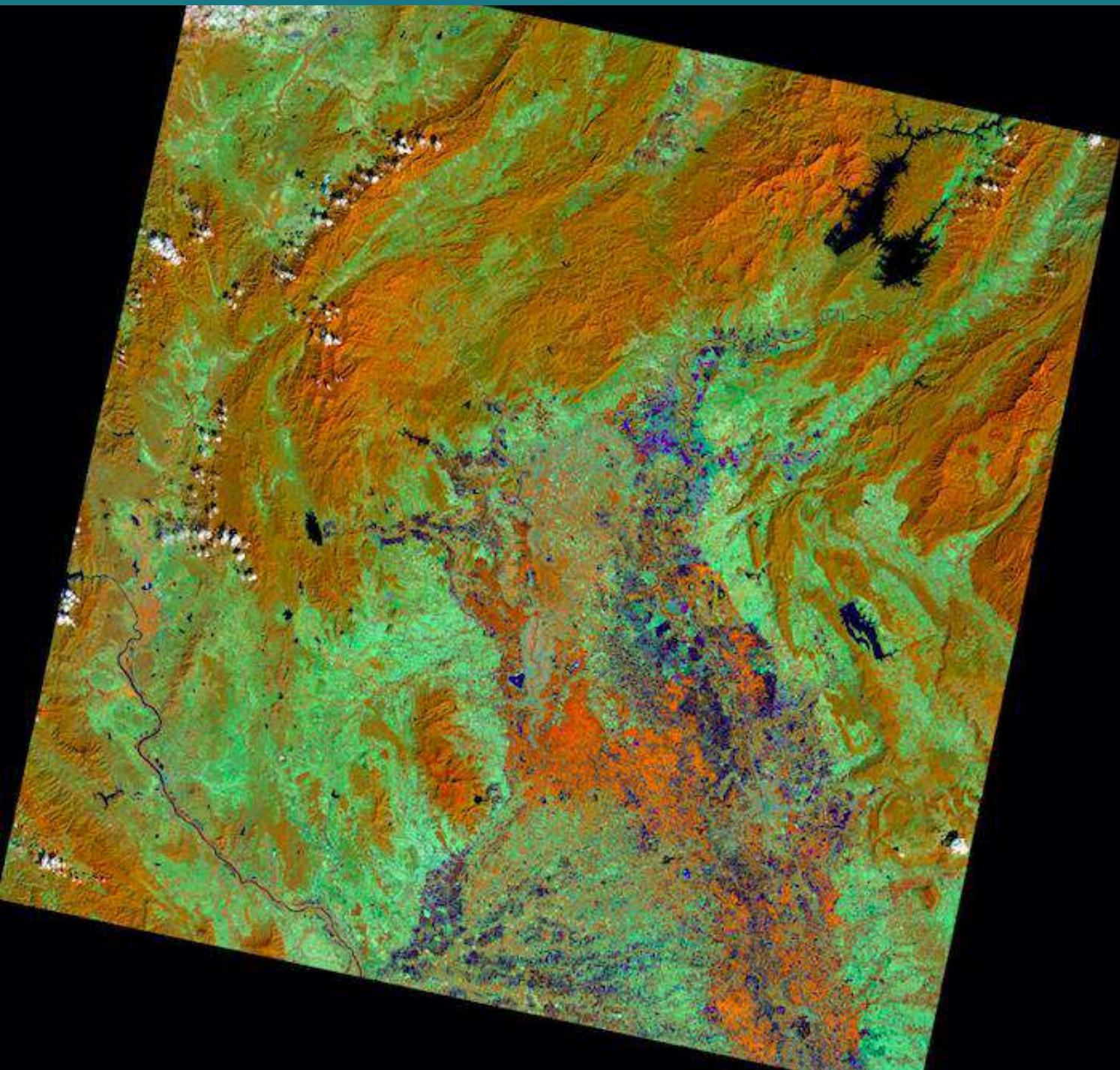
Prediction



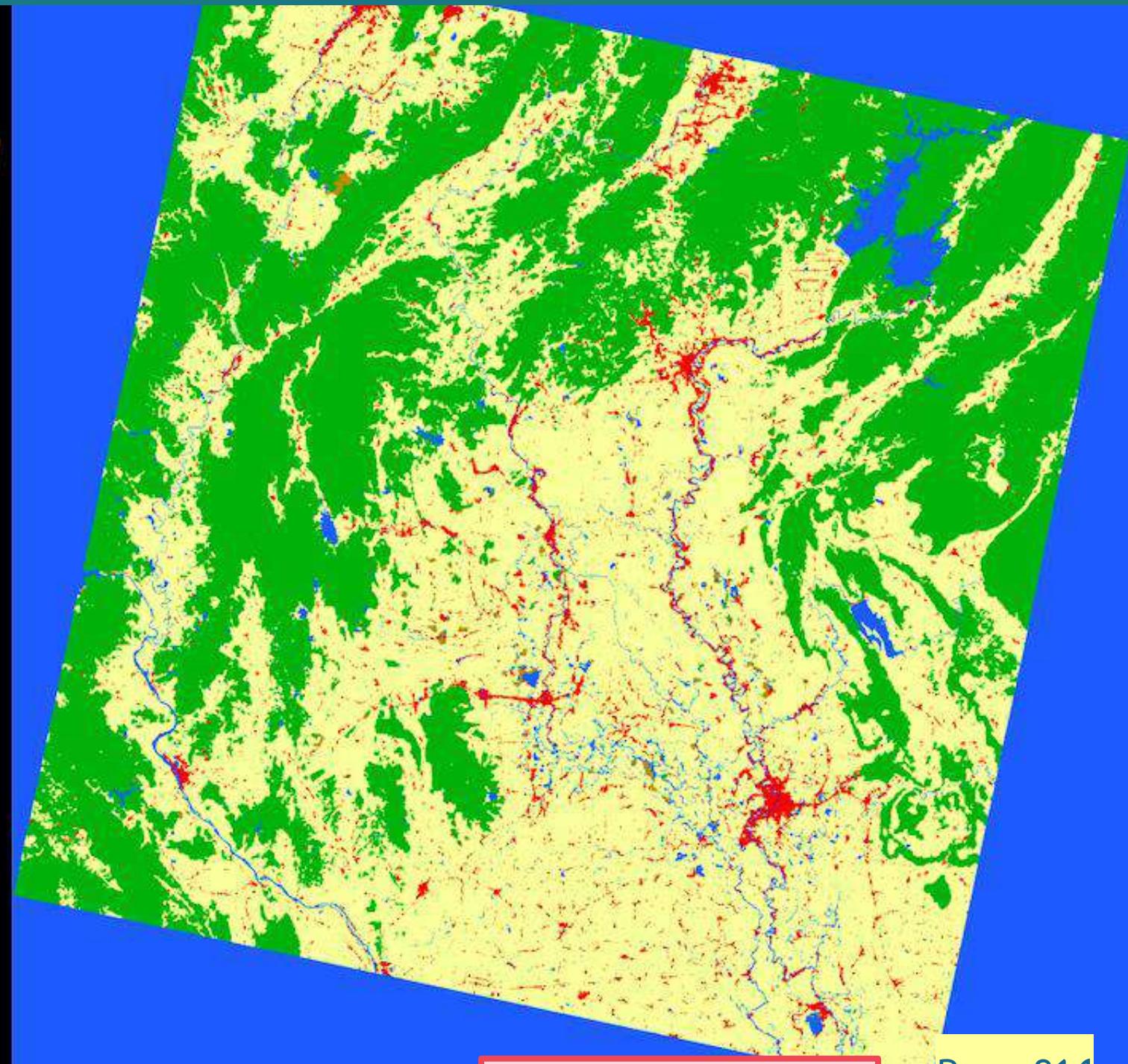
Satellite Image without ground truth



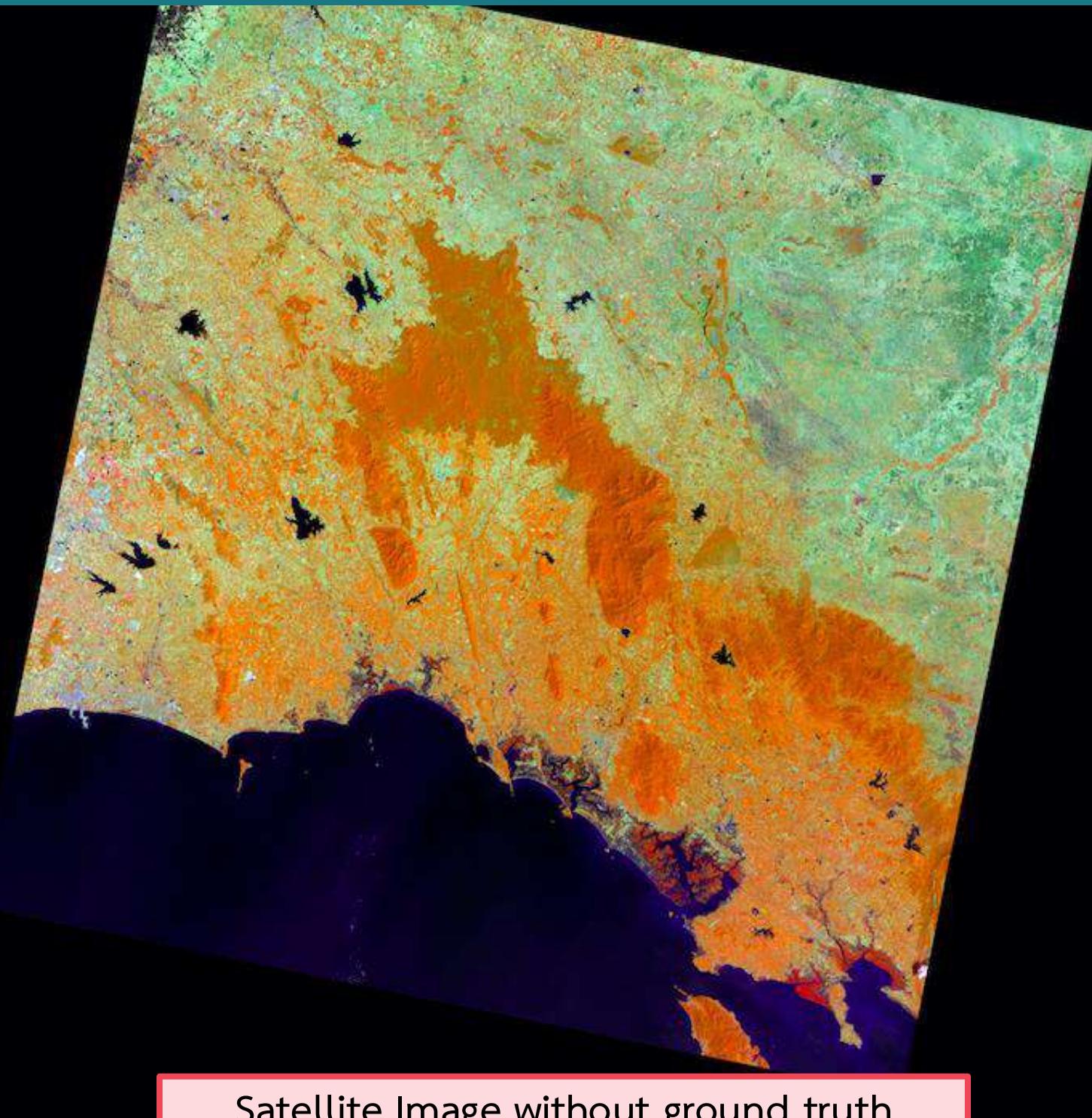
Prediction



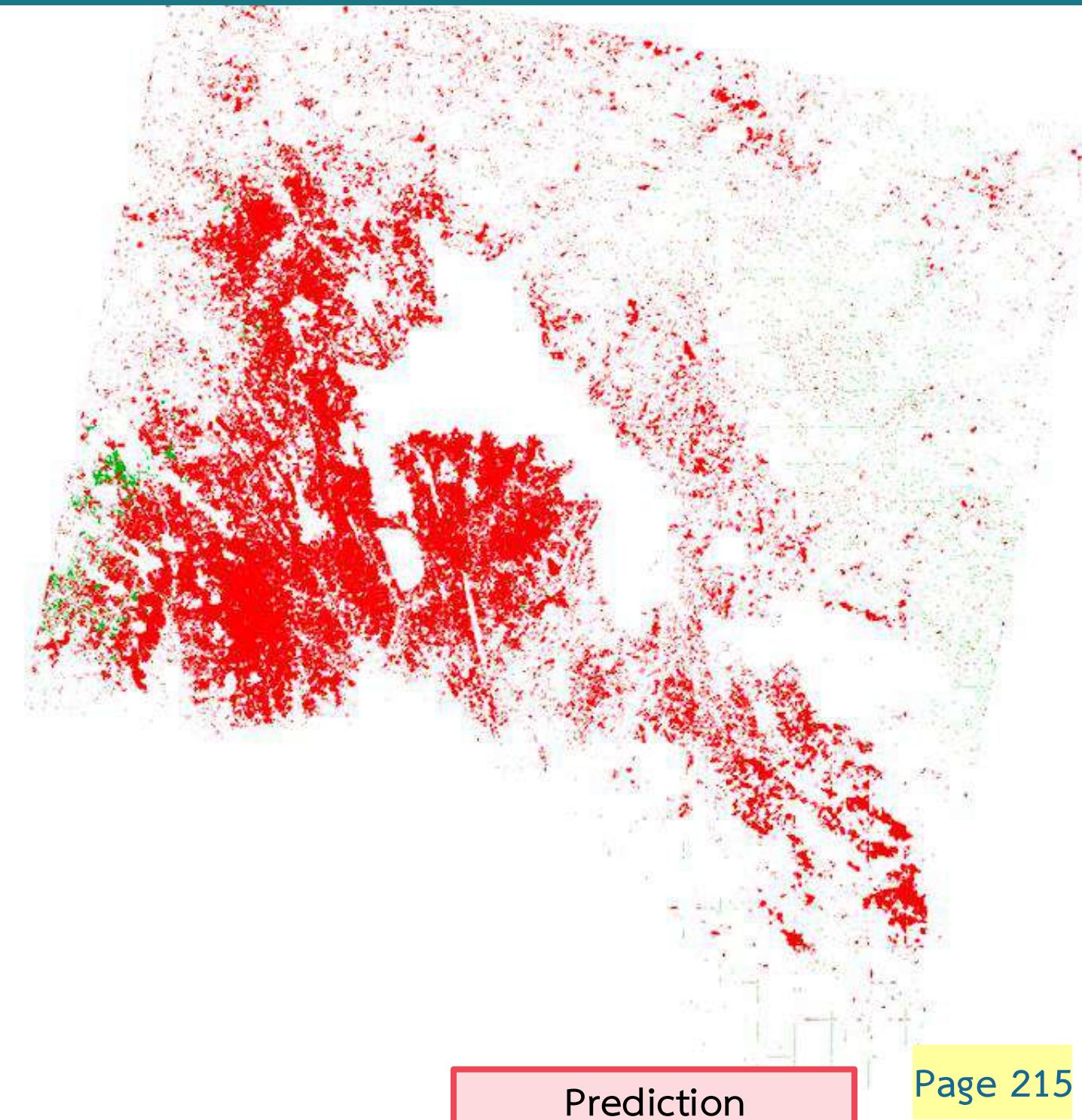
Satellite Image without ground truth



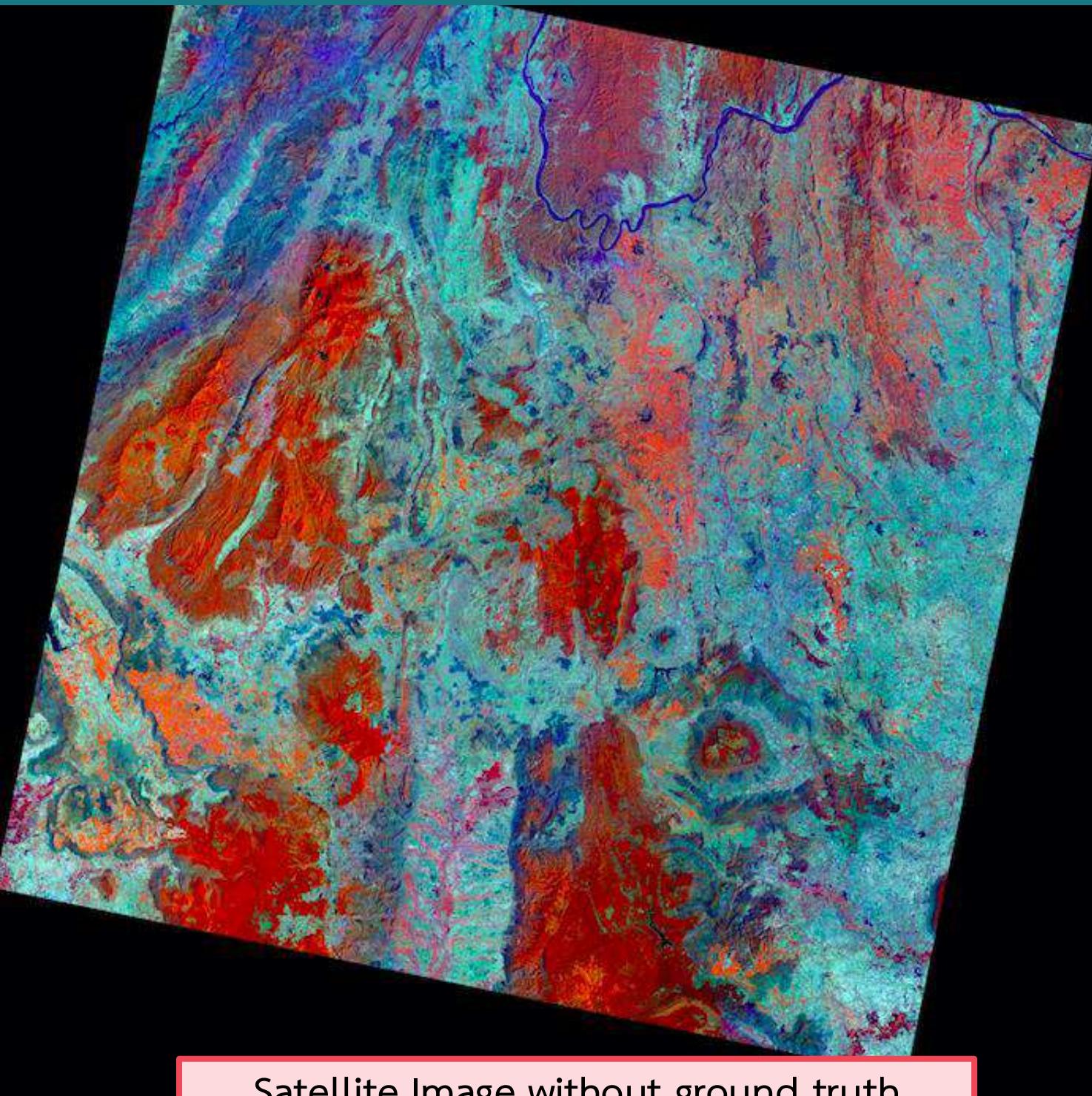
Prediction



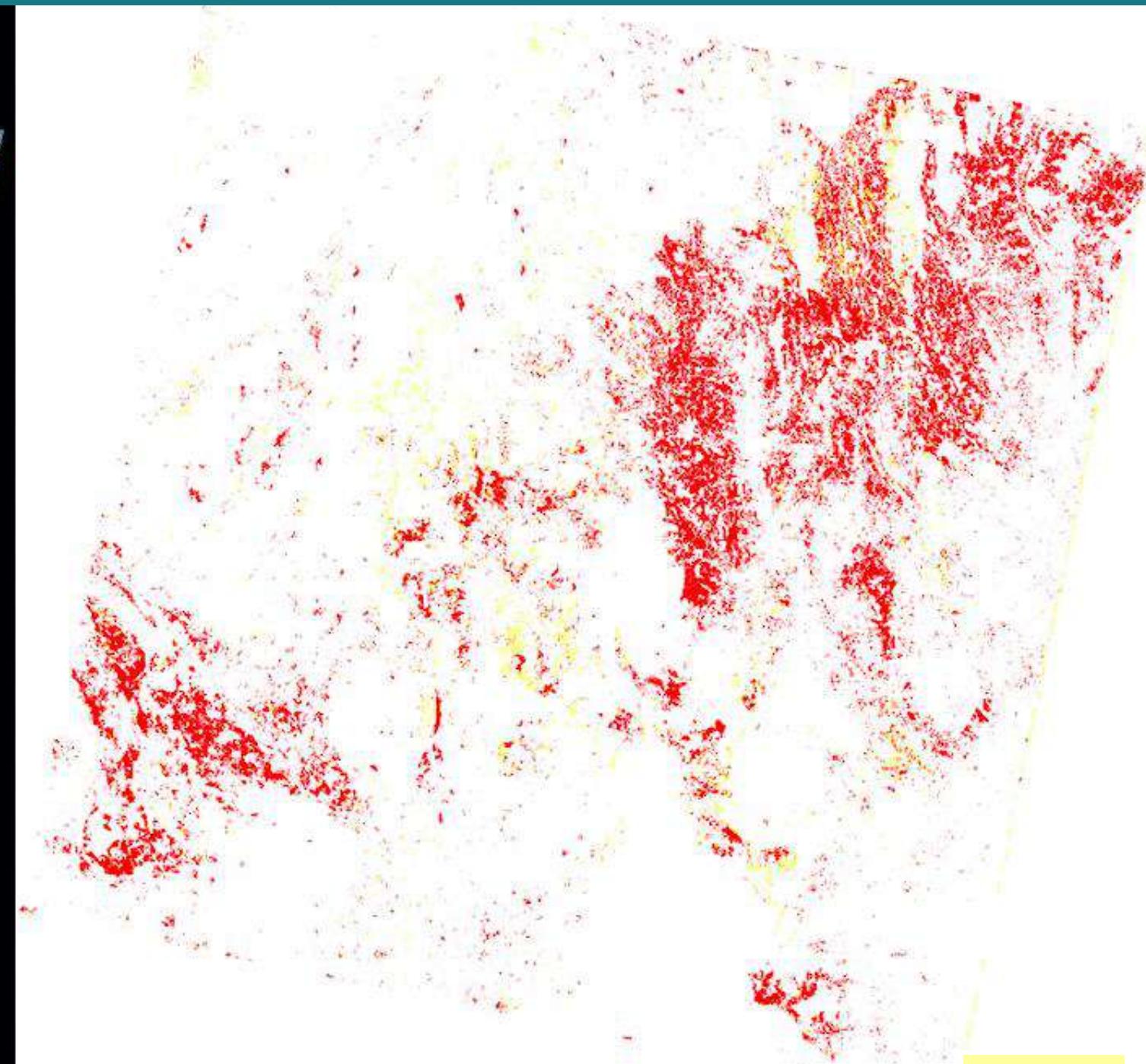
Satellite Image without ground truth



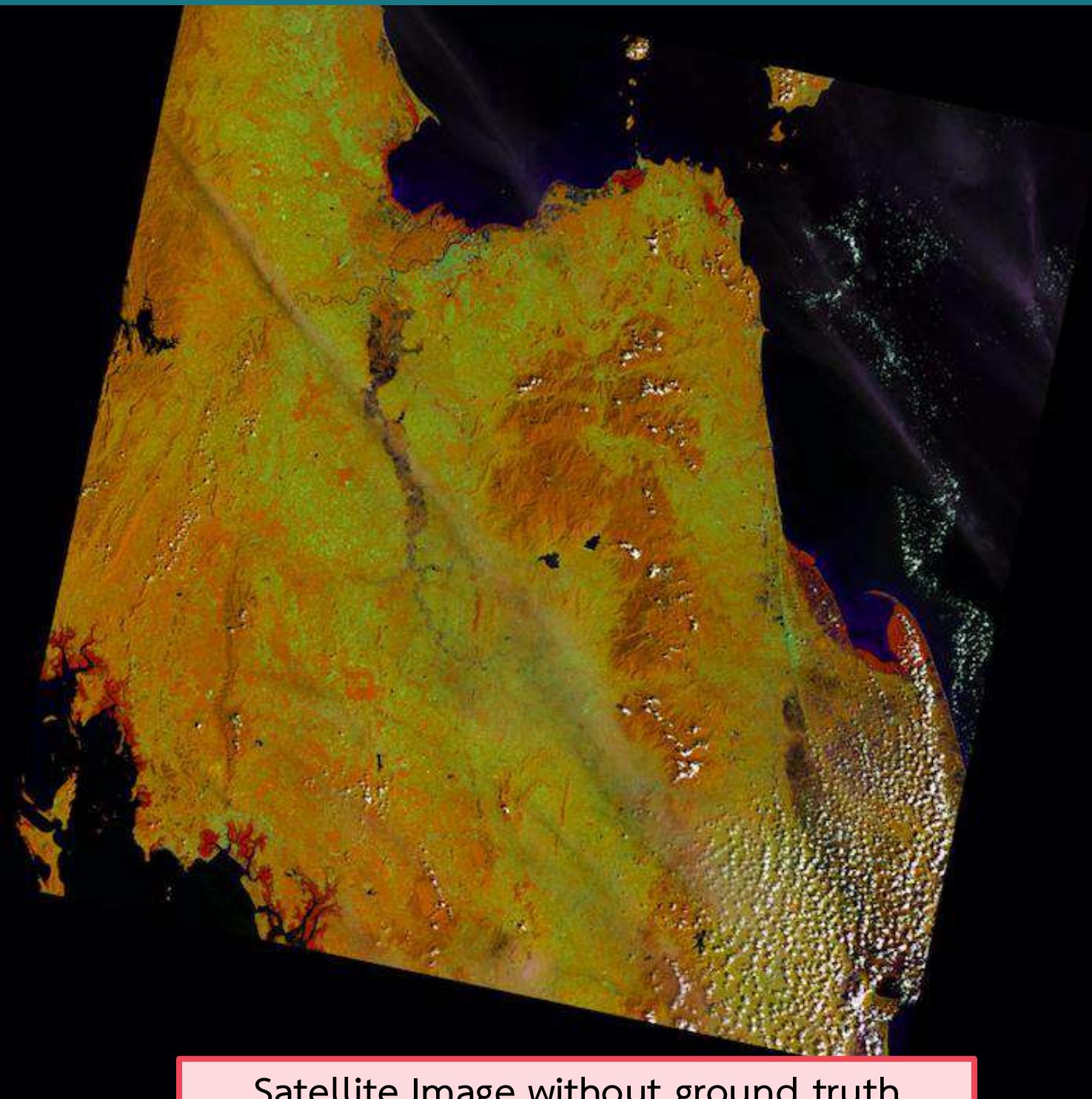
Prediction



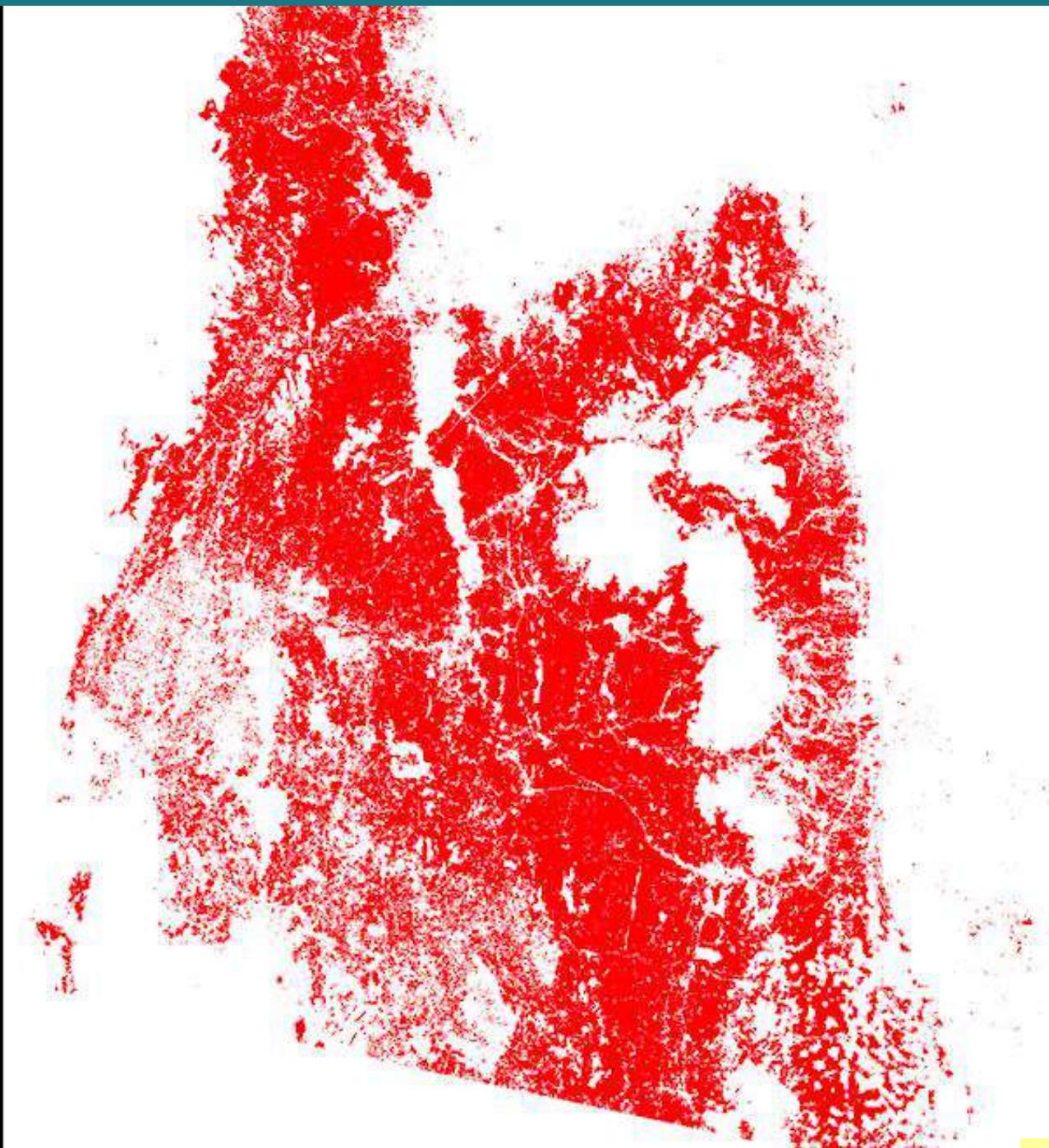
Satellite Image without ground truth



Prediction

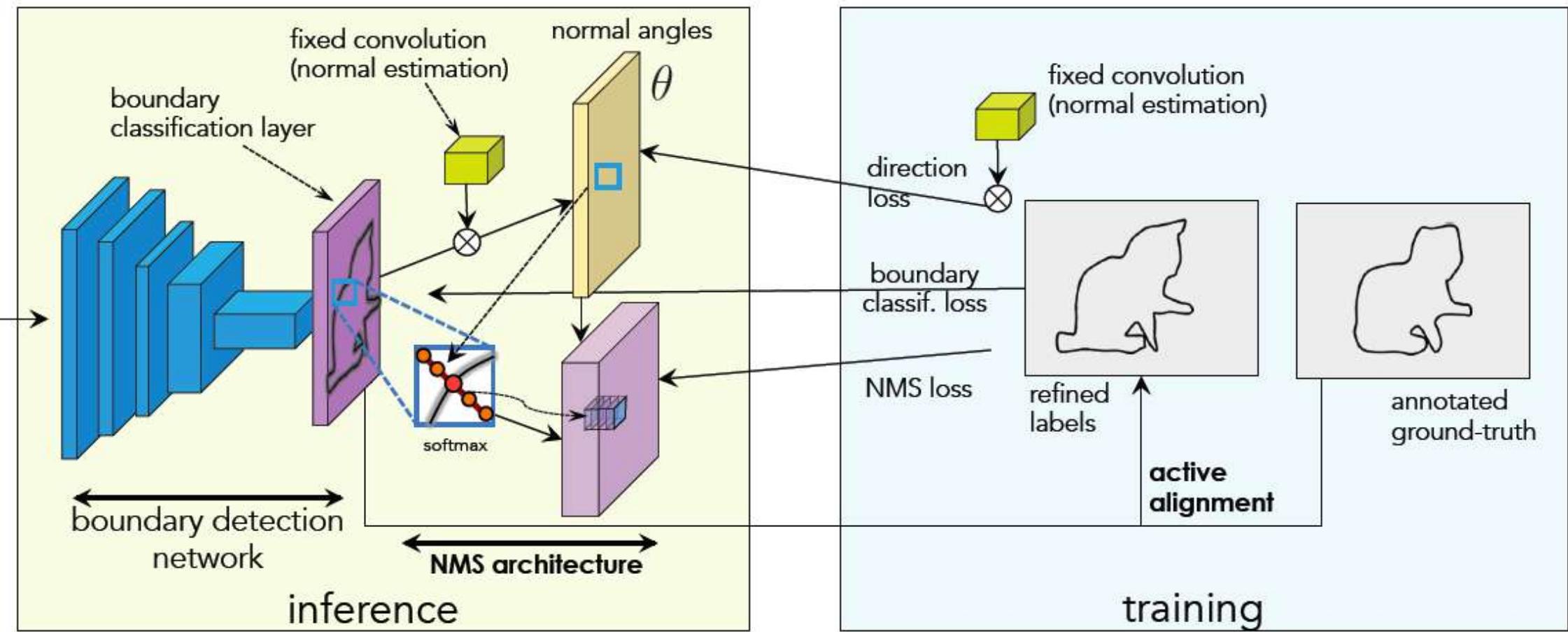


Satellite Image without ground truth

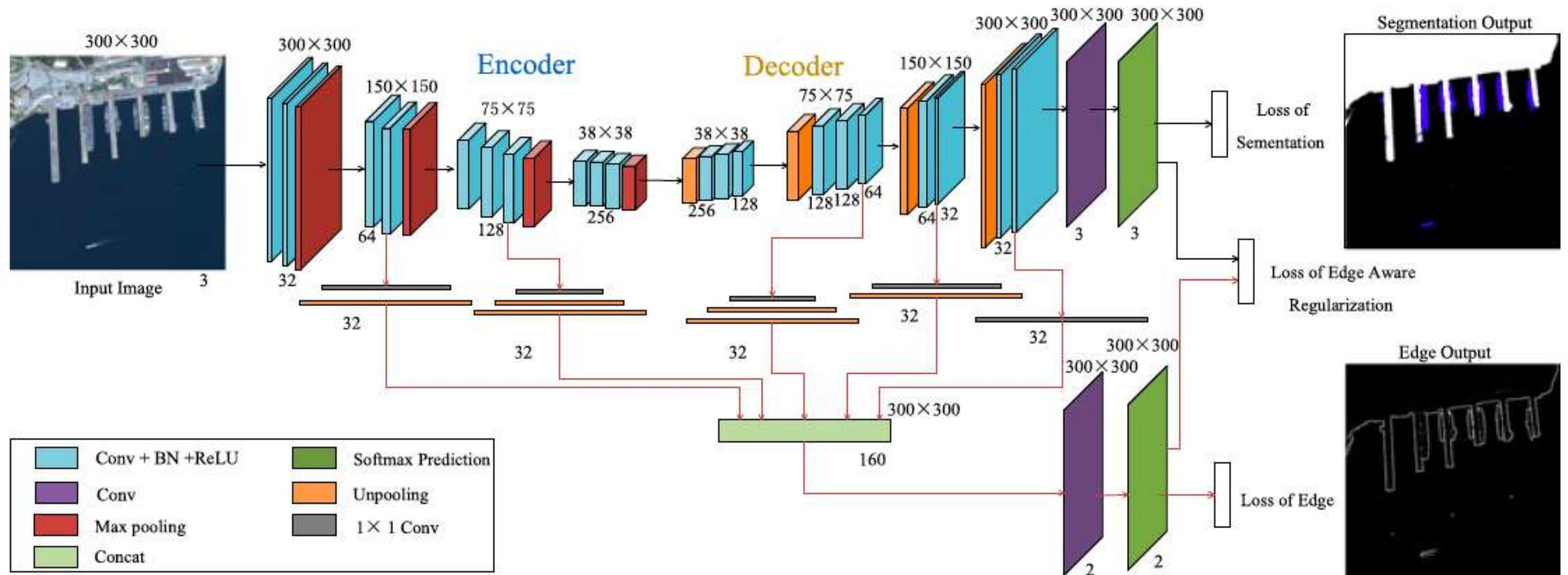


Prediction

# New Idea (1)



# New Idea (1)



# Outline | Related Theory

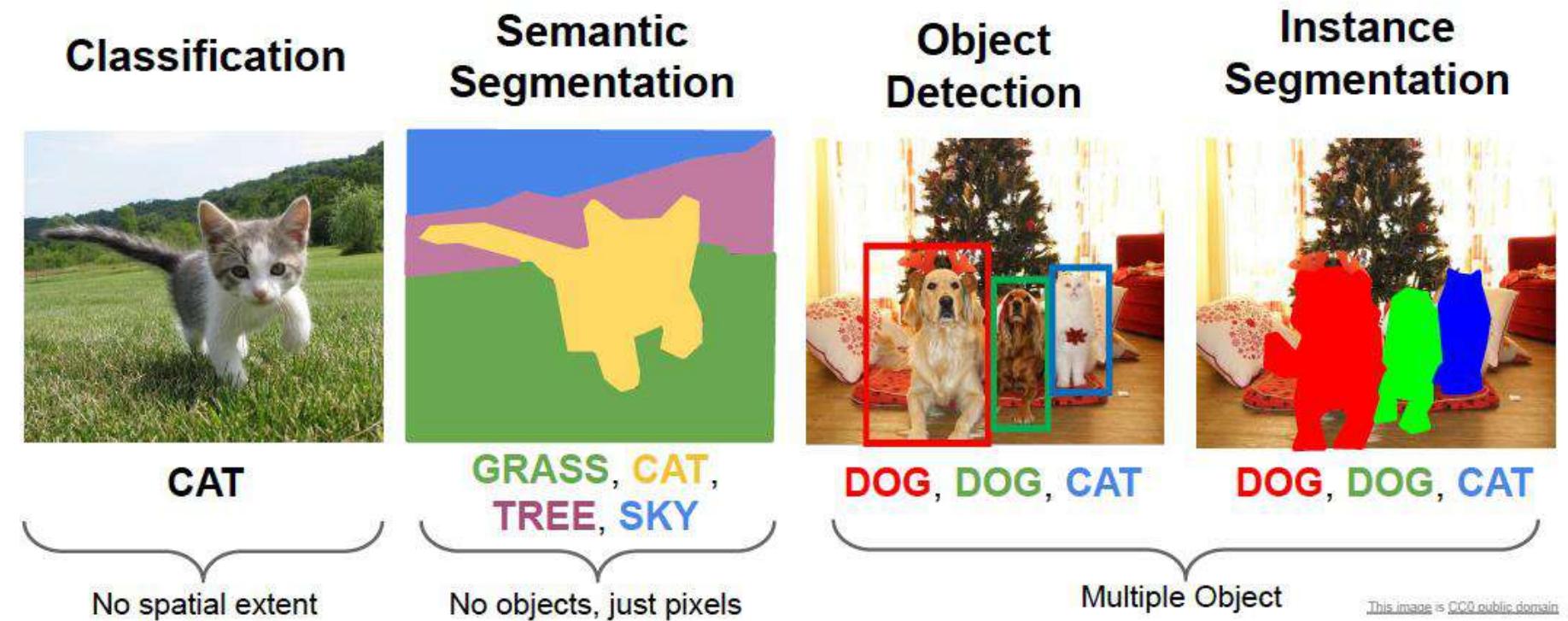
- Introduction
- **Related Theory**
- Related Works
- Methodology (Proposed Method)
- Experimental Results
- Objectives and Procedure
- Conclusions
- Publication and Reference

# Related Theory

- (1) Computer Vision Tasks
- (2) Deep Convolutional Neural Networks (CNNs)
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

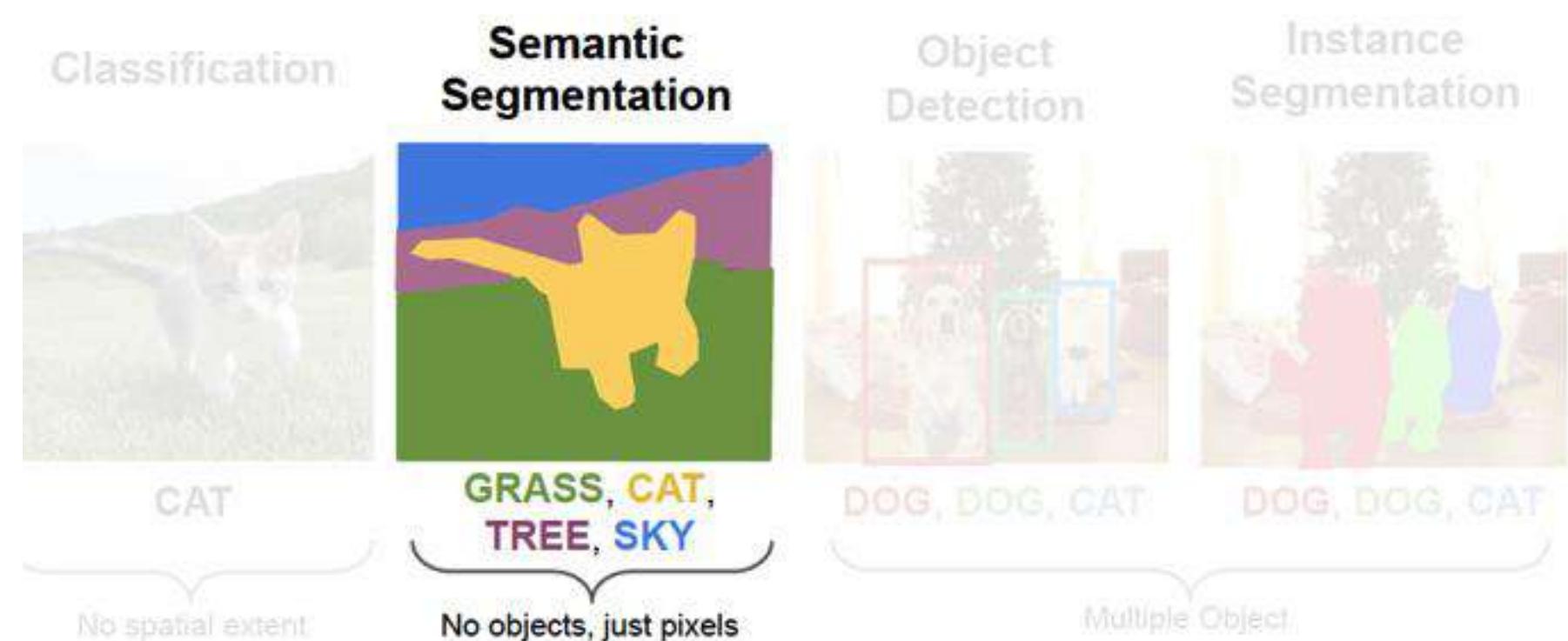
# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



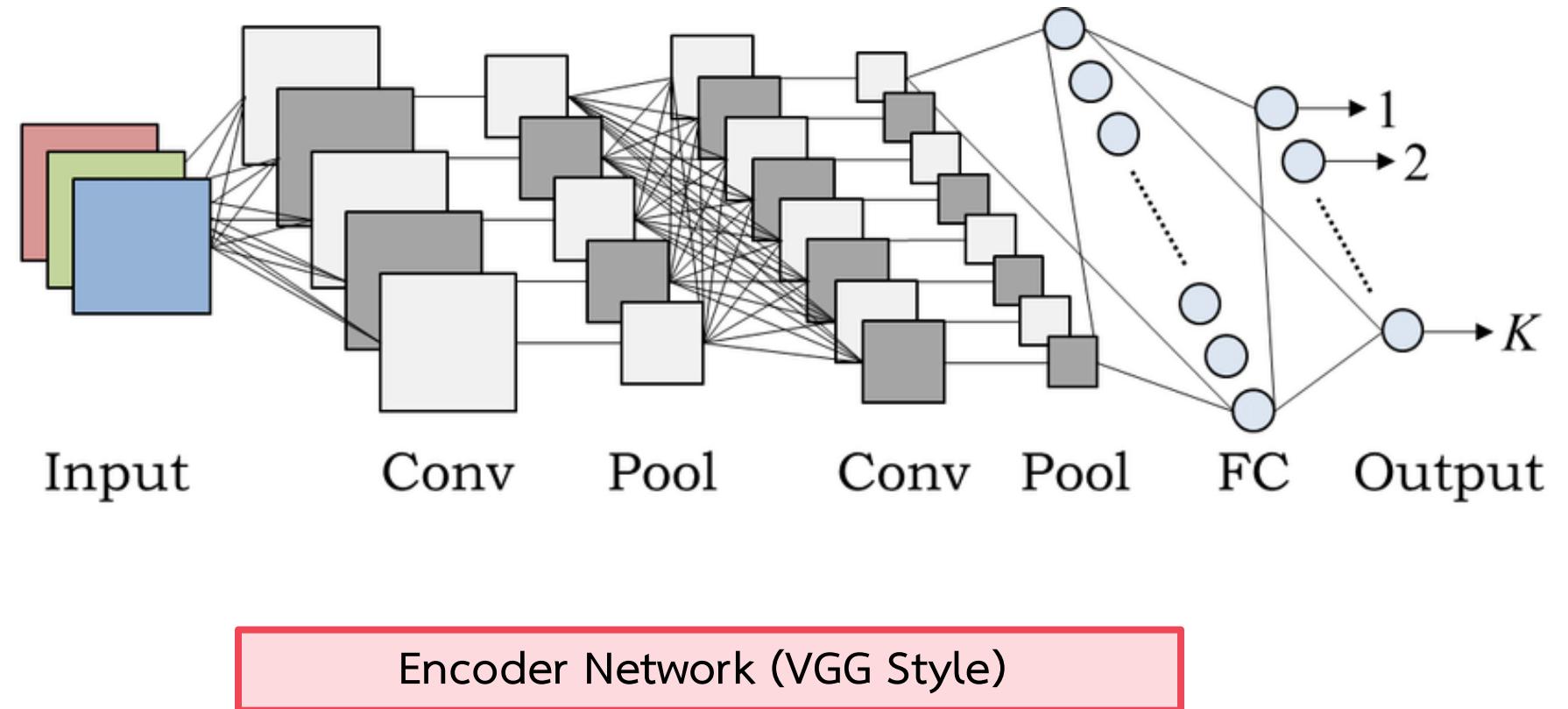
# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



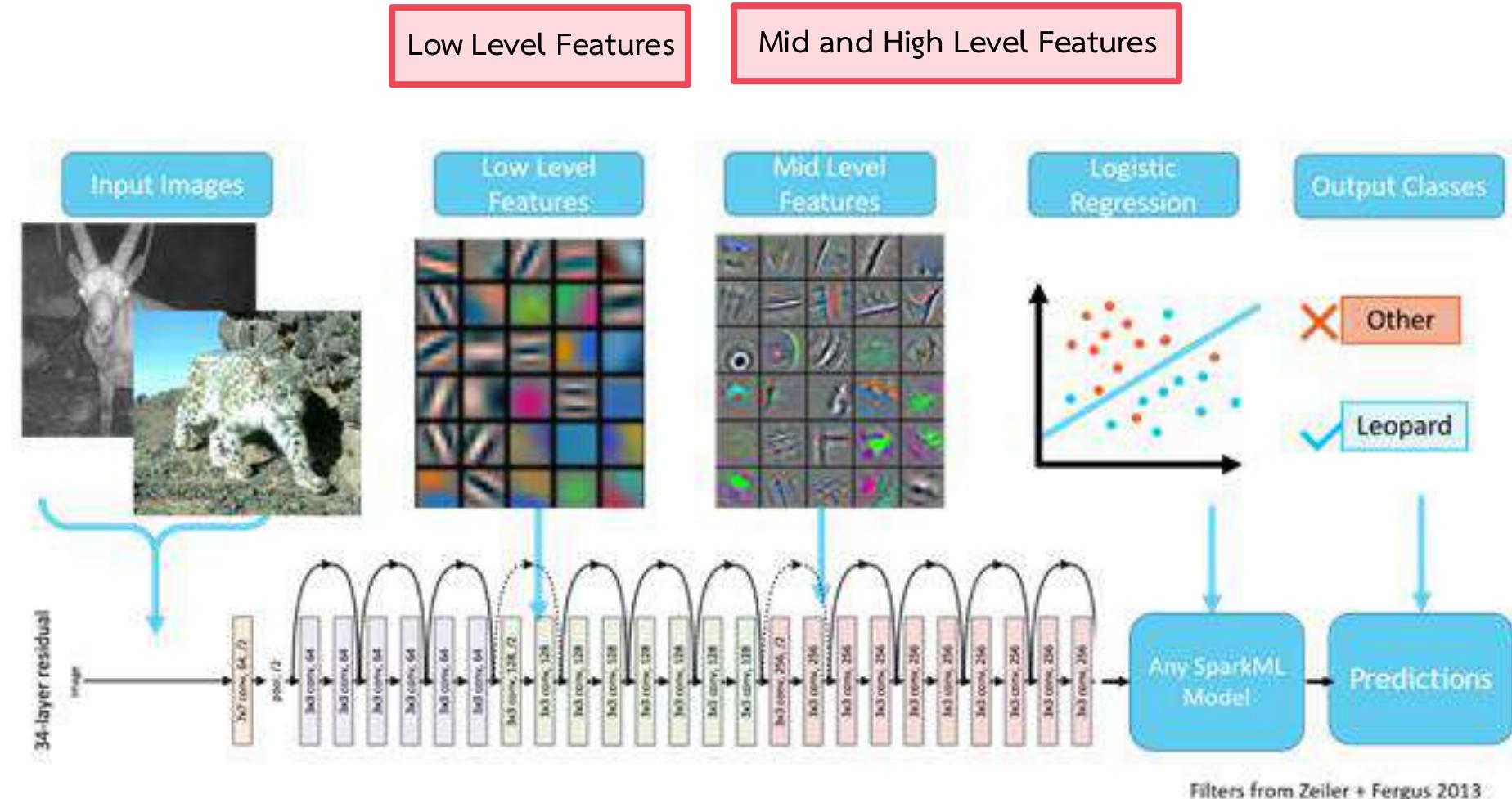
# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

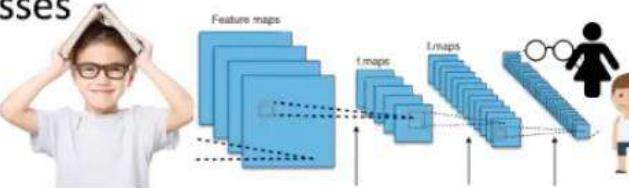


# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

Each feature can be discovered without the need for seeing the exponentially large number of configurations of the other features

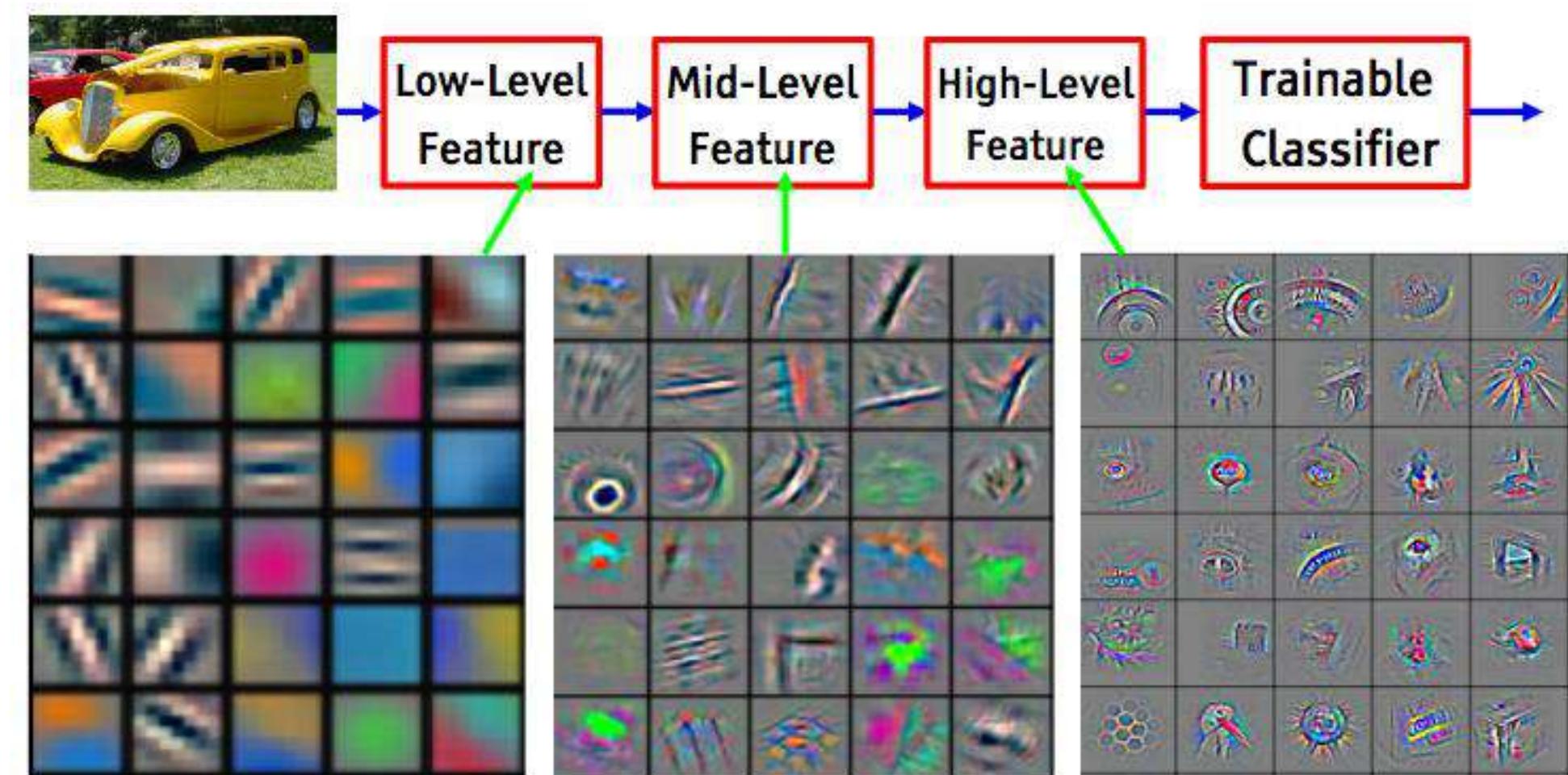
- Consider a network whose hidden units discover the following features:
    - Person wears glasses
    - Person is female
    - Person is a child
    - Etc.
- If each of  $n$  feature requires  $O(k)$  parameters, need  $O(nk)$  examples
- Parallel composition of features: can be exponentially advantageous
- Non-distributed non-parametric methods would require  $O(n^d)$  examples



zoom

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



# Related Theory

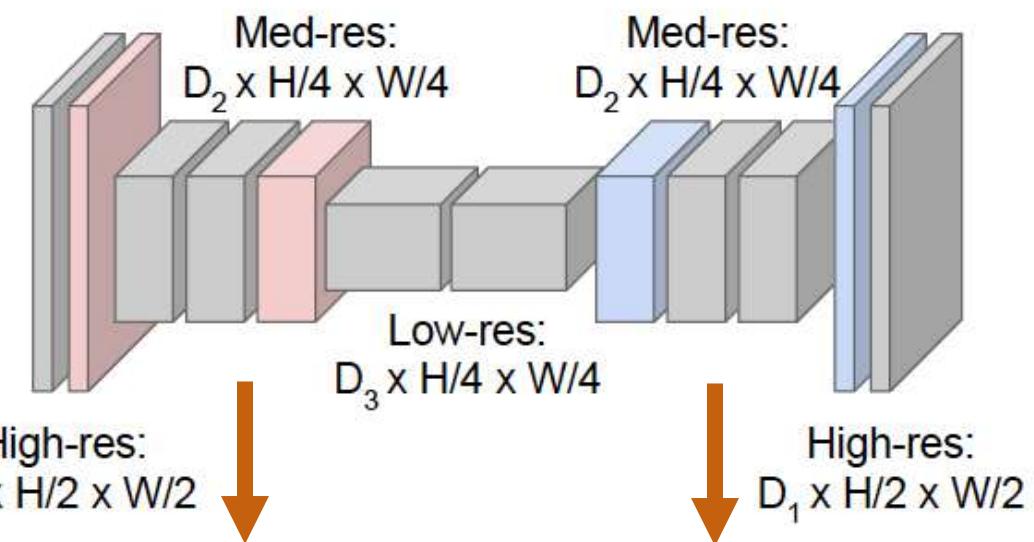
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

**Downsampling:**  
Pooling, strided convolution



Input:  
 $3 \times H \times W$

Design network as a bunch of convolutional layers, with  
**downsampling** and **upsampling** inside the network!



Encoder Network

Decoder Network

Deep Encoder-Decoder Network (DCED)

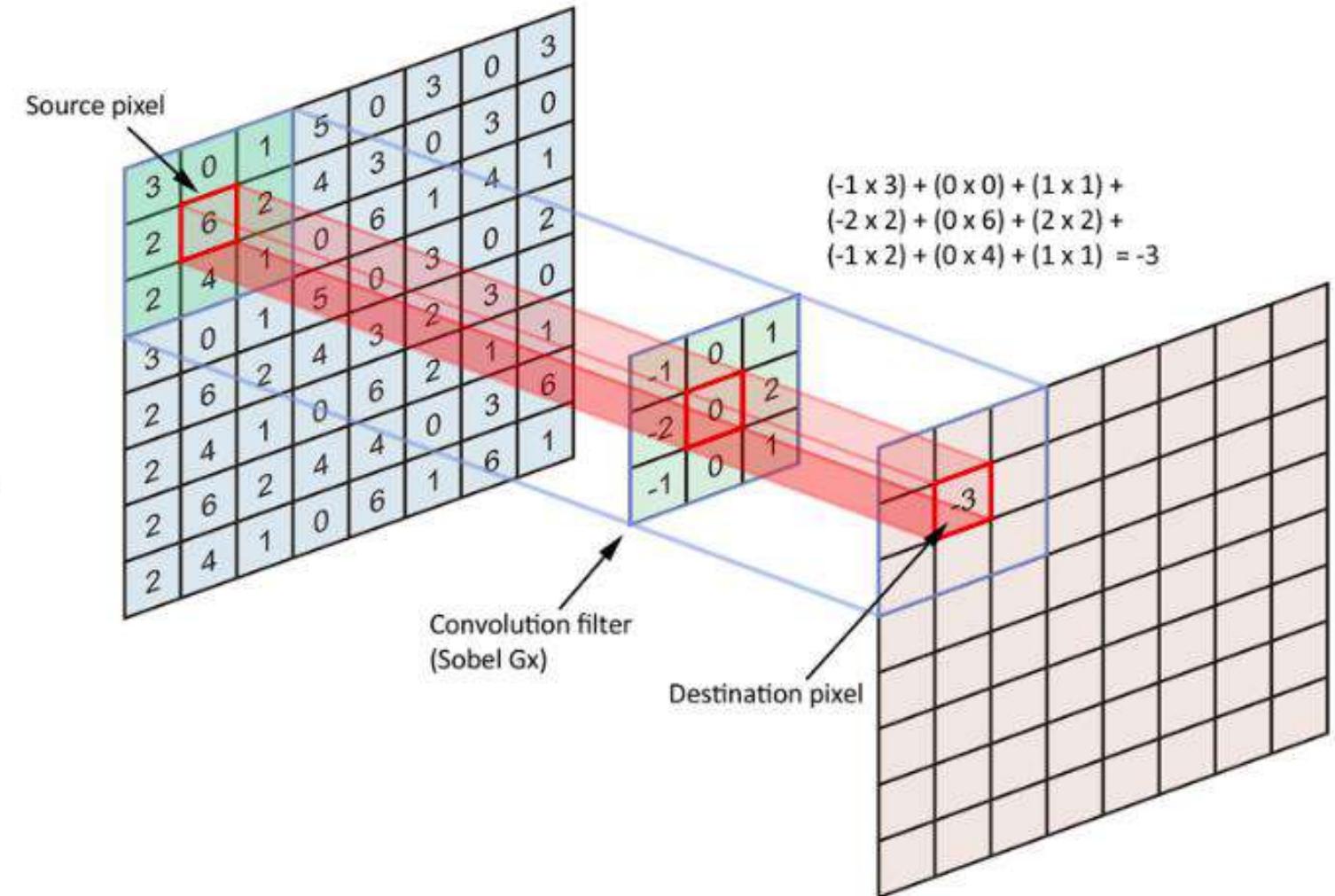
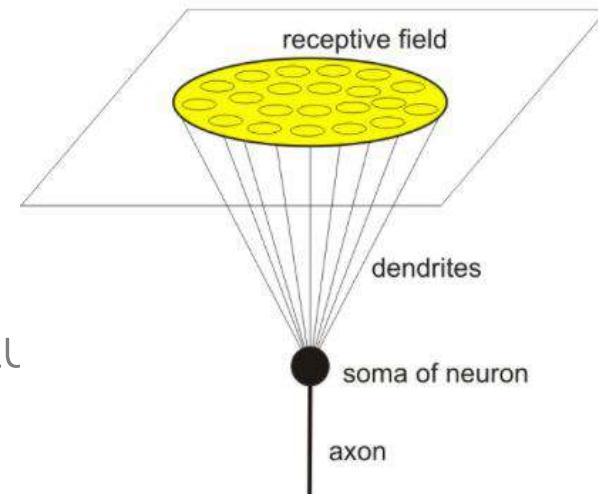
**Upsampling:**  
Unpooling or strided transpose convolution



Predictions:  
 $H \times W$

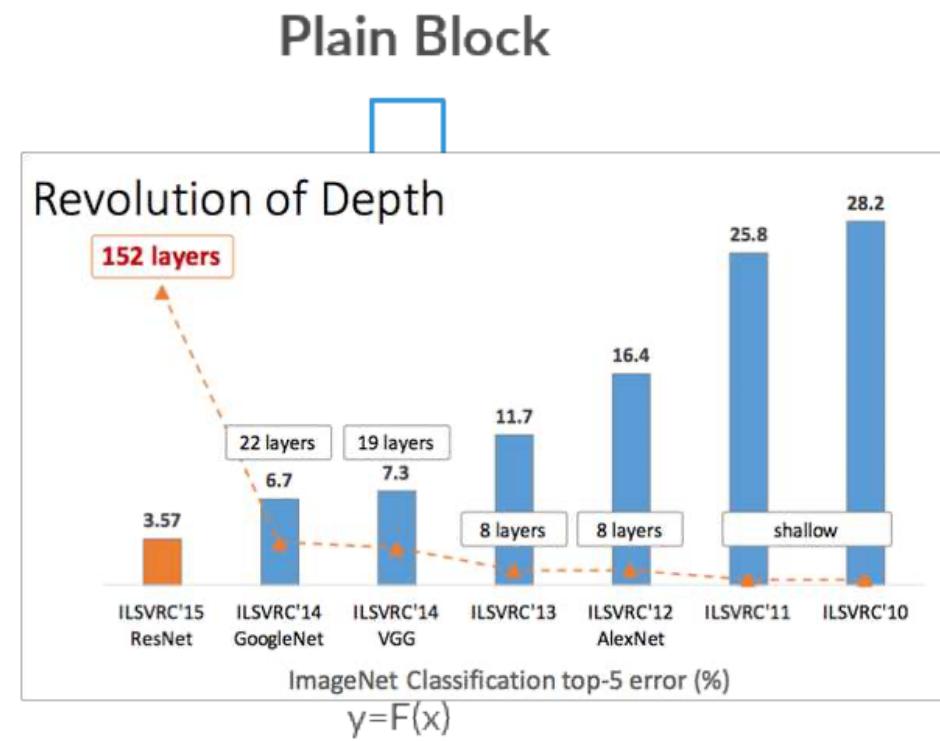
# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

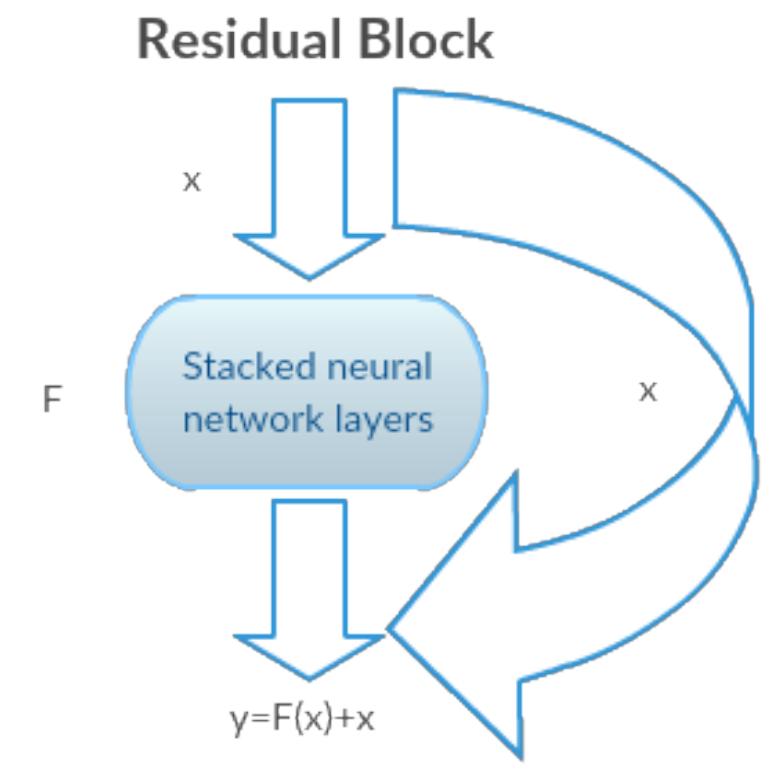


# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



Hard to get  $F(x)=x$  and make  $y=x$   
an identity mapping



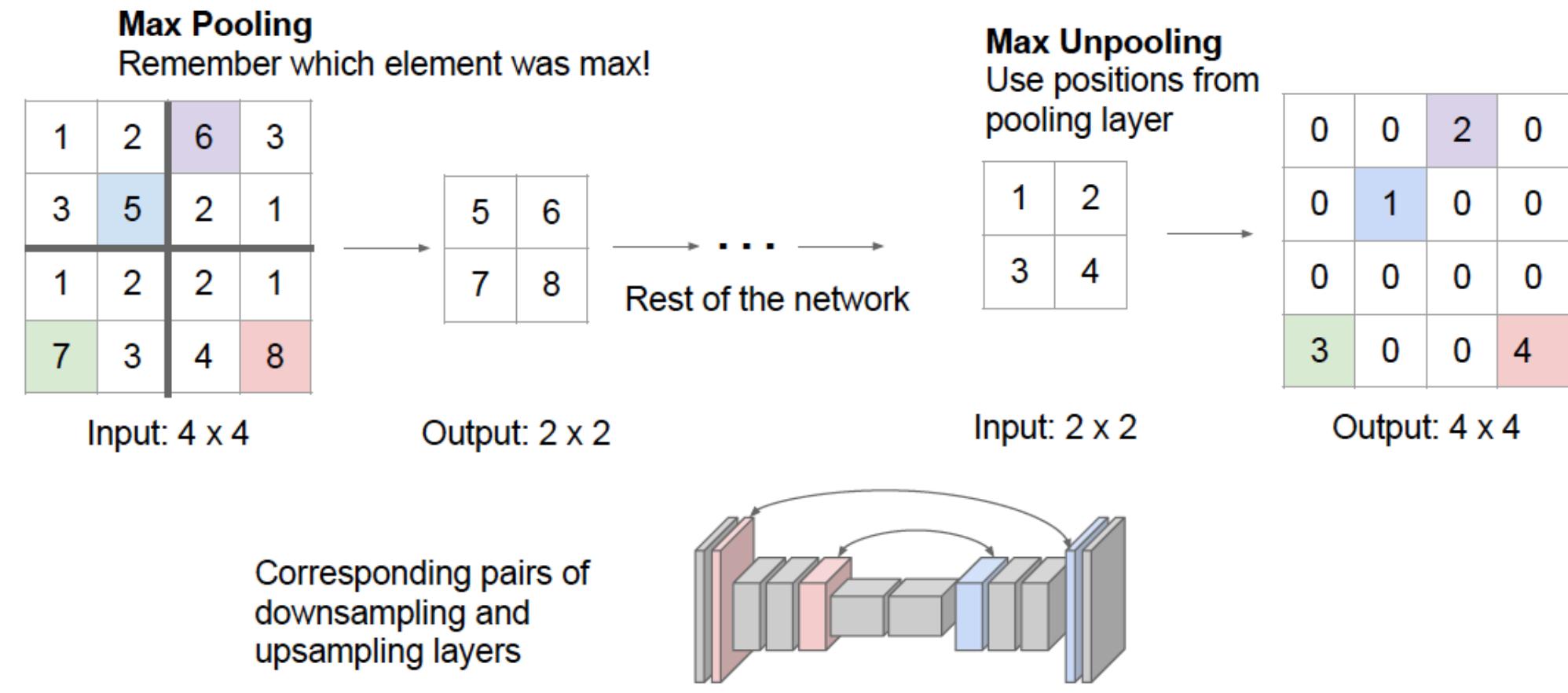
Easy to get  $F(x)=0$  and make  $y=x$   
an identity mapping

**Encoder Network (VGG (Residual) Style)**

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

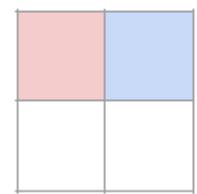
## Max-Pooling and Max-Unpooling Layer



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

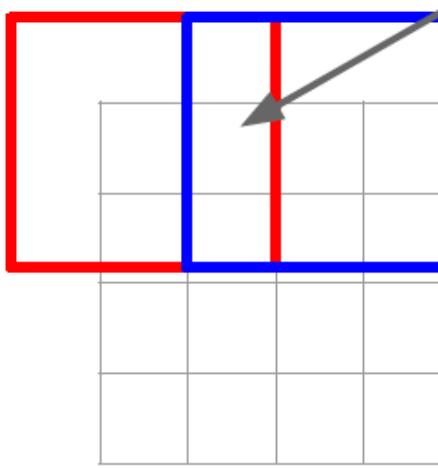
**Other names:**  
-Deconvolution (bad)  
-Upconvolution  
-Fractionally strided convolution  
-Backward strided convolution



Input: 2 x 2

3 x 3 transpose convolution, stride 2 pad 1

Input gives weight for filter



Output: 4 x 4

Sum where output overlaps

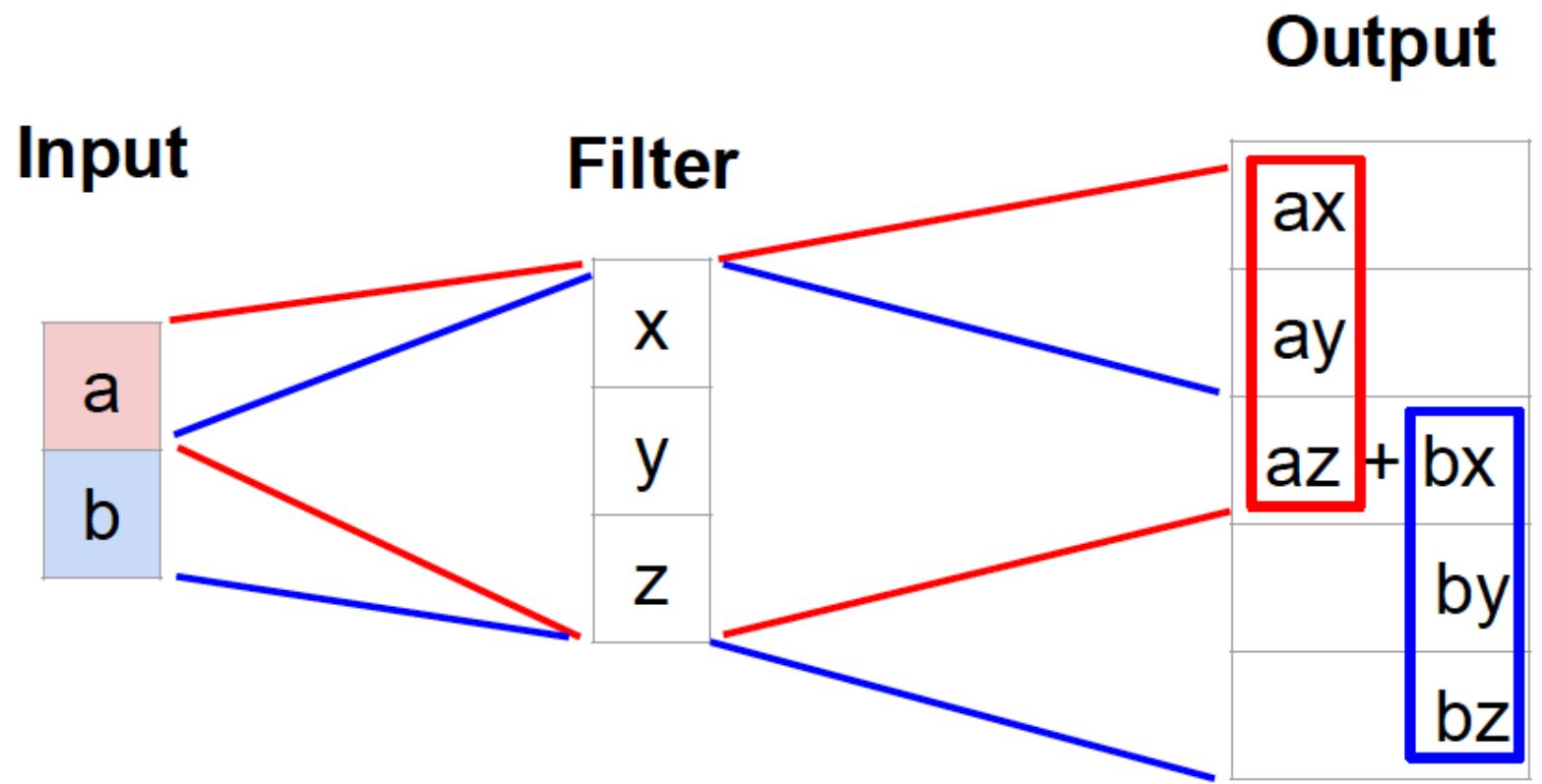
Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

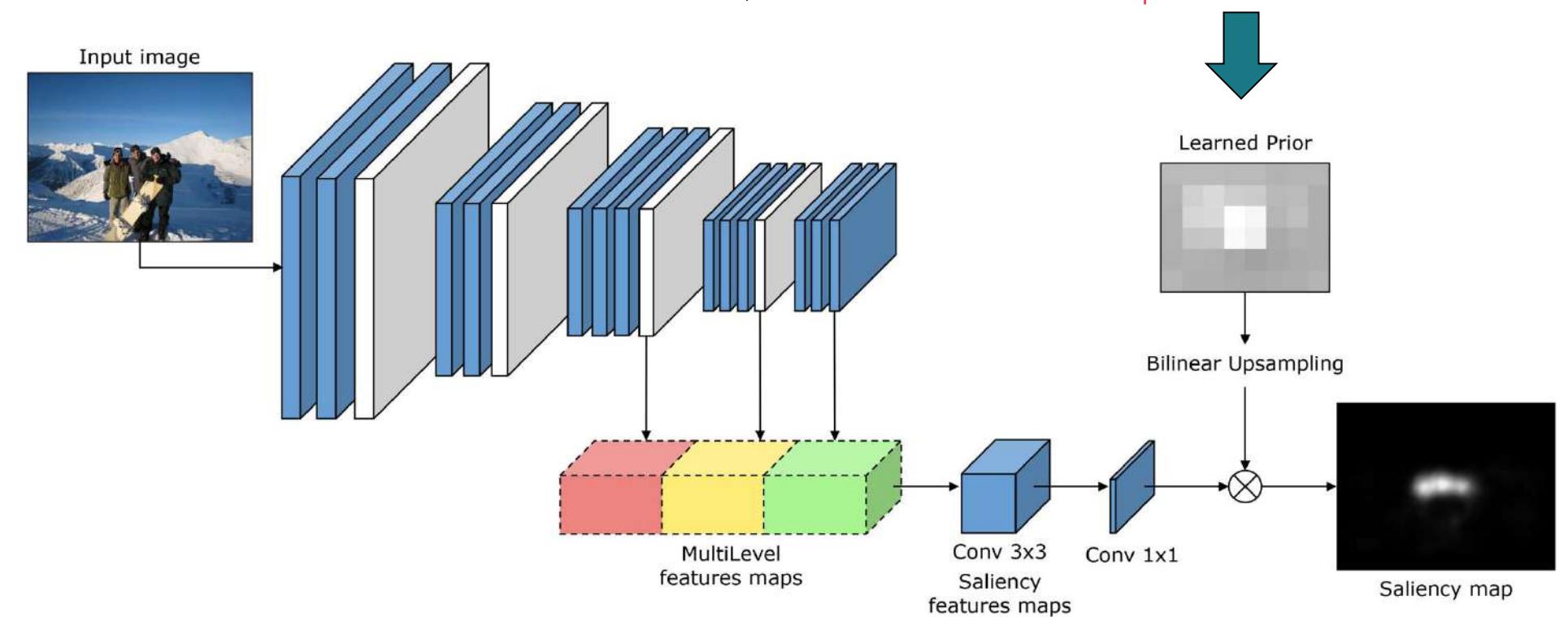
Learnable Up-sampling: Transpose Convolution



# Related Theory

Other Layers

- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - (4) Channel Attention
  - (5) Feature Fusion
  - (6) Depthwise Convolution
  - (7) Design CNNs
- Interpolation Layer: Interpolation layer
    - performs resizing operation along the spatial dimension.
    - In our network, we use **bilinear interpolation**.



# Related Theory

Other Layers

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs
- Elementwise Layer: Elementwise layer
  - performs elementwise operations on two or more previous layers, in which the feature maps must be of the same number of channels and the same size.
  - There are three kinds of elementwise operations:
    - product, add (sum), max.
    - In our network, we use add operation.

# Related Theory

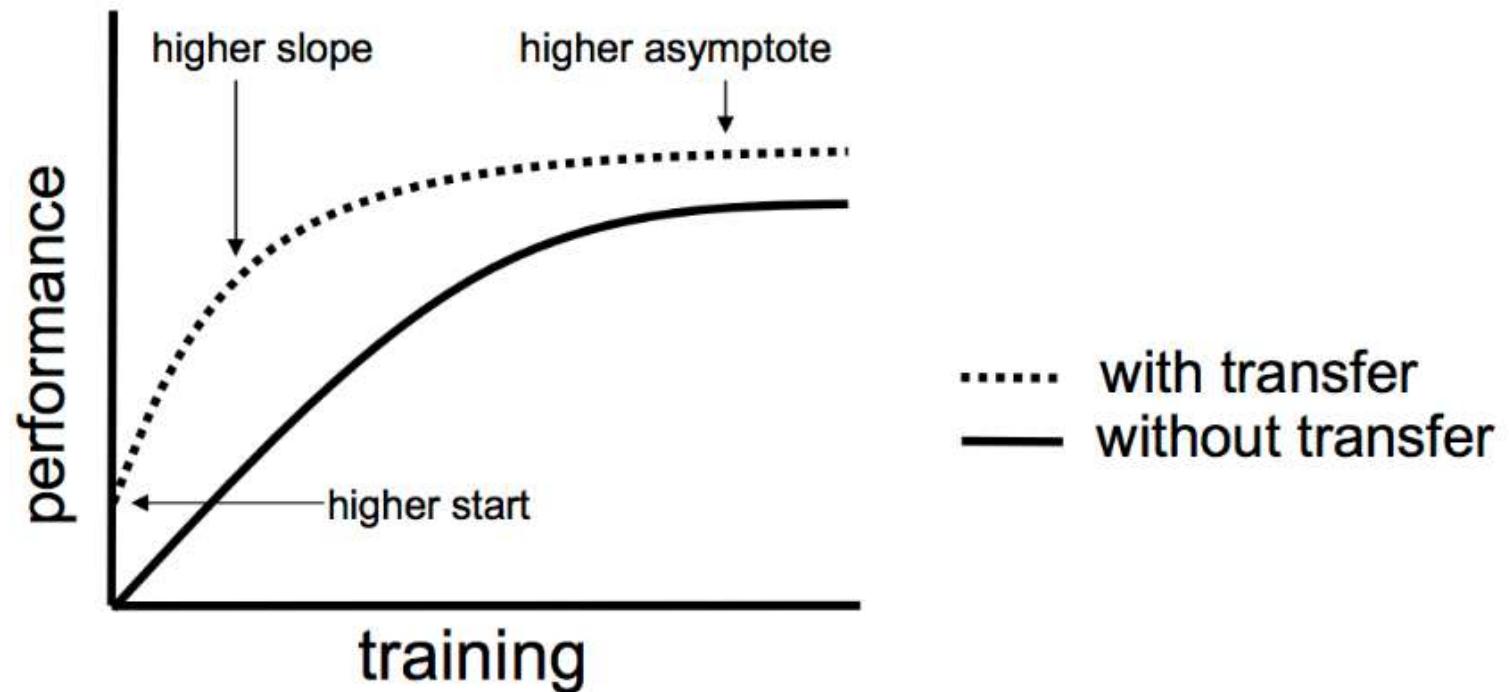
Other Layers

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs
- ReLU Layer: The rectified linear unit (ReLU) (Hinton, 2010)
  - It is usually chosen as the nonlinearity layer
  - It thresholds the non-positive value as zero and keeps the positive value unchanged
  - It can achieve a considerable reduction in training time
- Batch Normalization Layer:
  - It normalizes layer inputs to a Gaussian distribution with zero-mean and unit variance.
  - Aiming at addressing the problem of internal covariate shift
- Softmax Layer: The softmax nonlinearity (Bridle, 1989)
  - It is applied to the output layer in the case of multiclass classification
  - It outputs the posterior probabilities over each category

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- **(3) Transfer Learning**
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

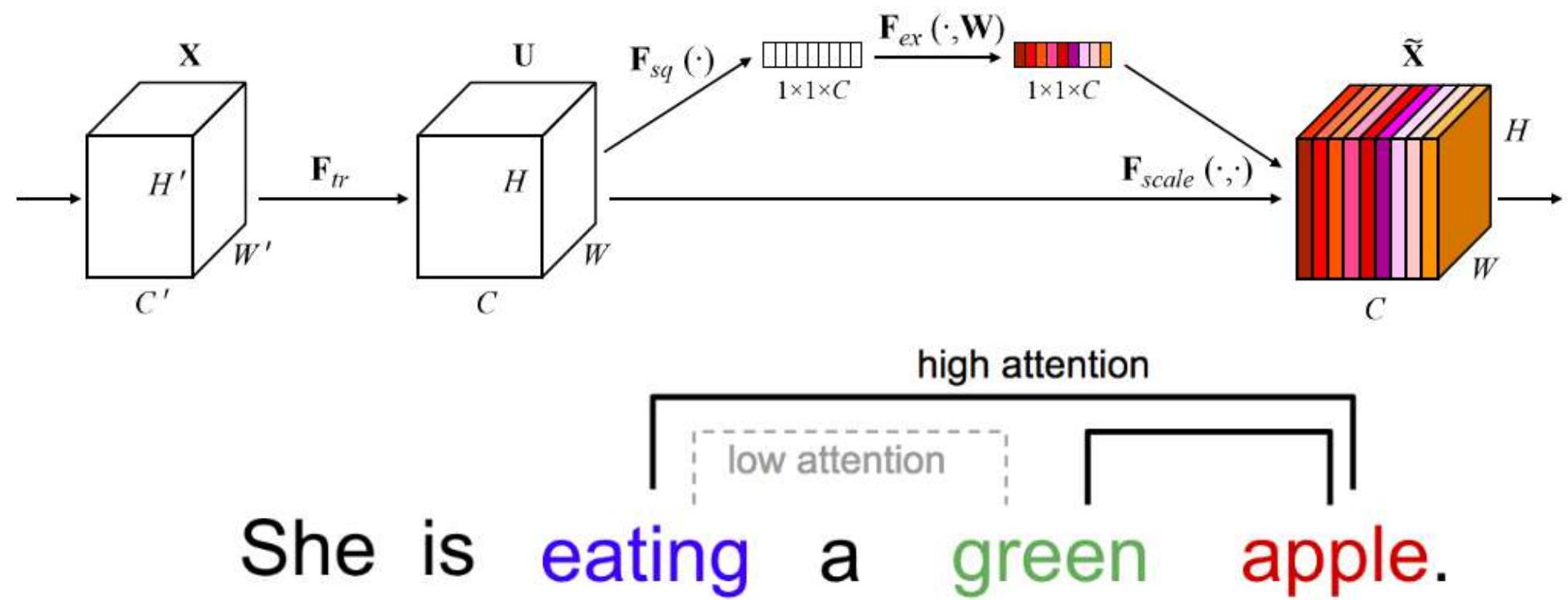
"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned."



# Related Theory

Attention

- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - **(4) Channel Attention**
  - (5) Feature Fusion
  - (6) Depthwise Convolution
  - (7) Design CNNs
- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features



Refers to Squeeze-and-Excitation Networks and BiSeNet

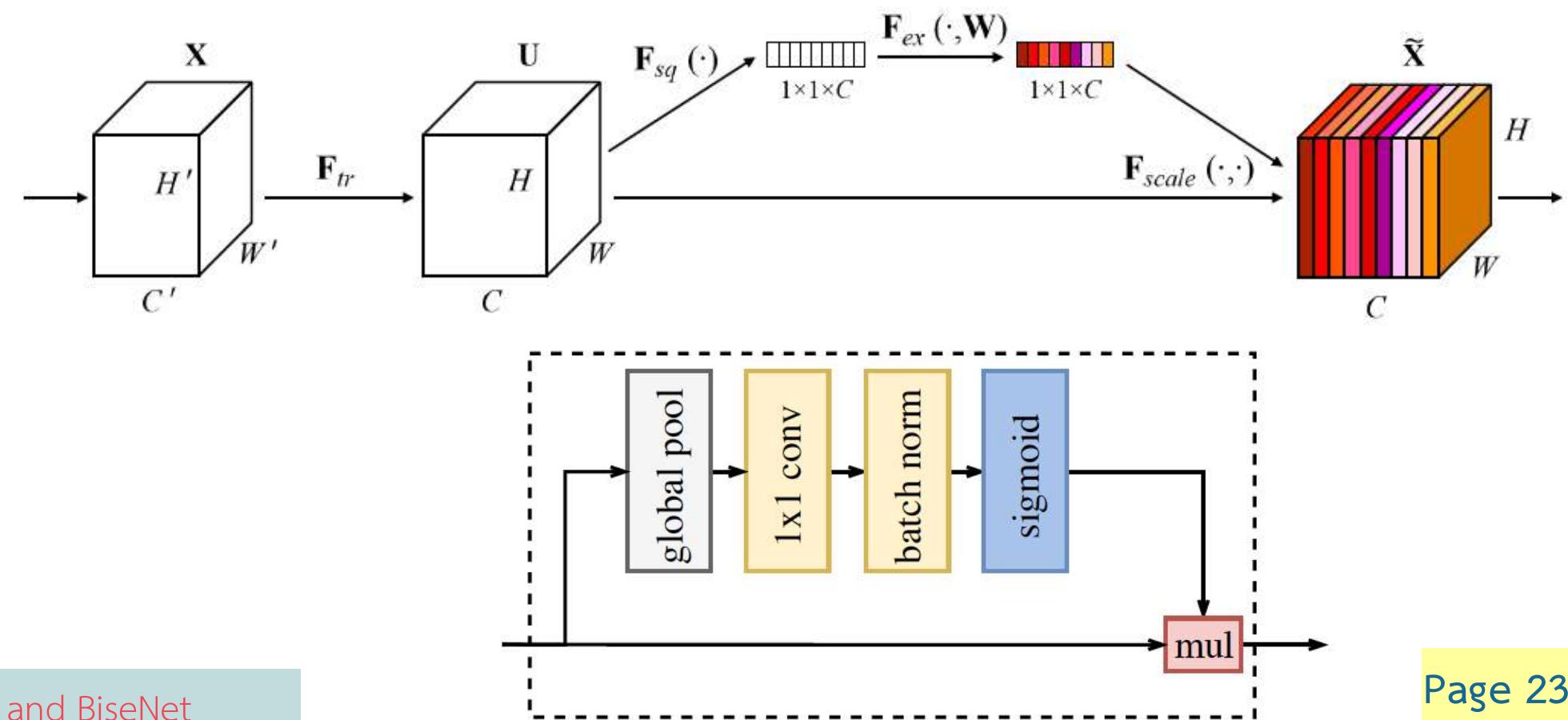
One word “attends” to other words in the same sentence differently.

Page 238

# Related Theory

Attention

- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - **(4) Channel Attention**
  - (5) Feature Fusion
  - (6) Depthwise Convolution
  - (7) Design CNNs
- Attention is helpful to focus on what we want
- We utilize channel attention to select the important features



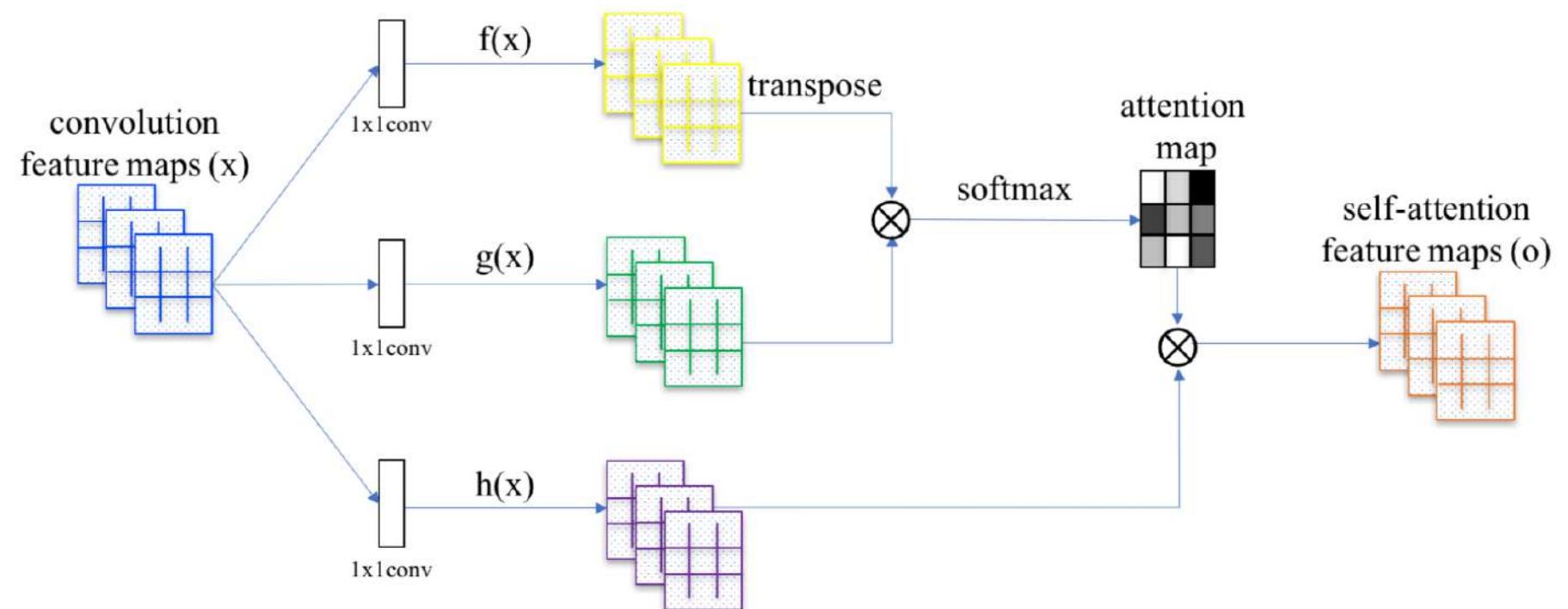
Refers to Squeeze-and-Excitation Networks and BiSeNet

Page 239

# Related Theory

Attention

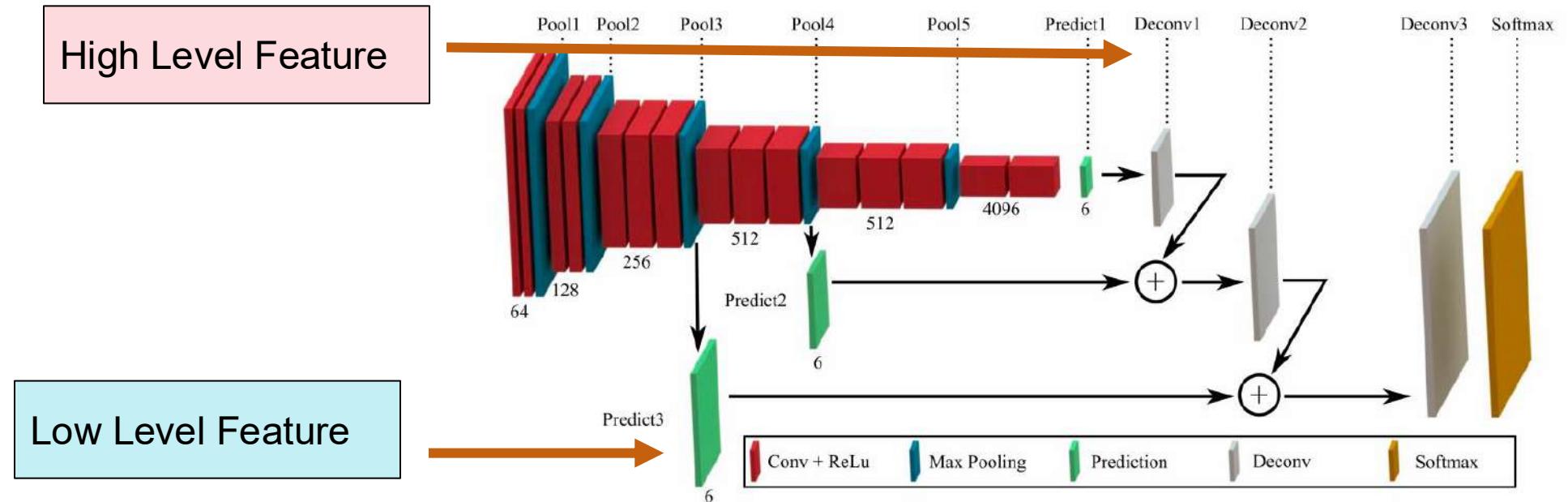
- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - **(4) Channel Attention**
  - (5) Feature Fusion
  - (6) Depthwise Convolution
  - (7) Design CNNs
- Attention is helpful to focus on what we want
  - We utilize channel attention to select the important features



# Related Theory

## Feature Fusion (1)

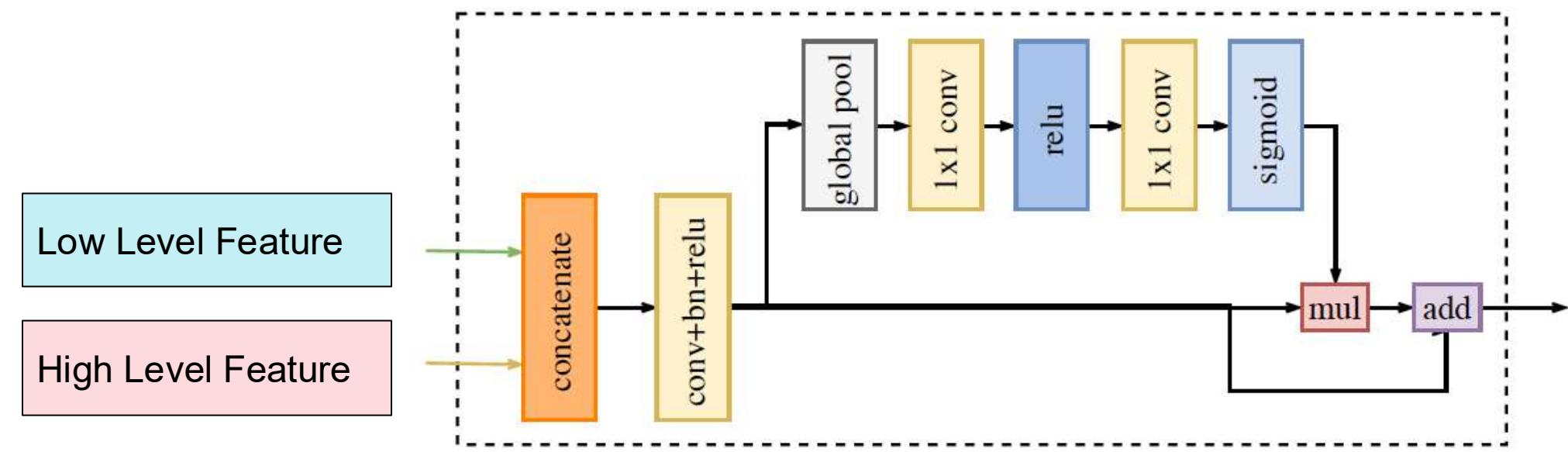
- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - (4) Channel Attention
  - **(5) Feature Fusion**
  - (6) Depthwise Convolution
  - (7) Design CNNs
- The features of the two paths are different in level of feature representation
  - Simply *sum up* low and high features
    - Utilization of low-level features for objects refinement



# Related Theory

## Feature Fusion (2)

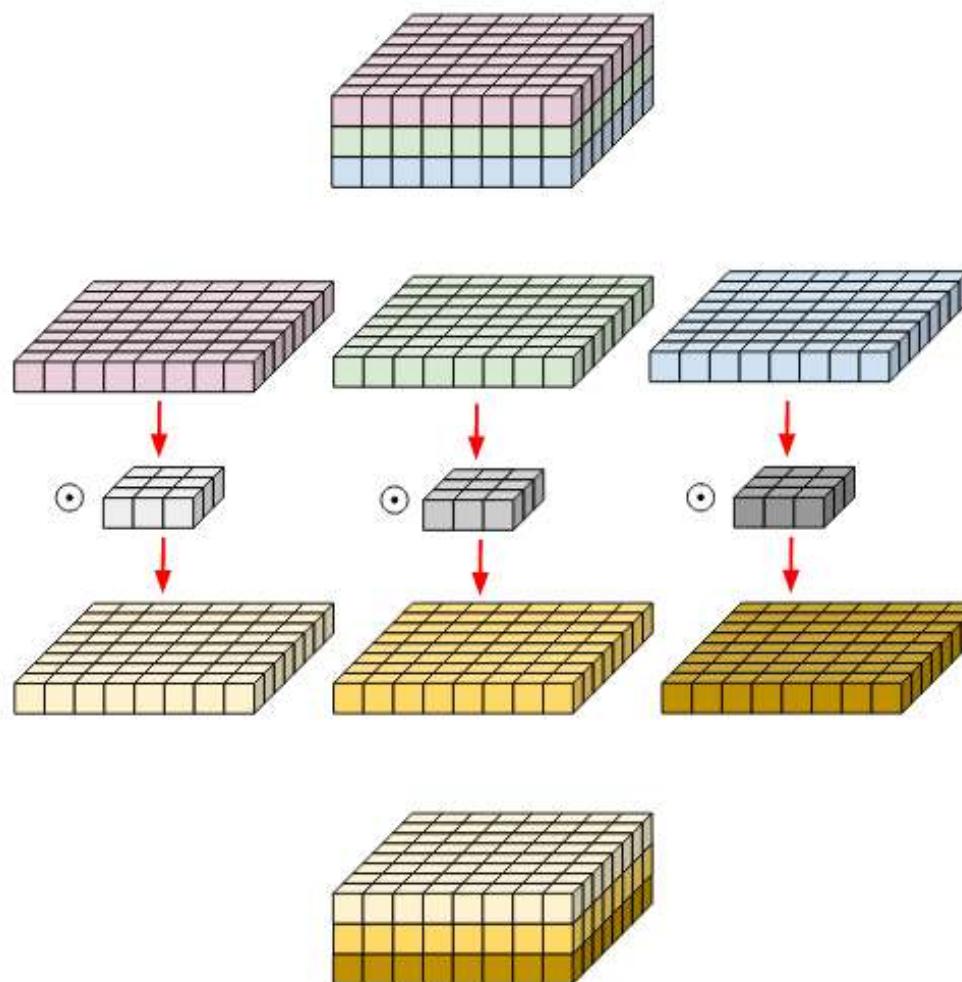
- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - (4) Channel Attention
  - **(5) Feature Fusion**
  - (6) Depthwise Convolution
  - (7) Design CNNs
- The features of the two paths are different in level of feature representation
  - Fuse spatial path (low level features) and context path (high level feature) together



# Related Theory

## Depth-wise Convolution

- (1) Computer Vision Tasks
  - (2) CNNs
    - Traditional CNNs
    - Deep Learning Layers
  - (3) Transfer Learning
  - (4) Channel Attention
  - (5) Feature Fusion
  - (6) Depthwise Convolution
  - (7) Design CNNs
- Filters and image have been broken into three different channels and then convolved separately and stacked thereafter



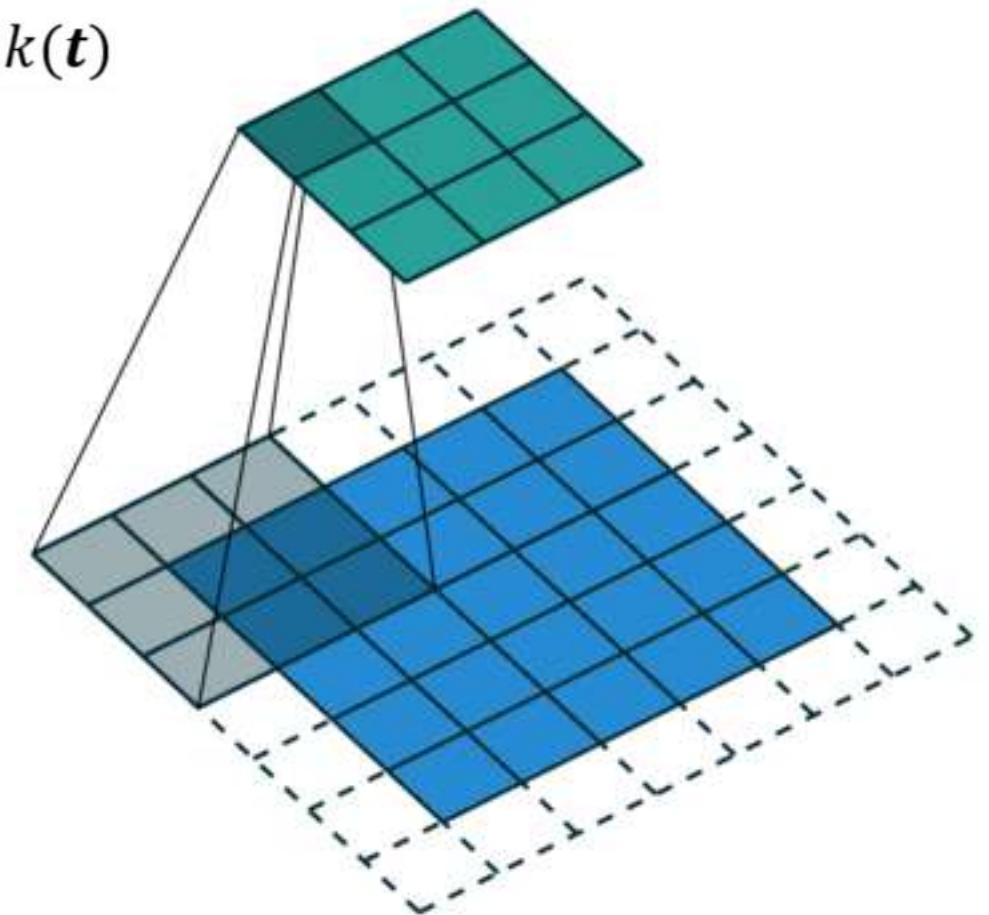
# Related Theory

## Depth-wise Convolution

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs

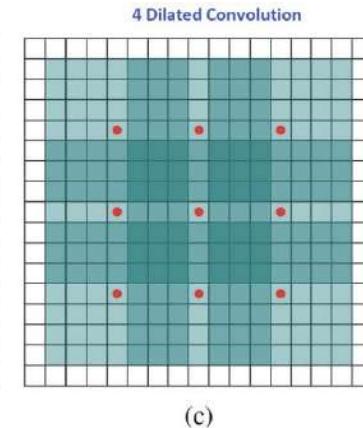
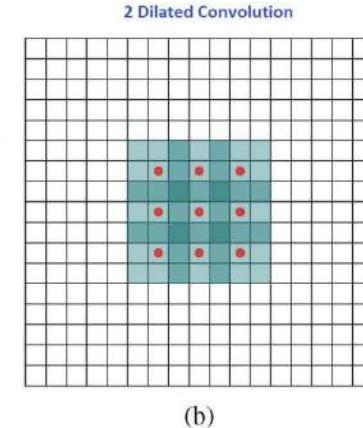
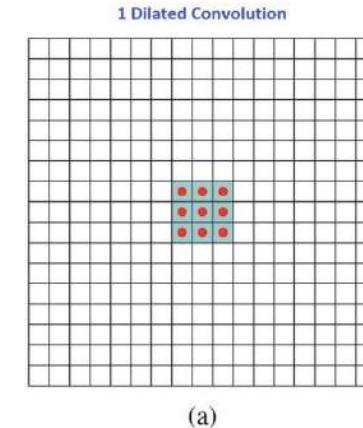
This is the standard discrete convolution:

$$(F * k)(\mathbf{p}) = \sum_{s+t=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$



# Related Theory

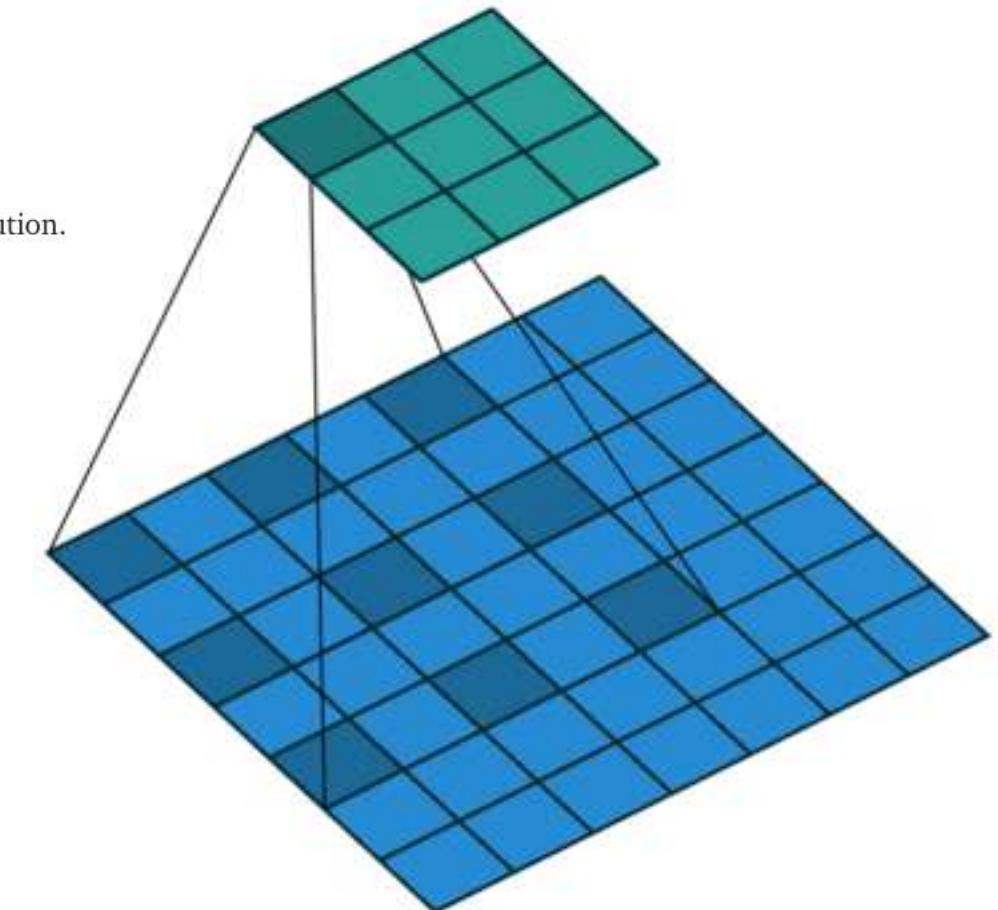
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs



The dilated convolution follows:

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s} + l\mathbf{t} = \mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$

When  $l = 1$ , the dilated convolution becomes as the standard convolution.



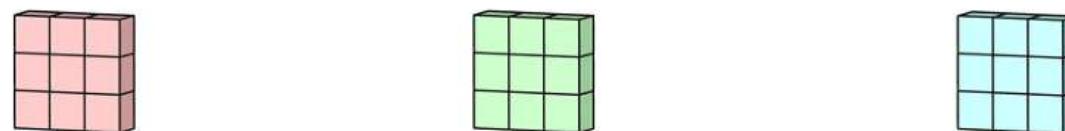
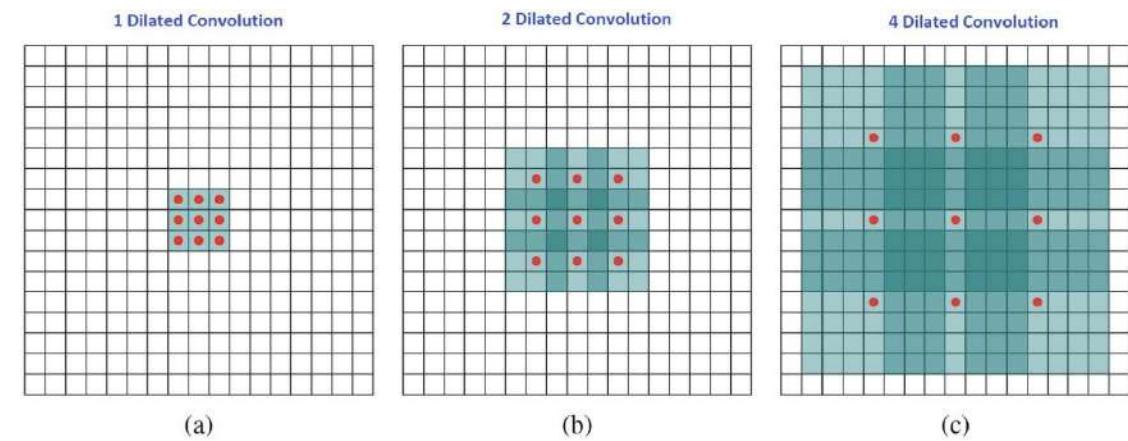
## Dilated Convolution (Atrous Convolution)

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs

## Dilated Convolution (Atrous Convolution)

- **Multi-scale context aggregation** by dilated convolutions

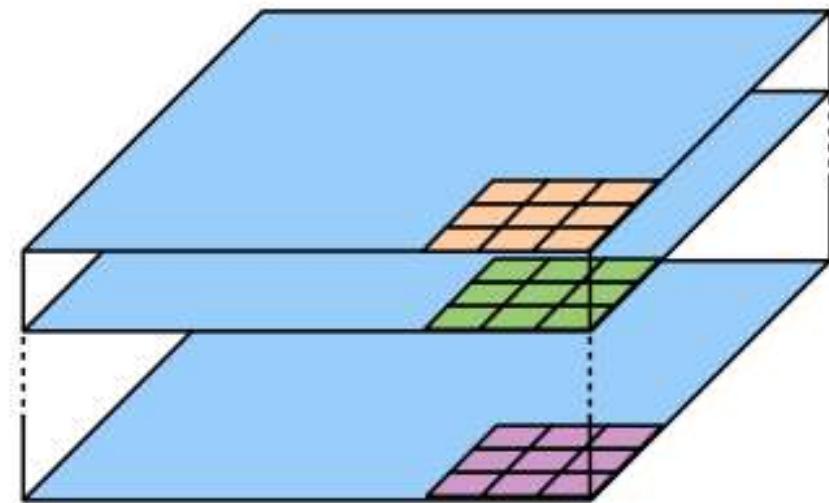


# Related Theory

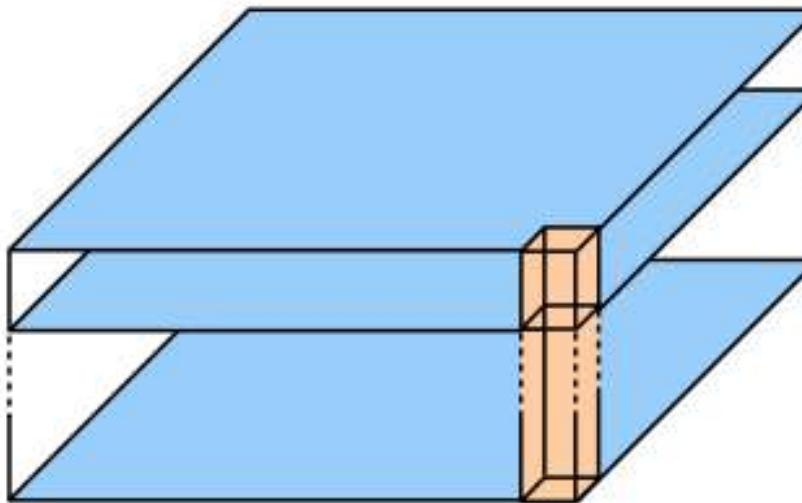
## Dilated Convolution (Atrous Convolution)

- **(6) Depthwise Convolution**

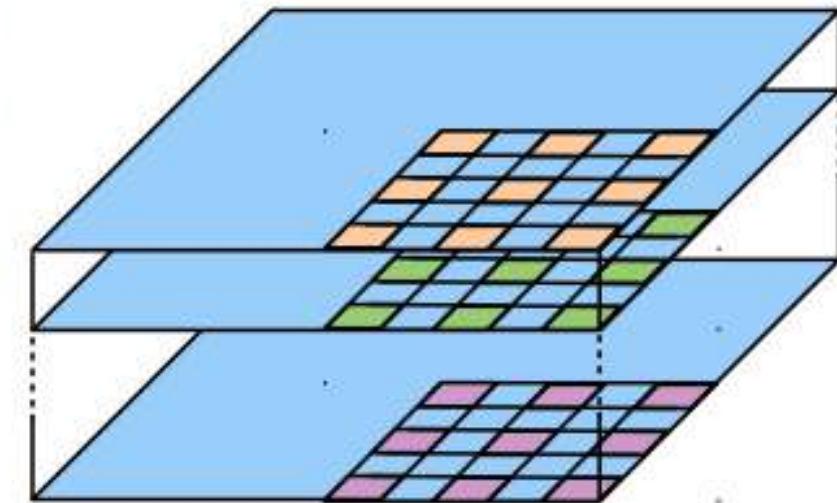
- **Multi-scale context aggregation** by dilated convolutions
- 3x3 Depthwise separable convolution decomposes a standard convolution into (a) a depthwise convolution (applying a single filter for each input channel) and (b) a pointwise convolution (combining the outputs from depthwise convolution across channels). In this work, we explore atrous separable convolution where atrous convolution is adopted in the depthwise convolution, as shown in (c) with rate = 2.



(a) Depthwise conv.



(b) Pointwise conv.

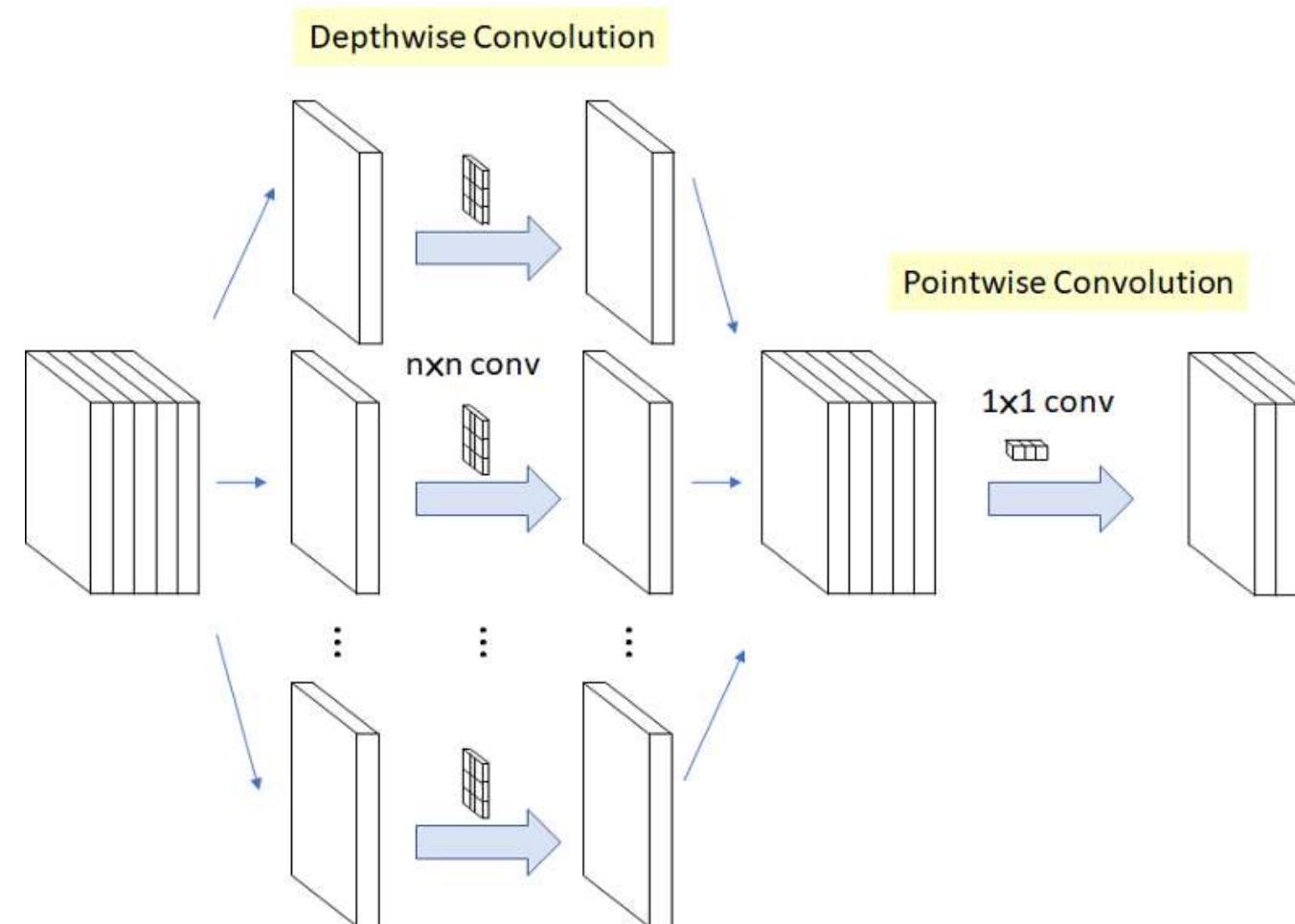


(c) Atrous depthwise conv.

# Related Theory

## Dilated Convolution (Atrous Convolution)

- (6) Depthwise Convolution vs Pointwise Convolution



# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs

## Dilated Convolution (Atrous Convolution)

- Multi-scale context aggregation by dilated convolutions

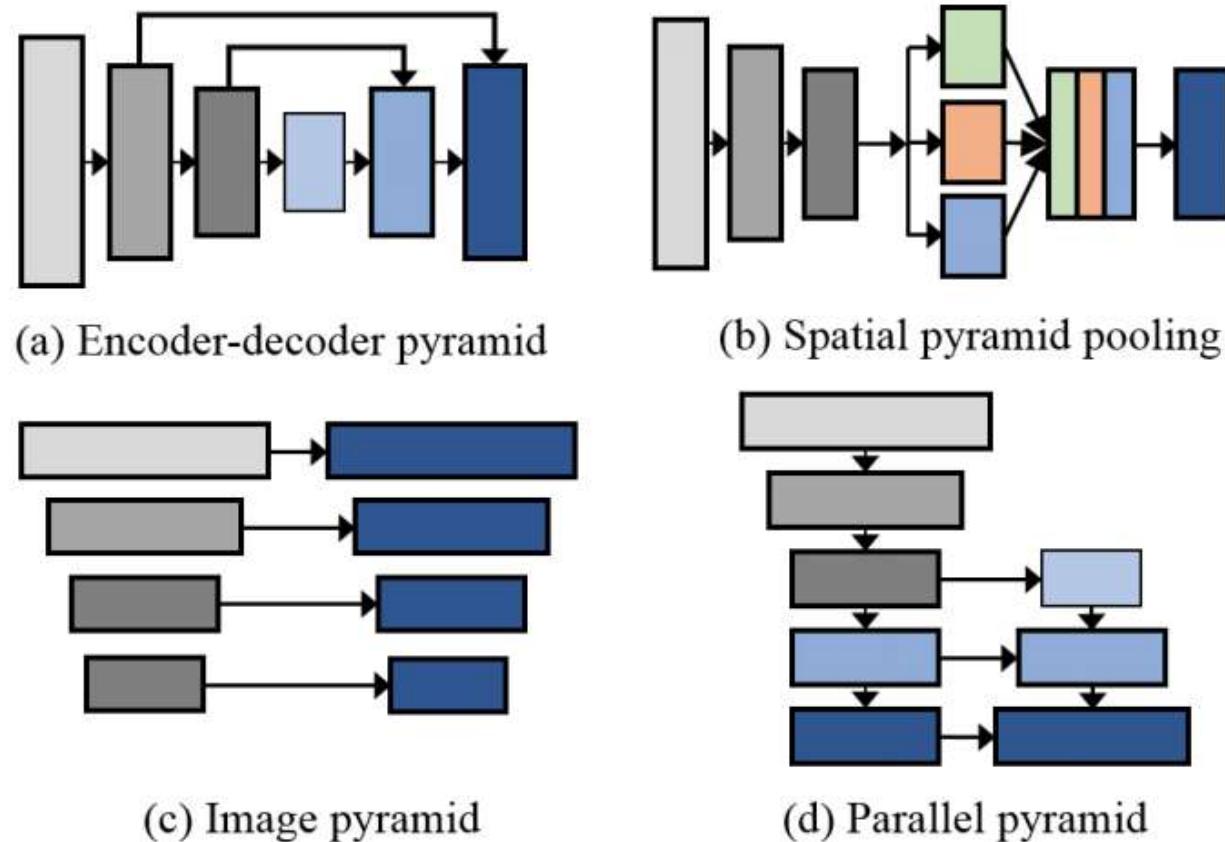


Figure 2. Different pyramids for capturing multi-scale features.

# Related Theory

## Dilated Convolution (Atrous Convolution)

- Multi-scale context aggregation by dilated convolutions

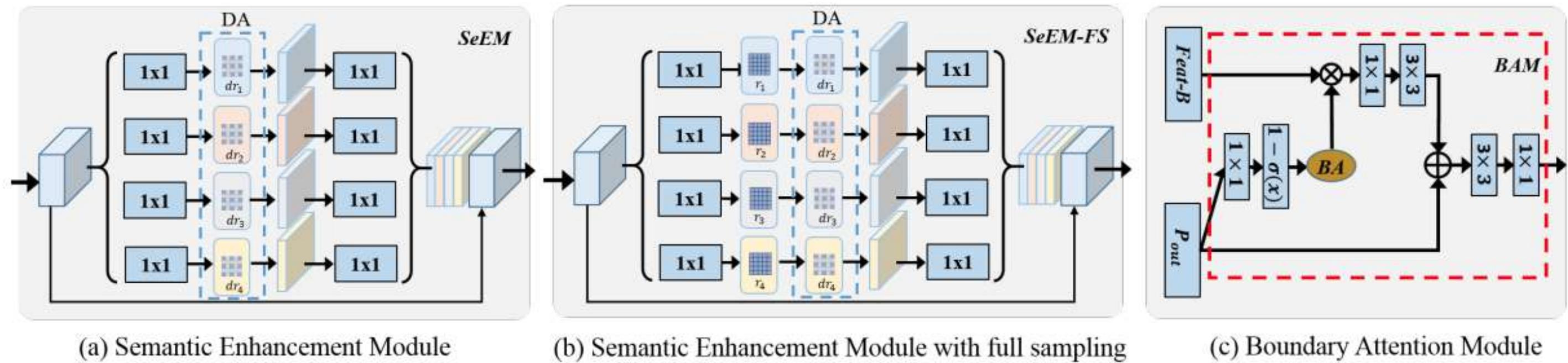


Figure 4. Semantic Modules in the proposed parallel pyramid method for improving feature fusion. We introduce semantic enhancement modules (a) and (b) to enhance the semantics of shallow features, and propose a boundary attention module (c) to extract complementary information from very shallow features and enhance the deep features. ‘DA’ represents depthwise atrous convolution. ‘ $dr_i$ ’ represents the dilation rate. ‘ $r_i$ ’ represents the kernel size of convolutional layer. ‘BA’ represents boundary attention.

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- **(6) Depthwise Convolution**
- (7) Design CNNs

## Dilated Convolution (Atrous Convolution)

- **Multi-scale context aggregation** by dilated convolutions

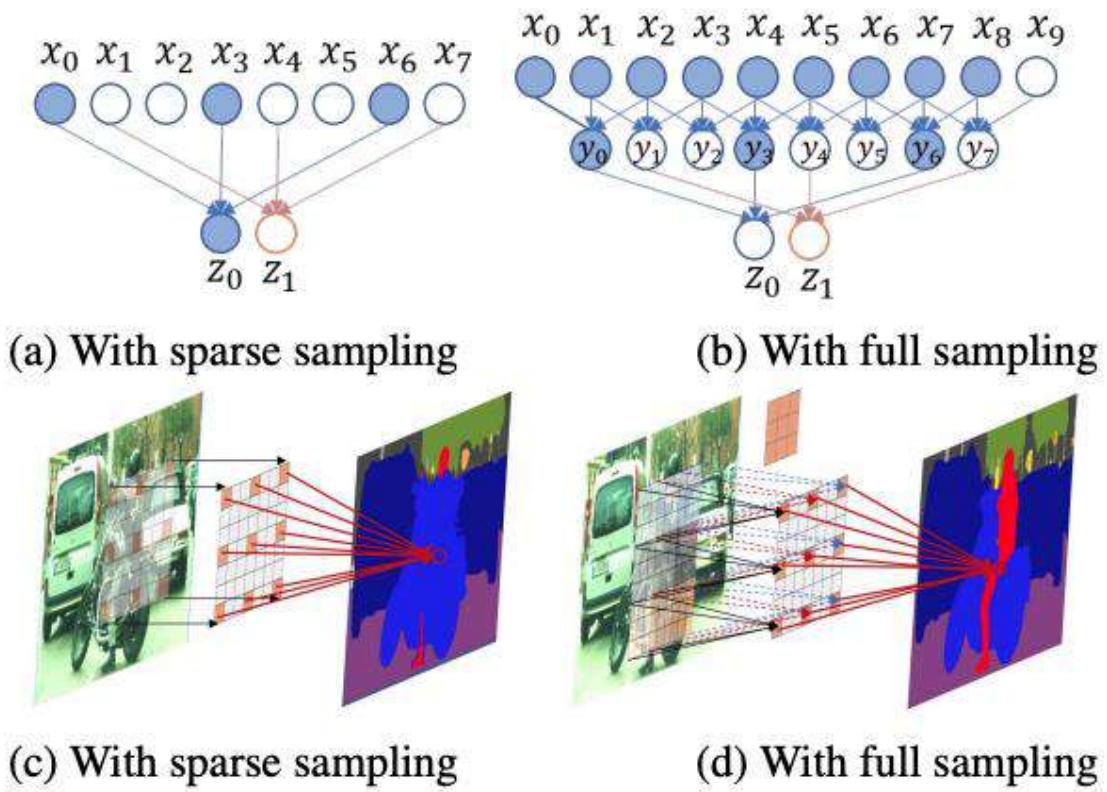
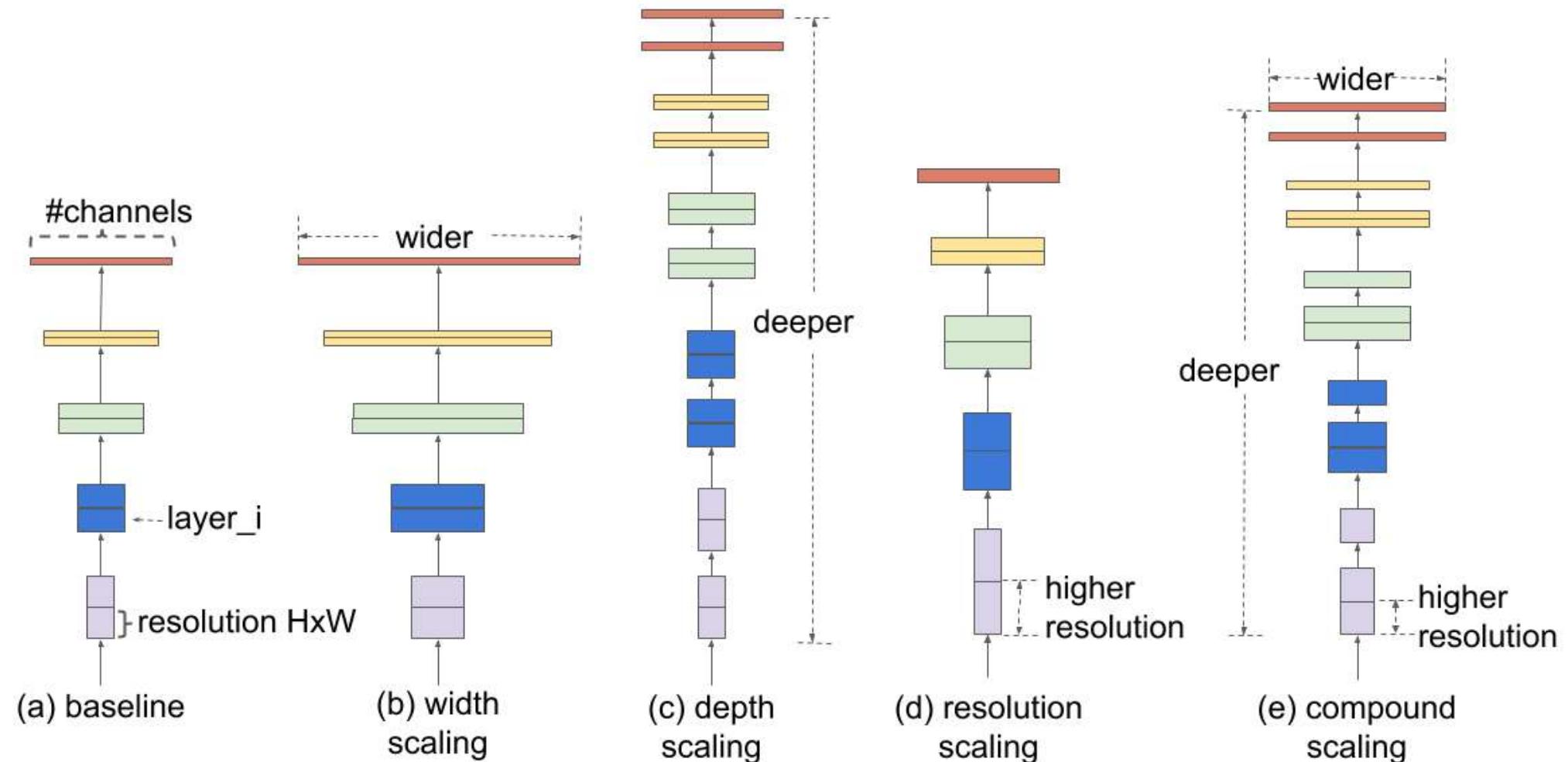


Figure 5. Atrous convolution with sparse sampling in SeEM and full sampling in SeEM-FS.

# Related Theory

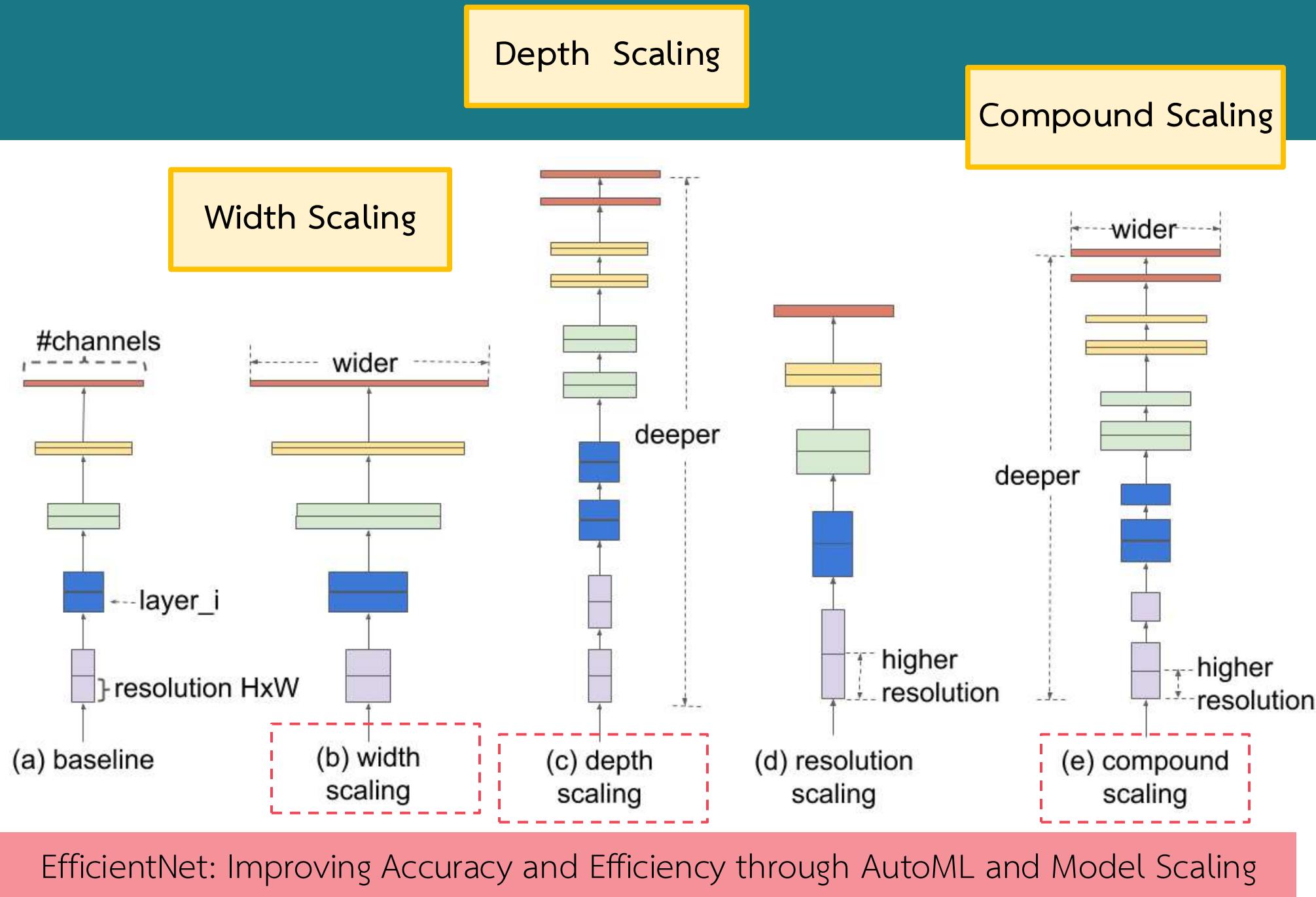
- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs



EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling

# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

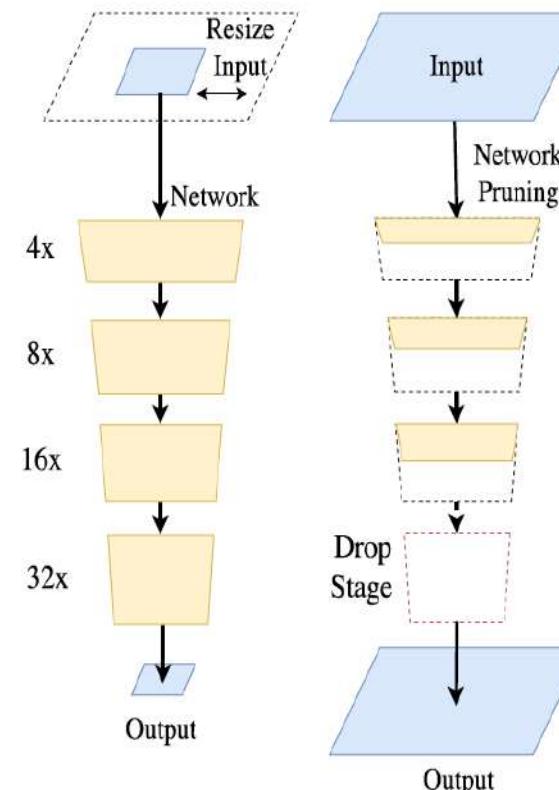


# Related Theory

- (1) Computer Vision Tasks
- (2) CNNs
  - Traditional CNNs
  - Deep Learning Layers
- (3) Transfer Learning
- (4) Channel Attention
- (5) Feature Fusion
- (6) Depthwise Convolution
- (7) Design CNNs

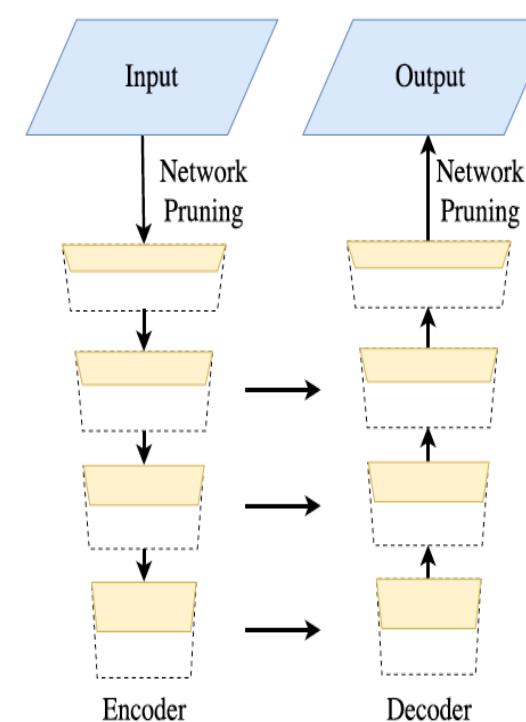
Dept Scaling

VGG Style  
(Depth Scaling)



Width Scaling

U-Shape Style  
(Width Scaling)



Compound Scaling

Context Path Style  
(Compound Scaling)

