**Author:** Teerapong **Panboonyuen**

**Reference:**


[1] Google Scholar: https://scholar.google.co.th/citations?user=myy0qDgAAAAJ&hl=en

[2] Research Gate (RG): https://www.researchgate.net/profile/Teerapong-Panboonyuen

[3] Web of Science ResearcherID: AAO-4985-2020: https://publons.com/researcher/1730918/teerapong-panboonyuen/


Panboonyuen's **Selected Articles:**


[1] **Panboonyuen, T.**; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Transformer-Based Decoder Designs for Semantic Segmentation on Remotely Sensed Images. Remote Sens. 2021, 13, 5100. https://doi.org/10.3390/rs13245100

[2] **Panboonyuen, T.**; Thongbai, S.; Wongweeranimit, W.; Santitamnont, P.; Suphan, K.; Charoenphon, C. Object Detection of Road Assets Using Transformer-Based YOLOX with Feature Pyramid Decoder on Thai Highway Panorama. Information 2022, 13, 5. https://doi.org/10.3390/info13010005

[3] **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. https://doi.org/10.3390/rs12081233

[4] **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. https://doi.org/10.3390/rs11010083

[5] **Panboonyuen, T.**; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Road Segmentation of Remotely-Sensed Images Using Deep Convolutional Neural Networks with Landscape Metrics and Conditional Random Fields. Remote Sens. 2017, 9, 680. https://doi.org/10.3390/rs9070680

*remote sensing*

MDPI

*Article*

# Transformer-Based Decoder Designs for Semantic Segmentation on Remotely Sensed Images

**Teerapong Panboonyuen [1], Kulsawasd Jitkajornwanich [2], Siam Lawawirojwong [3], Panu Srestasathiern [3] and Peerapon Vateekul [1,*]**

[1] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand; teerapong.panboonyuen@gmail.com
[2] Data Science and Computational Intelligence (DSCI) Laboratory, Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd, Ladkrabang, Bangkok 10520, Thailand; kulsawasd.ji@kmitl.ac.th
[3] Geo-Informatics and Space Technology Development Agency (Public Organization), 120, The Government Complex, Chaeng Wattana Rd, Lak Si, Bangkok 10210, Thailand; siam@gistda.or.th (S.L.); panu@gistda.or.th (P.S.)
* Correspondence: peerapon.v@chula.ac.th



**Citation:** Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Transformer-Based Decoder Designs for Semantic Segmentation on Remotely Sensed Images. *Remote Sens.* **2021**, *13*, 5100. https://doi.org/10.3390/rs13245100

Academic Editors: Alireza Taravat, Naoto Yokoya, Jon Atli Benediktsson, Hongjun Su, Cristina Rubio-Escudero, Antonio Morales Esteban, José L. Amaro-Mellado, Francisco Martínez-Álvarez, Ata Jahangir Moshayedi, Biplab Banerjee and Mercedes E. Paoletti

Received: 31 October 2021
Accepted: 10 December 2021
Published: 15 December 2021



**Abstract:** Transformers have demonstrated remarkable accomplishments in several natural language processing (NLP) tasks as well as image processing tasks. Herein, we present a deep-learning (DL) model that is capable of improving the semantic segmentation network in two ways. First, utilizing the pre-training Swin Transformer (SwinTF) under Vision Transformer (ViT) as a backbone, the model weights downstream tasks by joining task layers upon the pretrained encoder. Secondly, decoder designs are applied to our DL network with three decoder designs, U-Net, pyramid scene parsing (PSP) network, and feature pyramid network (FPN), to perform pixel-level segmentation. The results are compared with other image labeling state of the art (SOTA) methods, such as global convolutional network (GCN) and ViT. Extensive experiments show that our Swin Transformer (SwinTF) with decoder designs reached a new state of the art on the Thailand Isan Landsat-8 corpus (89.8% *F*1 score), Thailand North Landsat-8 corpus (63.12% *F*1 score), and competitive results on ISPRS Vaihingen. Moreover, both our best-proposed methods (SwinTF-PSP and SwinTF-FPN) even outperformed SwinTF with supervised pre-training ViT on the ImageNet-1K in the Thailand, Landsat-8, and ISPRS Vaihingen corpora.

**Keywords:** vision transformer; fully transformer networks; convolutional neural network; feature pyramid network; high-resolution representations; ISPRS Vaihingen; Landsat-8

## 1. Introduction

In general, automated semantic segmentation is studied to analyze remote sensing [1–3]. Research into semantic segmentation of aerial or satellite data has grown in importance. Over the years, due to its full range of autonomous driving, automatic mapping, and navigation application, significant progress has been made in this field. In the last decade, DL has been revolutionized by computer science. Among modern convolutional neural networks (ConvNet/CNNs), there are many techniques, e.g., dual attention [4] and self-attention [5], that have gained increasing attention due to their capability. Such techniques generate highly precise semantic segmentation from remote sensing data. Still, all suffer from issues regarding the accuracy of performance.

Currently, many deep learning architectures [2,6] have been applied in urban or agriculture segmentations, such as global convolutional networks [7], DeepLab [8], mask R-CNN [9], BiseNet [10], and CCNet [11]. These networks have been created for semantic recognition and consist of stacked convolution blocks. Due to reduced costs of computation, the use of kernel maps has decreased gradually.

Thus, the encoder network can learn more semantic visual theories with a steadily increased receptive field. Consequently, this also inflates a primary restriction of studying long-range dependency knowledge, which is significant for computer vision tasks. However, the situation is still challenging due to the limited size of the region in the input that produces the feature. These receptive fields require dense high-resolution predictions; transformers conduct self-attention on that receptive field. Previously, architecture has not fully leveraged various feature maps from convolution or attention blocks conducive to image segmentation, and this was a motivation for this work.

To overcome this weakness, completely new networks viz. Swin Transformer (SwinTF) [12] with Vision Transformer (ViT) [13] as the major backbone, have a tremendous capacity in long-range dependency acquisition and sequence-based picture modeling. Transformers are the first transduction models that rely entirely on self-attention to compute their input and output representations without using sequence-aligned RNNs or convolution. No recurrent units are used to obtain these features; they are simply weighted sums and activations, which prove to be very efficient and parallelizable [14].

ViT is one of the most well-known Transformers used in several computer vision tasks, such as hyperspectral image classification [15,16], bounding-box detection [17,18], and semantic segmentation [19,20]. ViT moves the window divider between successive levels of self-attention. The shifted windows provide links between the windows of the last layer, considerably increasing modeling capability.

Most relevant to our proposed method is the Vision Transformer (ViT) [13] and their follow-ups [21–25]. ViT is a deep learning architecture that utilizes the mechanism of attention, focusing on image recognition and is greatly valued in their works [21–25]. Several works of ViT directly employ a transformer model on non-overlapping medium-sized image patches for image classification. ViT reaches an exciting speed-performance trade-off on almost all computer vision tasks compared to previous DL networks. DeiT [26] introduces several training policies that also allow it to be efficient using the extra modest ImageNet-1K corpus.

The effects of ViT on computer vision tasks are encouraging. The ViT model is inappropriate for low-resolution kernel filters and the image size's quadratic improvement in complexity. Some works utilize ViT models for the dense image tasks of semantic segmentation and detection. Notably, ViT [12,27] models are seen to have the best performance-accuracy trade-offs among these methods on computer vision tasks, even though this work concentrates mostly on general-purpose performance rather than focusing on semantic segmentation.

Moreover, it usually takes high computational costs for the previous transformer network, e.g., Pyramid ViT [28], which is quadratic to the size of an image. In contrast, SwinTF has solved the computational issue and costs linear to the image size. SwinTF has improved the accuracy by operating the model regionally, enhancing receptive fields that highly correlate to visual signals. Furthermore, it is efficient and effective, achieving SOTA performance, e.g., *MeanIoU*, *AveragePrecision* on COCO object detection, and ADE20K semantic segmentation.

In this paper, transformer-based decoder designs for multi-object segmentation from medium-resolution (Landsat-8) and very high-resolution (aerial) images are introduced, as demonstrated in Figures 1 and 2. This work helps to further improve SOTA on semantic segmentation in Landsat-8 and aerial images. For better performance, three styles of decoder designs into transformer-based reasoning are implemented. Our goals are two-fold:

- Utilizing a pre-training ViT to retrieve the virtual visual tokens based on the vision patches from aerial and satellite images: we immediately fine-tune the model weights on downstream responsibilities by appropriating pre-training SwinTF under ViT, as a backbone, by appending responsibility layers and superimposing the pretrained encoder.

- Proposing the decoder designs to our DL network with three decoder designs including (i) U-Net [29], (ii) pyramid scene parsing (PSP) network [30], and (iii) feature pyramid network (FPN) [31] to perform pixel-level segmentation.

The experimental results on three remotely sensed semantic segmentation corpora, including two Thailand Landsat-8 data sets and one ISPRS Vaihingen [32] corpora, demonstrate the effectiveness of the proposed scheme. The results prove that our SwinTF with decoder designs can overcome the previous encoder–decoder network [33–36] on aerial and satellite images and Swin Transformer models [12] in terms of the *Precision*, *Recall*, and $F1$ score sequentially.

The remainder of this article is structured as follows. Section 2 discusses the materials and methods. The results are detailed in Section 3, and Section 4 presents our discussion, including our limitations and outlook. Finally, our conclusions are drawn in Section 5.

## 2. Material and Methods

### *2.1. Transformer Model*

#### 2.1.1. Transformer Based Semantic Segmentation

SwinTF follows a sequence-to-sequence vector with transformers [37] as well as a corresponding output vector with input vector fabrication, such as NLP. NLP concerns the interaction between computers and human language in order to process and analyze a large amount of matured language. Accordingly, the SwinTF, as described in Figure 1 allows a $1D$ sequence of vector embeddings $z \in R^{L \times C}$ as input, $L$ is the length of the vector, and $C$ is the hidden kernel size. The image sequence is consequently obliged to modify an input layer of image $x \in R^{H \times W \times 3}$ into $Z$.

The traditional SwinTF model [12] focuses on the relationship between a token (image patches); the other tokens are calculated. ViT focuses on the quadratic complexity concerning the number of image patches; finding it unsuitable for many image problems requiring an immense set of tokens for the softmax layer.

A traditional transformer-based encoder learns vector representations as to the $1D$ vector of embedding sequence $E$ input. This means that each ViT layer has a global receptive field, which answers the insufficient receptive field problems of the existing encoder–decoder deep neural network. The ViT encoder consists of $L_e$ layers of multilayer perceptron (MLP) and multi-head self-attention (MSA) modules.

A method for the sequence of image vectors is to flatten the pixel of values of images within a $1D$ vector with a size of $3 \times H \times W$. For a representative image, i.e., 512($H$) × 512($W$) × 3, the resulting vector will have a length of 786,432. It is not conceivable that such high-dimensional vectors can be handled in both time and vector space. Accordingly, tokenizing every pixel of the image as input to our SwinTF is subject to a linear embedding layer.

In the case whereby a conventional encoder designed for semantic segmentation would downsample a $2D$ image $x \in R^{HW3}$ into a grid via a featuremap $x_f \in R^{\frac{H}{16} \times \frac{W}{16} \times C}$, we decided to set the transformer input sequence length $L$ as $\frac{H}{16} \times \frac{W}{16} = \frac{W}{256}$. This means that the output of the vector sequence of ViT can be clearly reshaped to the point kernel map $x_f$.

To recover the $\frac{HW}{256}$-long vector sequence of our input, we divide the image $x \in R^{H \times W \times 3}$ into a grid of as $\frac{H}{16} \times \frac{W}{16}$ patches. Thus, several ViT modules with modified self-attention calculation (SwinTF modules) are adapted on these image patch tokens. The ViT module maintains the number of patches $\frac{H}{4} \times \frac{W}{4}$ and then makes a series out of this grid. Each vectorized patch p is mapped into a latent C-dimensional embedding space using a linear projection function. $f : p \to e \in R^C$, for a patch $x$; we obtain a 1D series of vector embeddings. Therefore, we obtain a unique embedding $p_i$ for each position $i$ to encode the patch spatial information, which is then added to $e_i$ to generate the final sequence input $E = \{e_1 + p_1, e_2 + p_2, ..., e_L + p_L\}$. In this process, spatial data is kept, notwithstanding the order-less attention type of transformers.

A classical transformer-based encoder accepts feature representations when given the 1D embedding sequence $E$ as input. This encoder means that each ViT layer has a global

receptive field, resolving the problem of the standard deep learning encoder's restricted sensory area once and for all. The encoder of SwinTF consists of $L_e$ vector of MLP and MSA modules (Figure 1). At each layer $l$, the input to self-attention is depicted as a triplet of $(query, key, value)$, and calculated from the input $Z^{l-1} \in R^{L \times C}$ as:

$$query = Z^{l-1}W_Q, key = Z^{l-1}W_K, value = Z^{l-1}W_V \quad (1)$$

where $W_Q/W_K/W_V \in R^{C \times d}$ are the learnable weights of three linear projection vectors and $d$ is the dimension of $(query, key, value)$. Self-attention (SA) is then expressed as:

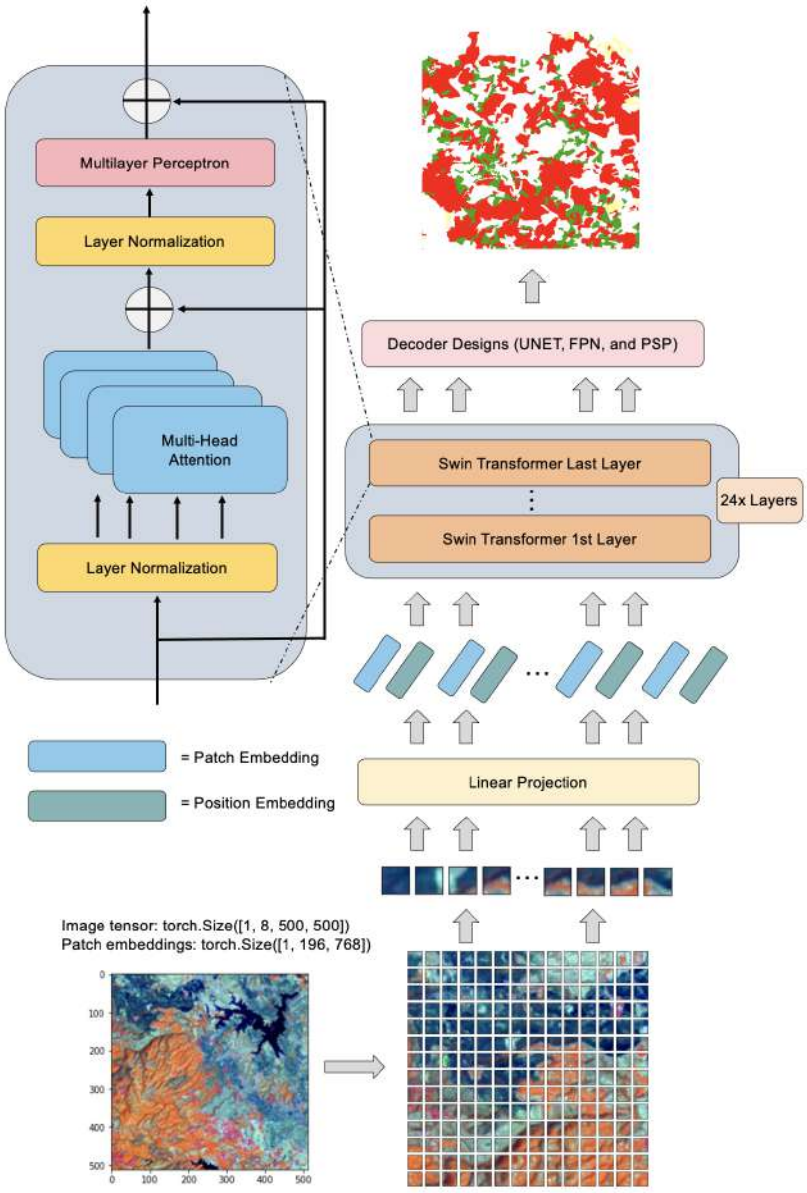



**Figure 1.** The overall architecture of our SwinTF.

$$SA(Z^{l-1}) = Z^{l-1} + softmax(\frac{Z^{l-1}W_Q(ZW_K)^T}{\sqrt{d}})(Z^{l-1}W_V) \quad (2)$$

MSA clearly calculated a reckoning with m self-supporting SA actions and projects their concatenated outputs: $MSA(Z^{l-1}) = [SA_1(Z_l - 1); SA_2(Z_l - 1); \ldots; SA_m(Z_l - 1)]W_O$. Where $W_O \in R^{md} \times C$. $d$ is typically set to $C/m$. The output of MSA is then transformed by an MLP module with a residual skip as the output layer as:

$$Z^l = MSA(Z_{l-1}) + MLP(MSA(Z^{l-1})) \in R^{L \times C}. \quad (3)$$

Lastly, a normalized layer is employed before MLP and MSA modules, which are omitted for clearness. We express $Z^1, Z^2, Z^3, \ldots, Z^{Le}$ as the weights of transformer vectors.

#### 2.1.2. Decoder Designs

To assess the effectiveness of SwinTF's encoder vector, as represented by Z, three various decoder designs as portrayed in Figure 2 are set up to achieve pixel-level labeling. Next, the three decoders can be expressed as:

(1) U-Net [29]: The expansion route (decoder) on the right-hand side applies transposed convolutions with ordinary convolutions. The image size gradually increases in the decoder, whereas the depth gradually decreases. To improve precision, we employ the skip connections at every stage of the decoder by concatenating the output of the transposed convolution layers with the feature maps from the encoder at the same level. The encoder path's high-resolution (but semantically infirm) characteristics are mixed and reused with the upsampled output in this way.

As seen in the diagram below, U-Net has an asymmetrical design. Every step in the expanding direction, consisting of an upsampling of the feature map followed by a $2 \times 2$ transpose convolution that halves the number of feature channels, is used in the Decoder route. Accordingly, we have a concatenation with the contracting path's appropriate feature map, as well as a $3 \times 3$ convolutional neural network (each followed by a Rectified Linear Unit (ReLU)). A $1 \times 1$ convolution transfers the channels to the required number of classes in the final layer. Such a purpose is to bridge the feature gap between the decoder and encoder feature maps before concatenation.

(2) For pixel-level scene parsing, the PSP network is used and provides excellent global contextual prior [30]. The pyramid pooling module can capture more representative levels of data than global average pooling (GAP). The concept of sub-region average pooling is comparable to SPPNet's Spatial Pyramid Pooling [38]. Bilinear interpolation is employed to make all the feature maps' sizes equal; the 11 convolution then concatenation is akin to the depthwise convolution in Depthwise Separable Convolution utilized by Xception [39] or MobileNet [40]. To minimize the detrimental effect as much as possible, upsampling to $2\times$ is limited.

As a result, full-resolution from $BZ^{Le}$ with size $A\frac{H}{16} \times \frac{W}{16}$ requires a total of four processes. The green layer, as seen in Figure 2, is the coarsest level, performing GAP over each feature map to provide a single bin output. The yellow layer is the second level, which divides the feature map into $2 \times 2$ sub-regions and performs average pooling for each of them. The third level, the light blue layer, separates the feature map into 33 sub-regions before serving average pooling for each sub-region. Finally, each low-dimension feature map is up-sampled to the same size as the original feature map (last blue layer), followed by a convolution layer to produce the final prediction map.

(3) FPN [31] is a characteristic extractor created with accuracy and speed in mind for such a pyramid idea. FPN takes the place of detectors, like Faster R-feature CNN's extractor [41]. Image recognition generates many feature map layers (multi-scale feature maps) and has superior quality to the traditional feature pyramid. FPN also utilizes specifically constructed transformers in a self-level, top-down, and bottom-up interactive pattern to change any feature pyramid into another feature pyramid of the same size but with richer contexts. The simple query, key, and value operation (Equation (1)) demonstrates its importance in choosing informative long-range interaction, which fits our objective of non-local interaction at appropriate sizes.

The higher-level feature using the visual qualities of the lower-level "pixels" is depicted. Each level's feature maps (red, yellow, and blue) are resized to their matching map size and concatenated with the original map before being sent to the convolution layer, which resizes them to the accurate "thickness". Higher-resolution features are upsampled from higher-pyramid-level feature maps, which are spatially coarser but semantically more robust. Spatial resolution is upsampled by a factor of two, with the nearest neighbor being used for simplicity. Each lateral link combines feature maps from the bottom-up and top-down paths of the same spatial size. To minimize the channel dimensions, the feature maps from the bottom-up course are convolutional (11 times).

In addition, element-wise addition is used to combine the feature maps from the bottom-up and top-down pathways. Finally, a 33 convolution is applied to each merged map to form the final feature map to reduce the aliasing impact of upsampling. This last collection of feature maps corresponds to the precise spatial dimensions. As all layers of the pyramid, as in a standard featured picture pyramid, employ joint classifiers/regressors, the feature dimension at output d is fixed at $d = 256$. As a result, the outputs of all further convolutional layers are 256-channel.

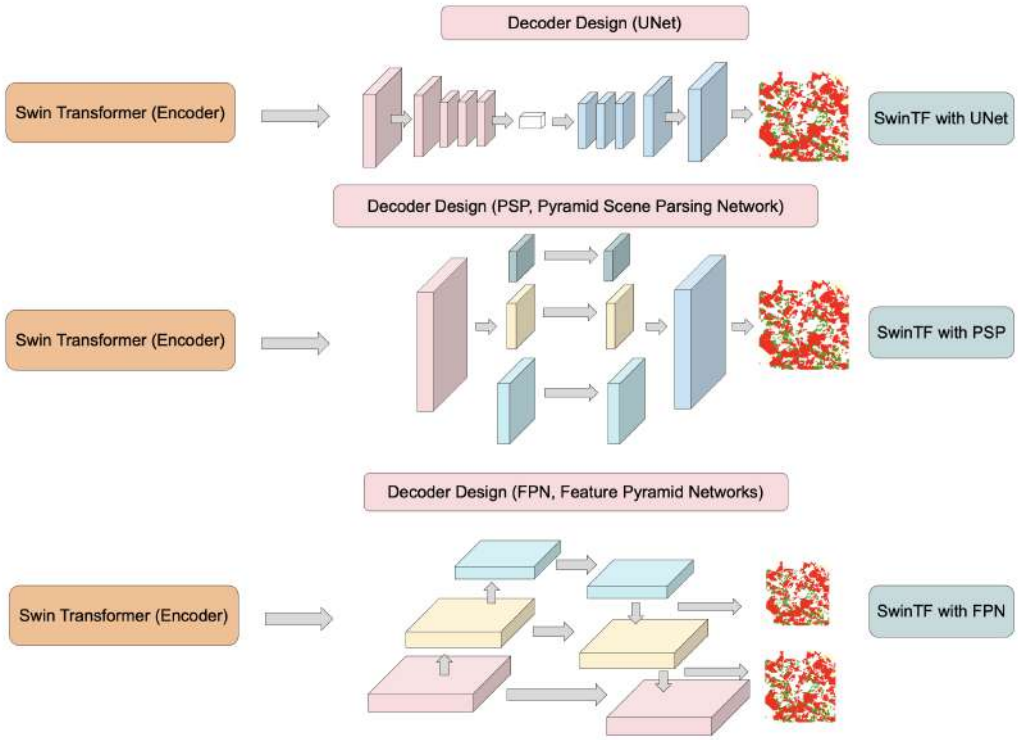



**Figure 2.** SwinTF with three variations of our decoder designs: SwinTF-UNet, SwinTF-PSP, and SwinTF-FPN.

#### 2.1.3. Environment and Deep Learning Configurations

Herein, a stochastic depth dropout of 0.25 for the first 70% of training iterations is employed, and the dropout ratio to 0.6 is increased for the last 20%. As for the multi-scale flipping testing, testing scales of 0.5, 0.75, 1.0, 1.25, 1.5, and 1.75 are presented along with random horizontal flips by following standard practices, as in the literature (e.g., [12,13,31,37]) throughout training for all the experiments.

As the optimizer, a learning rate (LR) schedule is used with Stochastic gradient descent (often abbreviated SGD) for optimizing an the loss function with suitable smoothness properties. Weight decay and momentum are locked to 0.25 and 0.75, sequentially, for all the experiments on the three datasets. The initial LR of 0.0001 is set up on the Thailand Landsat-8 corpora and 0.001 on the ISPRS Vaihingen data set. Finally, batch normalization

in the fusion layers is employed and carried out using batch size 48. Images are resized to 512 pixels side length.

### *2.2. Aerial and Satellite Imagery*

There are three primary sources of data in our experiments: one public and two private data sets. The private data sets are medium resolution imagery gathered from the satellite "Landsat-8" owned by the Thai government's Geo-Informatics and Space Technology Development Agency (GISTDA). As there are two different annotations, the Landsat-8 data is divided into two categories (Isan and North corpora), as illustrated in Table 1. The public data collection consists of high-resolution imagery from the "ISPRS Vaihingen (Stuttgart)" standard benchmark.

In our works, two types of data sets are used: satellite data and aerial data. Table 1 displays one aerial corpus (ISPRS Vaihingen data set) and two satellite data sets (TH-Isan Landsat-8 and TH-North Landsat-8 data sets). The Vaihingen data set contains 16 patches. Such data have been collected at particular locations with different sizes of resolution.

**Table 1.** Numbers of training, validation, and testing sets.

| Data Set | Total Images | Training Set | Validation Set | Testing Set |
|---|---|---|---|---|
| TH-Isan Landsat-8 Corpus | 1420 | 1000 | 300 | 120 |
| TH-North Landsat-8 Corpus | 1600 | 1000 | 400 | 200 |
| ISPRS Vaihingen Corpus | 16 (Patches) | 10 | 2 | 4 |

#### 2.2.1. North East (Isan) and North of Thailand Landsat-8 Corpora

The Isan district of Thailand's northeast is characterized by gently undulating topography, which mostly ranges in altitude from 90 to 180 m (300 to 600 feet), sloping from the Phetchabun Mountains in the west down to the Mekong River. The plateau is separated into different plains: the Mun and Chi rivers drain the southern Khorat plain, while the Loei and Songkhram rivers drain the northern Sakon Nakhon plain. The two tables are divided by the Phu Phan Mountains. The land is primarily sandy, with a great deal of salt deposits.

The north of Thailand is known for its varied landforms: low hills, crisscrossing mountains and valleys, with a large area of land suitable for cultivation, such as corn, pineapple, and para rubber. The Ping, Wang, Yom, and Nan rivers travel south through mountain valleys before uniting to form the Chao Phraya in Nakhon Sawan Province in central Thailand.

All the images in this data set were captured in Thailand's northern and Isan regions (Changwat). The Landsat-8 satellite contributed to the data gathering, which included 1420 and 1600 satellite images for the north data set and Isan data set, respectively, as seen in sampled data as in Figures 3 and 4. This data set has a massive collection of (46,128 $\times$ 47,616) pixel medium-quality images; corn (yellow), para rubber (red), and pineapple (green) are the three classes. A total of 1420 images are divided into 1000 training, 300 validation images, and 120 test images for the northern corpus. A total of 1600 images are divided into 1000 training, 400 validation images, and 200 test images for the Isan corpus for comparability with other baseline methods.

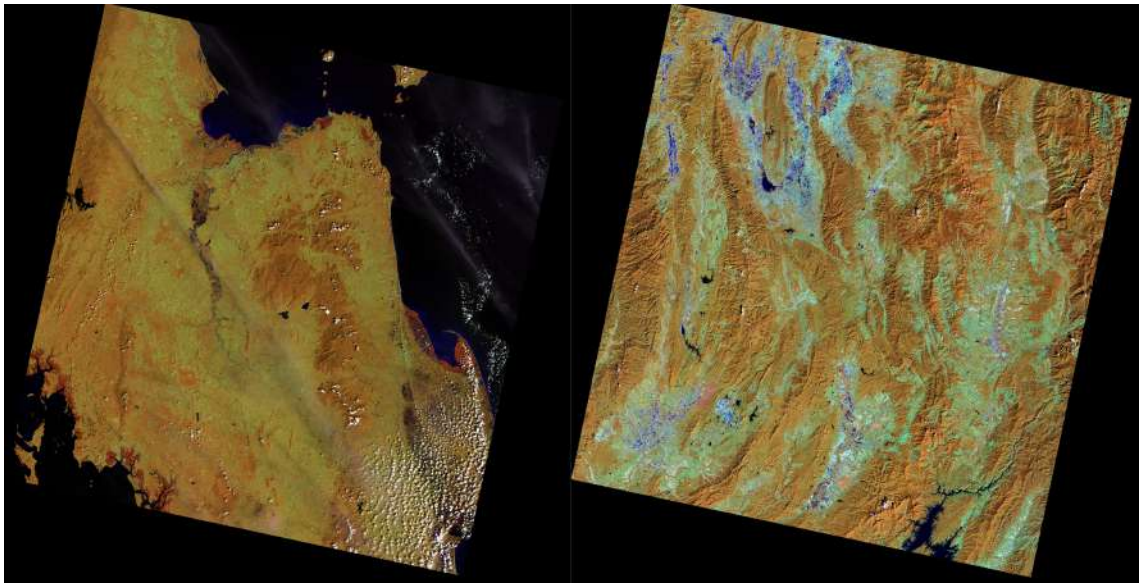


**Figure 3.** An illustration of a Landsat-8 scene (northern province (**left**) and northeastern region (**right**).

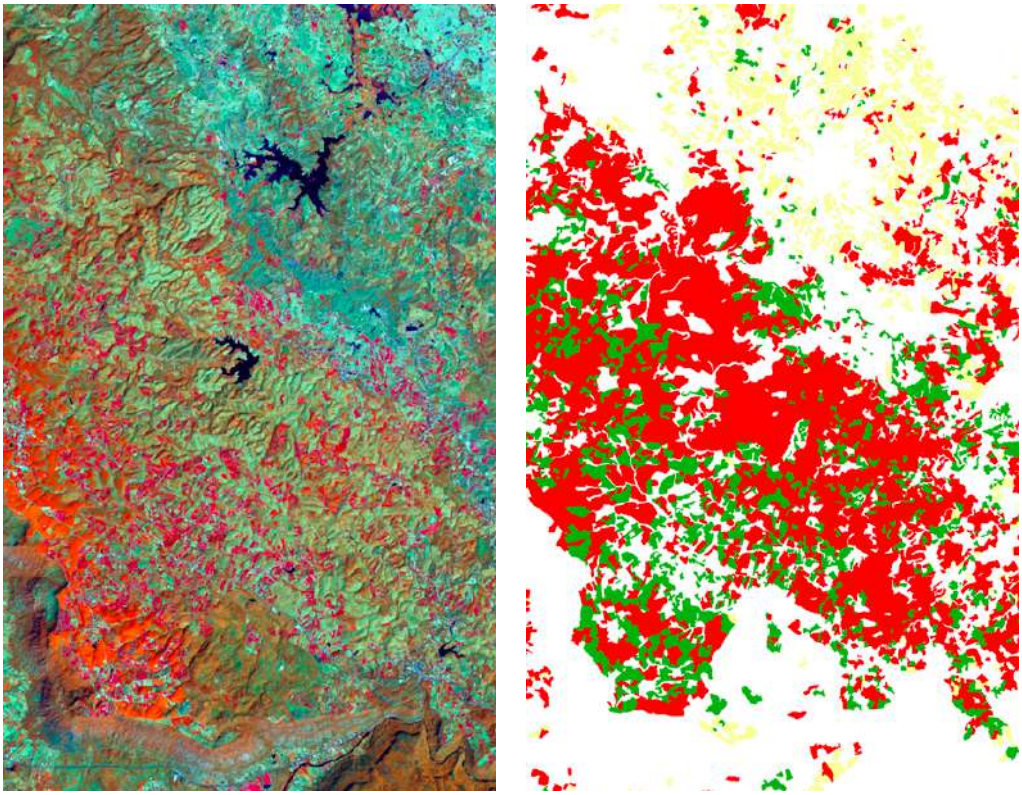


**Figure 4.** The left image is a sample of the northern province, and the right is the target image from the TH-Isan Landsat-8 corpus. Three classes comprise the target of the medium-resolution data set: para rubber (red), corn (yellow), and pineapple (green).

#### 2.2.2. ISPRS Vaihingen Corpus

Our benchmark dataset is the ISPRS semantic segmentation challenge [32] (Figures 5 and 6) in Vaihingen (Stuttgart). They seized command of the German city of Vaihingen. The ISPRS Vaihingen corpus contains 3-band IRRG (Red, Infrared, and Green) image data, corresponding NDSM (Normalized Digital Surface Model), and DSM (Digital Surface Model) data. The latter highlights 33 scenes with a resolution of approximately 2500 $\times$ 2000 pixels and a capacity of

about 9 cm. According to prior approaches, four locations, such as scenes 5, 7, 23, and 30, were eliminated from the training set as a testing set.


**Figure 5.** Very high-resolution imagery: ISPRS Vaihingen data set.

2.2.3. Evaluation Metrics

A true negative (TN) is an outcome where the model predicts the negative class correctly. Similarly, a true positive (TP) is an outcome where the model correctly predicts the positive class. A false negative (FN) is an outcome where the model incorrectly predicts the negative class, and a false positive (FP) is an outcome where the model incorrectly predicts the positive class.

$F1$ is the weighted average of Precision and Recall. Accordingly, this score needs both false negatives and false positives to verify the calculation. However, its *Accuracy* is not straightforward. Although $F1$ is regularly more valuable than *Accuracy*, especially with an uneven class distribution, *Accuracy* is achieved only if false positives and false negatives have similar costs.

It is noted that, for all corpora, the performance of "Pretrained SwinTF with decoder designs" is assessed for $F1$ and *Accuracy*. The *Intersection ovenion* $(IoU)$, $F1$, *Precision*, *recall*, and *Accuracy* metrics are used to evaluate class-specific performance; the symphonious average of recall, and accuracy is used to calculate it. The core metrics of *Precision*, *Recall*, $IoU$, $F1$ as well as the *Accuracy*, which divides the number of properly categorized locations by the total number of reference positions are all implemented. Applying Equations (4)–(8), the *Accuracy*, $IoU$, and $F1$ metrics can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

$$Intersection\ over\ Union\ (IoU) = \frac{TP}{TP + FP + FN} \tag{5}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

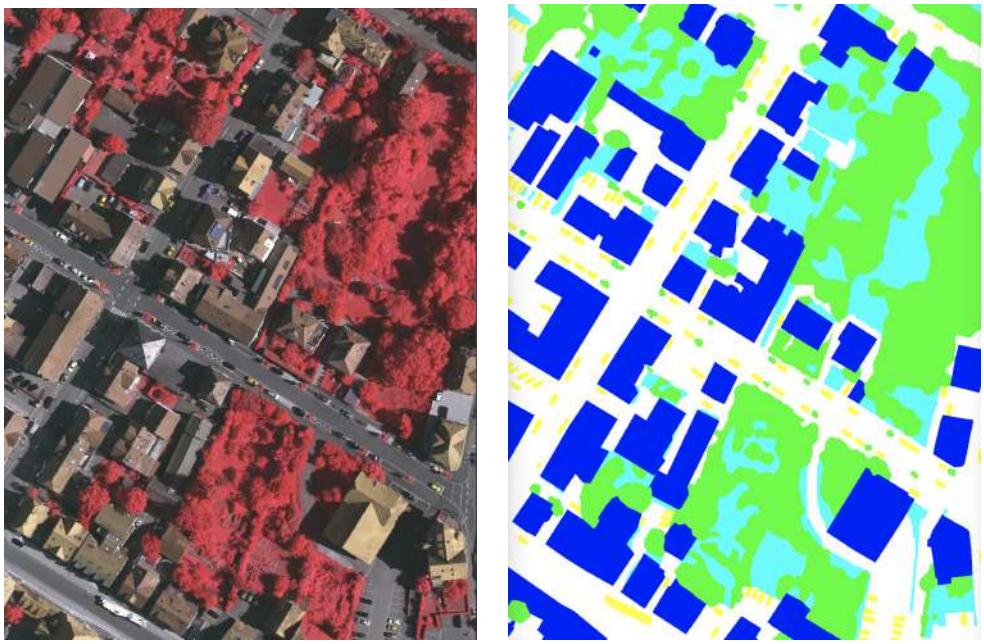


**Figure 6.** This is an example scene from Figure 5. The input image (**left**) depicts an example of an input scene and a target image (**right**). Tree (green), building (blue), jumble/background (red), low vegetation or LV (greenish-blue), and impervious surface or IS (white) are the five categories in the annotated Vaihingen data set.

## 3. Results

Regarding the DL environmental setup, the "TensorFlow Core v2.6.0 (TF)" [42] was created as an end-to-end open-source platform. All experiments were carried out via servers with Intel® Xeon® Scalable 4210R (10 core, 2.4 GHz, 13.75 MB, 100W), 256 GB of memory, and the NVIDIA RTX™ 1080Ti (11 GB) × 2 cards. As designated in Table 2, there are eight procedural acronyms in all proposed designs.

**Table 2.** Acronyms on our proposed scheme strategies.

| Acronym | Representation |
|---|---|
| DL | Deep Learning |
| FPN | Feature Pyramid Network |
| LR | Learning Rate |
| PSP | Pyramid Scene Parsing Network |
| ResNet152 | 152-layer ResNet |
| SwinTF | Swin Transformer |
| SwinTF-FPN | Swin Transformer with FPN Decoder Design |
| SwinTF-PSP | Swin Transformer with PSP Decoder Design |
| SwinTF-UNet | Swin Transformer with U-Net Decoder Design |
| TH-Isan Landsat-8 corpus | North East Thailand Landsat-8 data set |
| TH-North Landsat-8 corpus | North Thailand Landsat-8 data set |
| ViT | Vision Transformer |

### *3.1. Results for TH-Isan Landsat-8 Corpus*

#### 3.1.1. Effect of Swin Transformer and Pretrained Models

To ensure the contribution of the transformer module, SwinTF was compared with and without Pretrained models on ImageNet-1K by adding or removing the concatenation of this feature in our backbone architecture. The results presented in Tables 3 and 4 suggest that the Pretrained model on ImageNet-1K of Transformer is crucial for the segmentation.

In Table 3, the segmentation *F*1 scores are significantly improved by 3.4% for the backbone networks as compared with SwinTF without Pretrained and GCN-A-FF-DA with Res152.

Furthermore, in Table 4, the impact on the corn is 19.82%; this feature is due to the higher accuracy for almost all classes except the para rubber class. *F*1 scores of 87.74% can still be achieved with the same backbone networks in Table 3 as compared with SwinTF without Pretrained and GCN-A-FF-DA with Res152. Results suggest that the network of the transformer was compatible with end-to-end deep learning.

#### 3.1.2. Effect of Transformer with Decoder Designs

To investigate the transformer-based decoder designs, we evaluate our deep architecture with FPN, PSP, and UNet. In Table 3 of our proposed methods, SwinTF-PSP decoder design (the best-proposed model) achieves an *F*1 score of 88.95%, with the FPN decoder design achieving an *F*1 score of 89.80% and with the U-Net decoder design achieving an *F*1 score of 88.30%. Using the same training schedule, our best-proposed model (SwinTF-PSP) significantly outperforms the baselines (GCN), achieving *F*1 of 6.4% and the baselines (Pretrained SwinTF), achieving *F*1 of 2.05% with a clear margin.

Moreover, the decoder designs of our transformers yield concretely better results than original pretrained Swin Transformers. In Table 3 comparing SwinTF-PSP with SwinTF with Pretrained, our best model (PSP decoder designs) achieves 0.14%, 3.85%, and 2.06% improvements for *precision*, *recall*, and *F*1, respectively.

**Table 3.** Results on our testing set: TH-Isan Landsat-8 corpus.

| | **Pretrained** | **Backbone** | **Model** | ***Precision*** | ***Recall*** | ***F*1** | ***IoU*** |
|---|---|---|---|---|---|---|---|
| Baseline | Yes | - | DeepLab V3 [8] | 0.7547 | 0.7483 | 0.7515 | 0.6019 |
| | Yes | - | UNet [29] | 0.7353 | 0.7340 | 0.7346 | 0.5806 |
| | Yes | - | PSP [30] | 0.7783 | 0.7592 | 0.7686 | 0.6242 |
| | Yes | - | FPN [31] | 0.7633 | 0.7688 | 0.7660 | 0.6208 |
| | Yes | Res152 | GCN-A-FF-DA [36] | 0.7946 | 0.7883 | 0.7909 | 0.6549 |
| | Yes | RestNest-K50-GELU | GCN-A-FF-DA [36,43] | 0.8397 | 0.8285 | 0.8339 | 0.7154 |
| | No | ViT | SwinTF [12,13,37] | 0.8778 | 0.8148 | 0.8430 | 0.7319 |
| | Yes | ViT | SwinTF [12,13,37] | 0.8925 | 0.8637 | 0.8774 | 0.7824 |
| **Proposed Method** | Yes | ViT | SwinTF-UNet | 0.8746 | 0.8955 | 0.8830 | 0.7936 |
| | Yes | ViT | SwinTF-PSP | 0.8939 | **0.9022** | **0.8980** | **0.8151** |
| | Yes | ViT | SwinTF-FPN | **0.8966** | 0.8842 | 0.8895 | 0.8025 |

**Table 4.** Results on our testing set: TH-Isan Landsat-8 corpus (each class).

| | **Pretrained** | **Backbone** | **Model** | **Corn** | **Pineapple** | **Para Rubber** |
|---|---|---|---|---|---|---|
| Baseline | Yes | - | DeepLab V3 [8] | 0.6334 | 0.8306 | 0.7801 |
| | Yes | - | UNet [29] | 0.6210 | 0.8129 | 0.7927 |
| | Yes | - | PSP [30] | 0.6430 | 0.8170 | 0.8199 |
| | Yes | - | FPN [31] | 0.6571 | 0.8541 | 0.8191 |
| | Yes | Res152 | GCN-A-FF-DA [36] | 0.6834 | 0.8706 | 0.8301 |
| | Yes | RestNest-K50-GELU | GCN-A-FF-DA [36,43] | 0.8982 | 0.9561 | 0.8657 |
| | No | ViT | SwinTF [12,13,37] | 0.7021 | 0.9179 | 0.8859 |
| | Yes | ViT | SwinTF [12,13,37] | 0.9003 | 0.9572 | 0.8763 |
| **Proposed Method** | Yes | ViT | SwinTF-UNet | 0.9139 | 0.9652 | 0.8876 |
| | Yes | ViT | SwinTF-PSP | **0.9386** | 0.9632 | **0.8985** |
| | Yes | ViT | SwinTF-FPN | 0.9234 | **0.9619** | 0.8886 |

### *3.2. Results for TH-North Landsat-8 Corpus*

#### 3.2.1. Effect of Swin Transformer and Pretrained Models

As presented in Tables 5 and 6, the results suggest that the Pretrained model on ImageNet-1K of Transformer proved significant for the segmentation. The results greatly improved the segmentation *F*1 score by 1.06% for the backbone networks and 4.8% for the baseline networks. Furthermore, there was little impact on the para rubber, corn, and pineapple (2%); this feature was due to the higher accuracy for all classes in Table 6. In Table 5, an *F*1 score of 88.73% with the same backbone networks can still be achieved as compared with SwinTF without Pretrained and GCN-A-FF-DA with Res152. This outcome suggests that the network architecture of the transformer was compatible with end-to-end DL.

#### 3.2.2. Effect of Transformer with our Decoder Designs

To examine the transformer-based decoder designs, our deep architecture with FPN, PSP, and UNet was assessed. In Table 5 of our proposed methods, the SwinTF-PSP network (remaining the best-proposed model) achieved an *F*1 score of 63.12%. Further, the FPN decoder design achieved an *F*1 score of 63.06%, and the UNet decoder design achieved an *F*1 score of 62.24%.

**Table 5.** Results on our testing set: TH-North Landsat-8 corpus.

| | Pretrained | Backbone | Model | *Precision* | *Recall* | *F*1 | *IoU* |
|---|---|---|---|---|---|---|---|
| Baseline | Yes | - | DeepLab V3 [8] | 0.5019 | 0.5323 | 0.5166 | 0.3483 |
| | Yes | - | UNet [29] | 0.4836 | 0.5334 | 0.5073 | 0.3398 |
| | Yes | - | PSP [30] | 0.4949 | 0.5456 | 0.5190 | 0.3505 |
| | Yes | - | FPN [31] | 0.5112 | 0.5273 | 0.5192 | 0.3506 |
| | Yes | Res152 | GCN-A-FF-DA [36] | 0.5418 | 0.5722 | 0.5559 | 0.3857 |
| | Yes | RestNest-K50-GELU | GCN-A-FF-DA [36,43] | 0.6029 | 0.5977 | 0.5977 | 0.4289 |
| | No | ViT | SwinTF [12,13,37] | 0.6076 | 0.5809 | 0.5940 | 0.4225 |
| | Yes | ViT | SwinTF [12,13,37] | 0.6233 | 0.5883 | 0.6047 | 0.4340 |
| **Proposed Method** | Yes | ViT | SwinTF-UNet | 0.6273 | 0.6177 | 0.6224 | 0.4519 |
| | Yes | ViT | SwinTF-PSP | **0.6384** | 0.6245 | **0.6312** | **0.4613** |
| | Yes | ViT | SwinTF-FPN | 0.6324 | **0.6289** | 0.6306 | 0.4606 |

**Table 6.** Results on our testing set: TH-North Landsat-8 corpus (each class).

| | Pretrained | Backbone | Model | Corn | Pineapple | Para Rubber |
|---|---|---|---|---|---|---|
| Baseline | Yes | - | DeepLab V3 [8] | 0.4369 | 0.8639 | 0.8177 |
| | Yes | - | UNet [29] | 0.4135 | 0.8418 | 0.7721 |
| | Yes | - | PSP [30] | 0.4413 | 0.8702 | 0.8032 |
| | Yes | - | FPN [31] | 0.4470 | 0.8743 | 0.8064 |
| | Yes | Res152 | GCN-A-FF-DA [36] | 0.4669 | 0.9039 | 0.8177 |
| | Yes | RestNest-K50-GELU | GCN-A-FF-DA [36,43] | 0.5151 | 0.9394 | 0.8442 |
| | No | ViT | SwinTF [12,13,37] | 0.5375 | 0.9302 | 0.8628 |
| | Yes | ViT | SwinTF [12,13,37] | 0.5592 | 0.9527 | 0.8873 |
| **Proposed Method** | Yes | ViT | SwinTF -UNet | 0.5850 | 0.9703 | 0.9117 |
| | Yes | ViT | SwinTF-PSP | **0.6008** | **0.9877** | **0.9296** |
| | Yes | ViT | SwinTF-FPN | 0.6006 | 0.9857 | 0.9245 |

Using the same training schedule, our best-proposed model (SwinTF-PSP) significantly outperformed both baselines (GCN), achieving an *F*1 score of 7.52% and the baselines (Pretrained Swin-TF), achieving an *F*1 score of 2.65% by a clear margin. It is evident that the decoder designs of our transformers yielded far better results than the original pretrained

Swin Transformers. In Table 5 comparing SwinTF-PSP with SwinTF with Pretrained, our best model (PSP decoder designs) achieved 1.5%, 3.6%, and 2.6% improvements for *precision*, *recall*, and *F*1, respectively.

### *3.3. Results for ISPRS Vaihingen Corpus*

This research aims to take semantic segmentation methods via modern deep learning and apply them to high-resolution geospatial corpora. These differences are summed up in Tables 7 and 8. We noted that our best model (Pretrained SwinTF-FPN) had more robust results on this corpus.

#### 3.3.1. Effect of Swin Transformer and Pretrained Models

In Tables 7 and 8, the results suggest that the Pretrained model on ImageNet-1K of Transformer is also significant for the segmentation. In Table 7, the results much improved the segmentation *F*1-score by 2.73% for the backbone networks and 8.01% for the baseline networks compared with SwinTF without Pretrained and GCN-A-FF-DA with Res152. In Table 8, there was little impact on the impervious surfaces, tree, and car classes at 2.19%, 2.48%, and 9.39%, respectively; this feature was due to the higher accuracy almost of all classes. It is clear that SwinTF-FPN can still achieve *F*1 scores of 94.94% with the same backbone network in Table 7. This result suggests that the network architecture of the transformer was compatible with end-to-end deep learning.

#### 3.3.2. Effect of Transformer with Our Decoder Designs

To investigate the transformer-based decoder designs, our deep architecture was evaluated via FPN, PSP, and UNet, respectively. In Table 7 of our proposed methods, SwinTF-PSP also achieved an *F*1 score of 94.83%. Furthermore, the FPN decoder design (the winner) achieved an *F*1 score of 94.94%, whilst the UNet decoder design achieved an *F*1 score of 94.38%. Using the same training schedule, our best-proposed model (SwinTF-FPN) significantly outperformed both the baselines (GCN), achieving *F*1 of 6.4% and the baselines (Pretrained SwinTF), achieving *F*1 of 2.05% by a clear margin.

**Table 7.** Results on our testing set: ISPRS Vaihingen Corpus.

| | **Pretrained** | **Backbone** | **Model** | ***Precision*** | ***Recall*** | ***F*1** | ***IoU*** |
|---|---|---|---|---|---|---|---|
| Baseline | Yes | - | DeepLab V3 [8] | 0.8672 | 0.8672 | 0.8672 | 0.7656 |
| | Yes | - | UNet [29] | 0.8472 | 0.8572 | 0.8522 | 0.7425 |
| | Yes | - | PSP [30] | 0.8614 | 0.8799 | 0.8706 | 0.7708 |
| | Yes | - | FPN [31] | 0.8701 | 0.8812 | 0.8756 | 0.7787 |
| | Yes | Res152 | GCN-A-FF-DA [36] | 0.8716 | 0.8685 | 0.8694 | 0.8197 |
| | Yes | RestNest-K50-GELU | GCN-A-FF-DA [36,43] | 0.9044 | 0.9088 | 0.9063 | 0.8292 |
| | No | ViT | SwinTF [12,13,37] | 0.8537 | 0.9356 | 0.8770 | 0.7701 |
| | Yes | ViT | SwinTF [12,13,37] | 0.9756 | 0.8949 | 0.9221 | 0.8753 |
| **Proposed Method** | Yes | ViT | SwinTF-UNet | 0.9203 | 0.9732 | 0.9438 | 0.8977 |
| | Yes | ViT | SwinTF-PSP | 0.9271 | **0.9820** | **0.9483** | **0.9098** |
| | Yes | ViT | SwinTF-FPN | **0.9296** | 0.9756 | 0.9494 | 0.9086 |

Moreover, the decoder designs of our transformers yielded much better results than the original pretrained SwinTF. In Table 7, comparing SwinTF-PSP with SwinTF and Pretrained, our best model (FPN decoder design) attained 8.07% and 2.76% improvements for the recall and *F*1, respectively.

Figure 7 shows the prediction results for the entire Isan scene, and Figure 8 shows the prediction results for the entire North scene. Agriculture regions are more dispersed in these zones, and the scenery is more varied. Furthermore, exposed rocks and patches of flora in semiarid environments may have comparable backscatter intensities to structures and be readily misclassified.

**Table 8.** Results on our testing set: ISPRS Vaihingen Corpus (each class).

| | Model | IS | Buildings | LV | Tree | Car |
|---|---|---|---|---|---|---|
| Baseline | DeepLab V3 [8] | 0.8289 | 0.8026 | 0.8257 | 0.7985 | 0.6735 |
| | UNet [29] | 0.8189 | 0.7826 | 0.7857 | 0.7845 | 0.6373 |
| | PSP [30] | 0.8273 | 0.8072 | 0.8059 | 0.8050 | 0.6781 |
| | FPN [31] | 0.8327 | 0.8111 | 0.8127 | 0.8117 | 0.6896 |
| | GCN-A-FF-DA [36] | 0.8431 | 0.8336 | 0.8362 | 0.8312 | 0.7014 |
| | GCN-A-FF-DA [36,43] | 0.9005 | 0.9076 | **0.8942** | 0.8877 | 0.8233 |
| | SwinTF [12,13,37] | 0.8811 | 0.8934 | 0.8878 | 0.8734 | 0.7866 |
| | Pretrained SwinTF [12,13,37] | 0.9137 | 0.9139 | 0.8803 | 0.8922 | 0.8118 |
| **Proposed Method** | Pretrained SwinTF-UNet | 0.9139 | 0.9101 | 0.8870 | 0.9035 | 0.9006 |
| | Pretrained SwinTF-PSP | 0.9259 | **0.9195** | 0.8790 | 0.9093 | 0.9019 |
| | Pretrained SwinTF-FPN | **0.9356** | 0.9157 | 0.8746 | **0.9169** | **0.9057** |

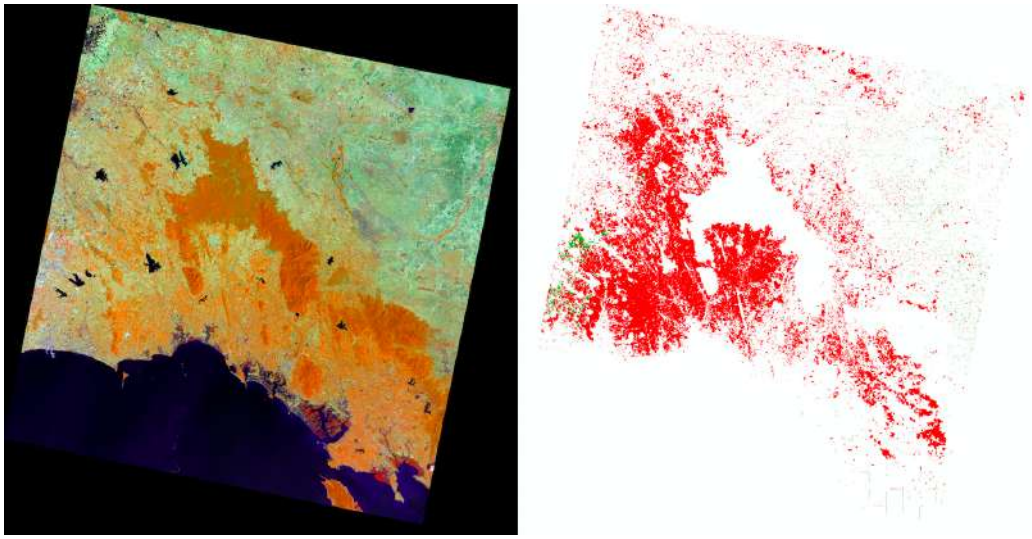


**Figure 7.** Prediction result of "Pretrained SwinTF-PSP" on the entire TH-Isan Landsat-8 corpus scene.

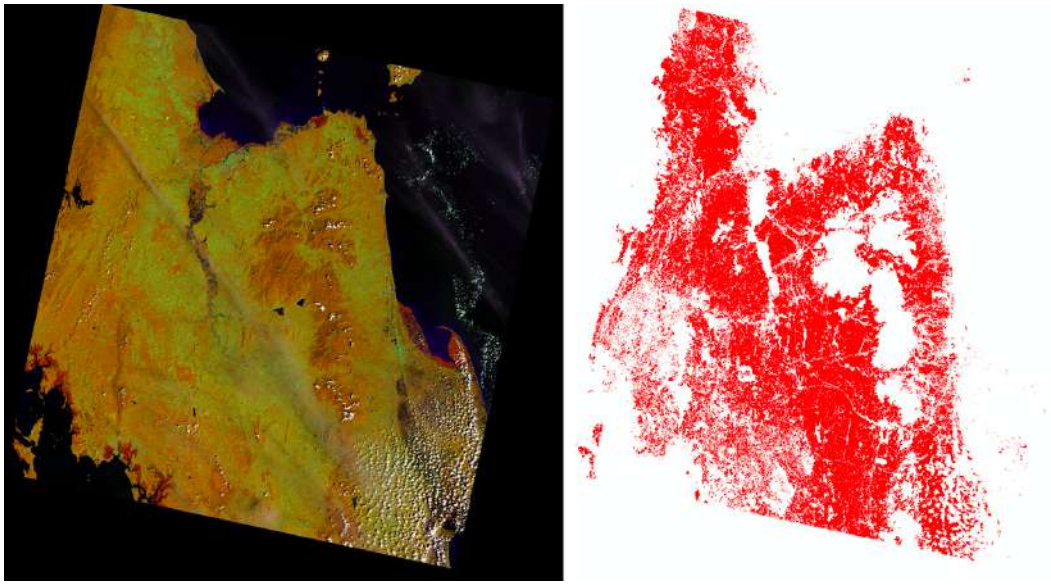


**Figure 8.** Prediction result of "Pretrained SwinTF-PSP" on the entire TH-North Landsat-8 corpus scene.

## 4. Discussion

In this work, the usefulness of SwinTF-based semantic segmentation models for the retrospective reconstruction of Thailand's agriculture region was investigated. For the chosen methodologies, the necessity to prepare sufficient training data may provide certain restrictions.

As shown in Figure 9, the Isan Landsat-8 corpus provided a qualitative segmentation comparison between SwinTF and decoder designs (SwinTF-PSP, SwinTF-FPN, and SwinTF-UNet) and the SOTA baseline (an enhanced GCN). The results of the PSP decoder design demonstrated more precise segmentation for bigger and thinner objects, e.g., the para rubber and corn areas. Moreover, the PSP decoder design also achieved more integrated segmentation on smaller objects, e.g., the pineapple class.

In the validation data of "SwinTF-PSP", Figure 10, there is a more profound disparity (hill) than that in the baseline, Figure 10a. In addition, Figures 10b and 11b show four learning graphs viz. *accuracy*, *precision*, *recall*, and *F*1 lines. The loss line of the "SwinTF-PSP" model appeared deceived (very soft) more than the traditional method in Figure 11a. The number at epoch 99 was selected as a pretrained weight for validation and transfer learning procedures.

As shown in Figure 12, the north Landsat-8 corpus provided a qualitative segmentation comparison between SwinTF and decoder designs (SwinTF-PSP, SwinTF-FPN, and SwinTF-UNet) and the SOTA baseline (an enhanced GCN). The results of the PSP decoder design revealed more precise segmentation for smaller objects, such as the corn area. Moreover, the PSP decoder design also achieved more integrated segmentation on oversized objects, e.g., the para rubber class.

There is a more profound disparity (hill) in the validation data of "SwinTF-PSP", Figure 13, than that in the baseline, Figure 13a. In addition, Figures 13b and 14b depict the four learning lines viz. *accuracy*, *precision*, *recall*, and *F*1 lines. The loss line of the "SwinTF-PSP" model appeared deceived (very soft) more than the traditional method in Figure 14a. The number at epoch 100 was chosen as a pretrained weight for validation and transfer learning procedures.

As shown in Figure 15, the ISPRS Vaihingen corpus provided qualitative segmentation comparison between SwinTF and decoder designs (SwinTF-PSP, SwinTF-FPN, and SwinTF-UNet) and the SOTA baseline (an enhanced GCN). The results of the FPN decoder design exhibited more precise segmentation for smaller objects, e.g., the car and tree (classes). Moreover, the PSP decoder design achieved more integrated segmentation on oversized objects, e.g., impervious surfaces. The number at epoch 99 was picked as a pretrained weight for validation and transfer learning procedures.

In Figure 16, there was a more profound disparity (hill) in the validation data of "SwinTF-PSP" than that in the baseline, Figure 16a. In addition, Figures 16b and 17b show the four learning lines viz. *accuracy*, *precision*, *recall*, and *F*1 lines. The loss line of the "SwinTF-PSP" model appeared deceived (very soft) more than the traditional method in Figure 17a. The number at epoch 95 was picked as a pretrained weight for validation and transfer learning procedures.

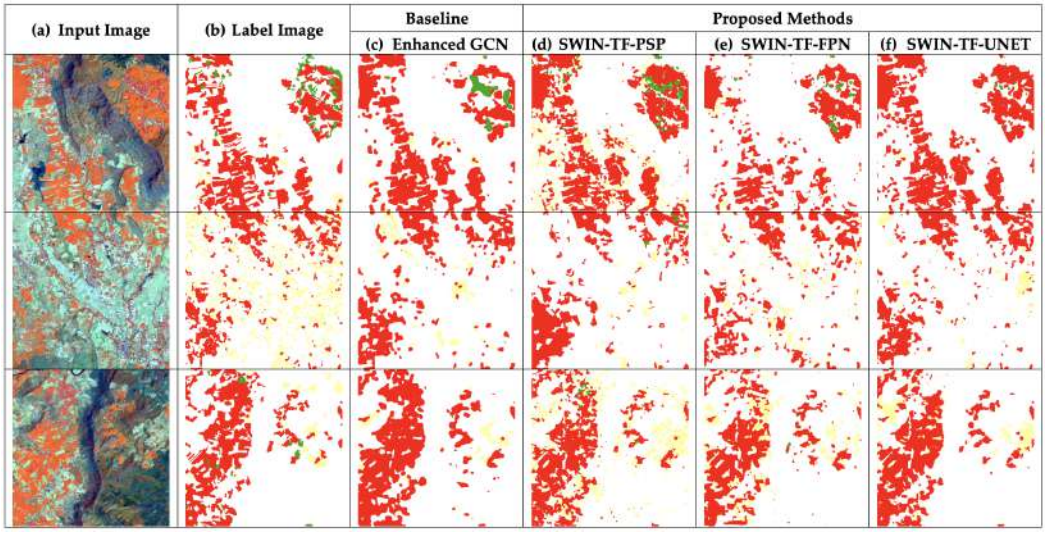
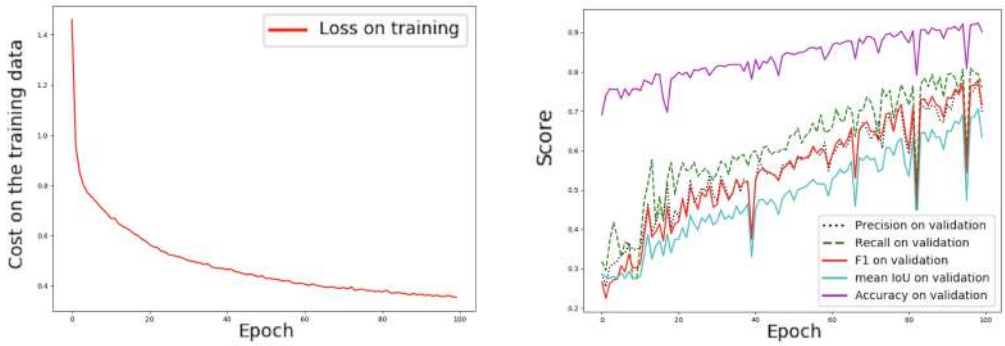
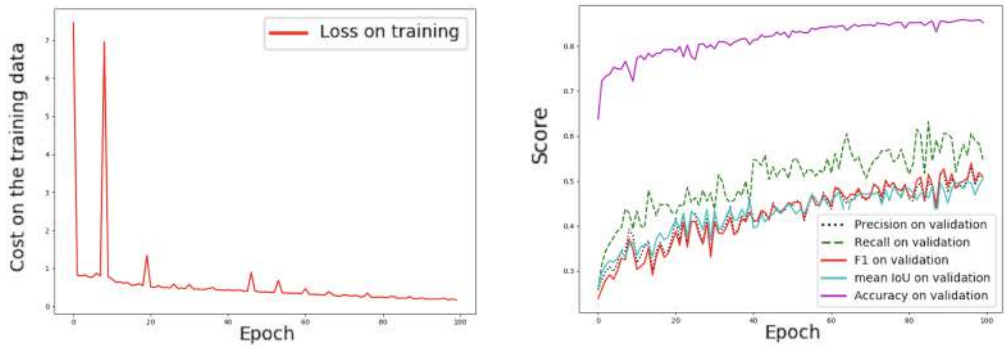



**Figure 9.** Comparisons between our proposed methods and baseline for the TH-Isan Landsat-8 corpus testing set.




(**a**) (**b**)

**Figure 10.** Graph (learning curves): on TH-Isan Landsat-8, the proposed approach, and SwinTF-PSP (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.




(**a**) (**b**)

**Figure 11.** Graph (learning curves): TH-Isan Landsat-8 corpus, the baseline approach, and SwinTF (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.

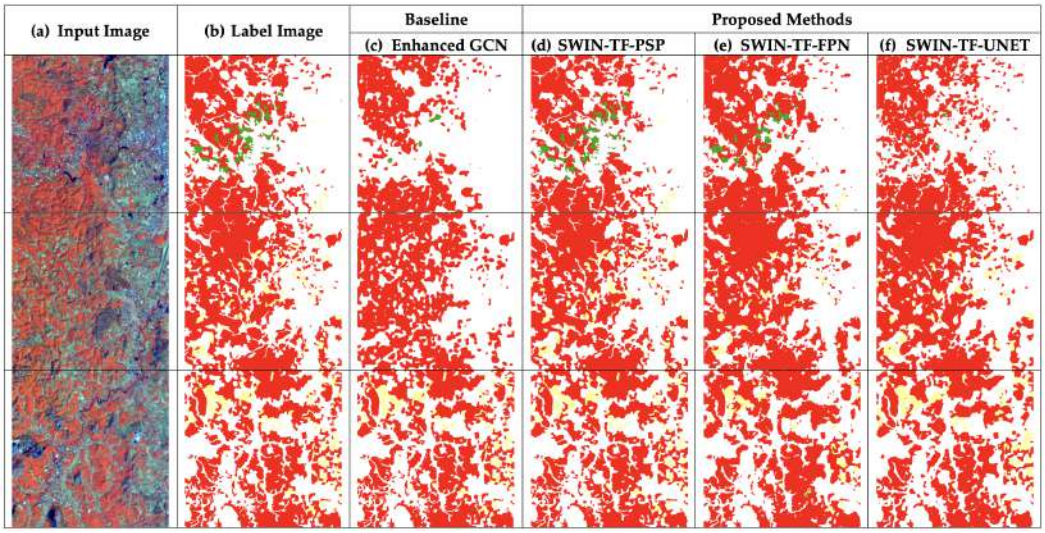
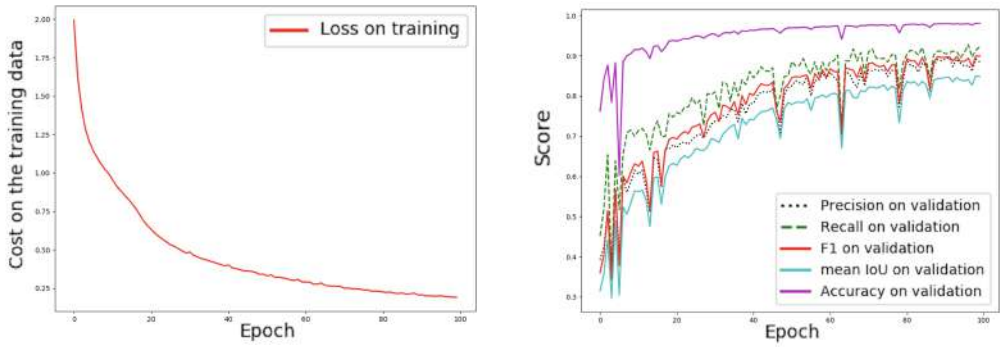
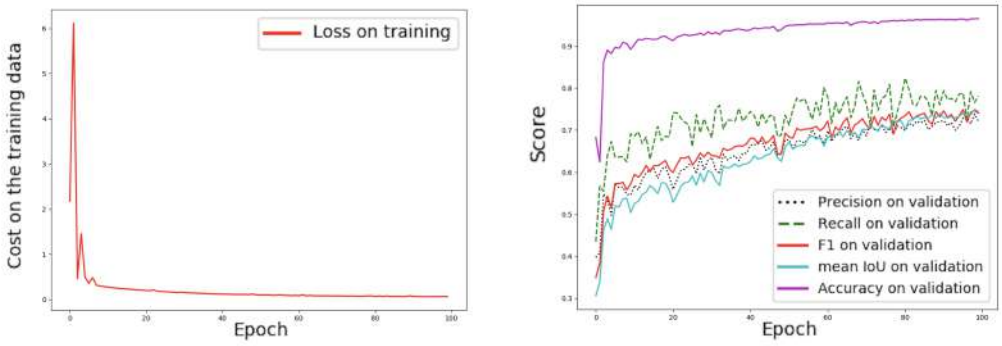



**Figure 12.** Comparisons between our proposed methods and baseline for the TH-North Landsat-8 corpus testing set.




(**a**) (**b**)

**Figure 13.** Graph (learning curves): TH-North Landsat-8 corpus, the proposed approach, and SwinTF-PSP (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.




(**a**) (**b**)

**Figure 14.** Graph (learning curves): TH-North Landsat-8 corpus, the baseline approach, and SwinTF (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.

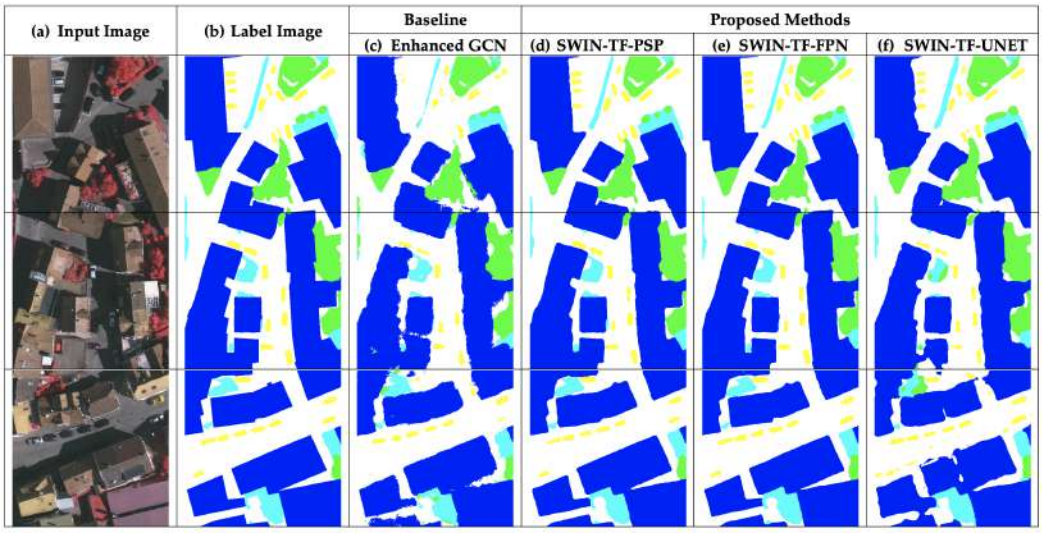
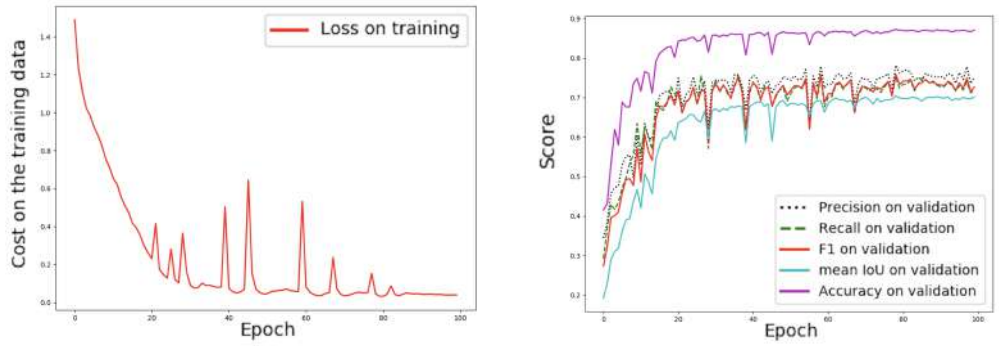
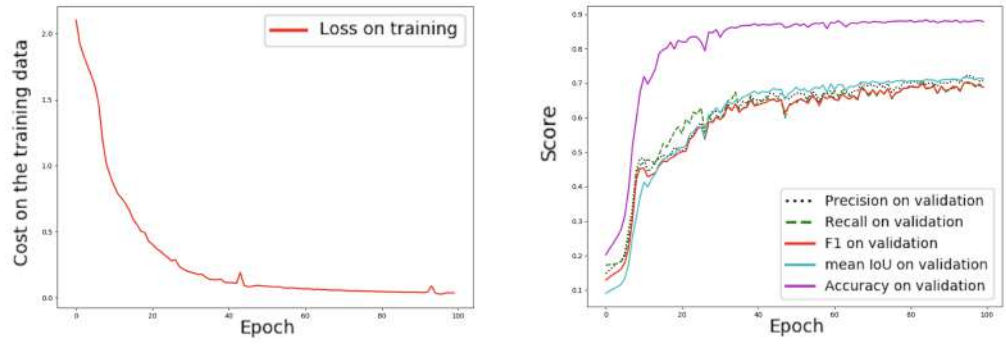



**Figure 15.** Comparisons between our proposed methods and baseline for the ISPRS Vaihingen corpus testing set.




(**a**) (**b**)

**Figure 16.** Graph (learning curves): ISPRS Vaihingen corpus, the proposed approach, and SwinTF-FPN (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.




(**a**) (**b**)

**Figure 17.** Graph (learning curves): ISPRS Vaihingen corpus, the baseline approach, and SwinTF (**a**) Plot of model loss (cross-entropy) on training and testing corpora; (**b**) performance plot on the testing corpus.

*Limitations and Outlook*

In this research, the appropriateness of transformer-based semantic segmentation models for the retrospective reconstruction of cultivation (corn, pineapple, and para rubber) in Thailand as well as the ISPRS Vaihingen data set (aerial images) was investigated. For

the selected methods, the requirement to prepare extensive training data may pose some limitations. For future studies, achieving high performance with limited training data using our approach must be cost-effective for multi-temporal agriculture mapping.

Therefore, further evaluation of the effectiveness of using modern DL methods with Landsat-8 data beyond a national scale is required. Notwithstanding some limitations, this study adds a baseline, including DeepLab v3, PSP, FPN, and UNet, for proving our best model performance. In future work, more varieties of modern image labeling, as well as some analytical perspectives, e.g., evolving reinforcement learning (RL) algorithms, generative adversarial networks (GANs), or quantization methods for efficient neural network inference, will be reviewed and analyzed for such tasks.

## 5. Conclusions

This paper exhibits an alternative viewpoint for semantic segmentation by prefacing decoder designs for transformer models. The experimental results show that (1) the pretrained transformer models on ImageNet-1K achieved good results for both the Landsat-8 (medium resolution) and ISPRS Vaihingen corpus (very high-resolution). The *F*1-scores were found to range from 84.3% to 87.74%, 59.4% to 64.47%, and 87.7% to 92.21% for the Isan, Nan, and ISPRS Vaihingen corpora, respectively. (2) Our results were compared with other decoder design methods, including FPN, PSP, and U-Net.

It is evident that the proposed approach proved its worth as a dependable technique. Our detailed qualitative and quantitative investigations on three complex remote sensing tasks revealed that both FPN and PSP decoder designs consistently outperformed the baselines and state-of-the-art techniques, thus, demonstrating their significant efficacy and capabilities. In addition, the average accuracy was better than 90% for almost all classes of the data sets.

**Author Contributions:** Conceptualization, T.P.; Formal analysis, T.P.; Investigation, T.P.; Methodology, T.P.; Project administration, T.P.; Resources, T.P.; Software, T.P.; Supervision, T.P. and P.V.; Validation, T.P., K.J., S.L. and P.S.; Visualization, T.P.; Writing—original draft, T.P.; Writing—review and editing, T.P. and P.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Ratchadapisek Somphot Fund for Postdoctoral Fellowship, Chulalongkorn University.

**Acknowledgments:** Teerapong Panboonyuen, also known as Kao Panboonyuen appreciates (thanks) and acknowledges the scholarship from Ratchadapisek Somphot Fund for Postdoctoral Fellowship, Chulalongkorn University, Thailand.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Li, R.; Zheng, S.; Zhang, C.; Duan, C.; Su, J.; Wang, L.; Atkinson, P.M. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Trans. Geosci. Remote. Sens.* **2021**. [CrossRef]
2. Yang, N.; Tang, H. Semantic Segmentation of Satellite Images: A Deep Learning Approach Integrated with Geospatial Hash Codes. *Remote Sens.* **2021**, *13*, 2723. [CrossRef]
3. Li, H.; Qiu, K.; Chen, L.; Mei, X.; Hong, L.; Tao, C. SCAttNet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 905–909. [CrossRef]
4. Li, X.; Xu, F.; Lyu, X.; Gao, H.; Tong, Y.; Cai, S.; Li, S.; Liu, D. Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images. *Int. J. Remote Sens.* **2021**, *42*, 3583–3610. [CrossRef]
5. Chen, Z.; Li, D.; Fan, W.; Guan, H.; Wang, C.; Li, J. Self-attention in reconstruction bias U-Net for semantic segmentation of building rooftops in optical remote sensing images. *Remote Sens.* **2021**, *13*, 2524. [CrossRef]
6. Tasar, O.; Giros, A.; Tarabalka, Y.; Alliez, P.; Clerc, S. Daugnet: Unsupervised, multisource, multitarget, and life-long domain adaptation for semantic segmentation of satellite images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1067–1081. [CrossRef]
7. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4353–4361.
8. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]

9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
10. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
11. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 603–612.
12. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
16. Qing, Y.; Liu, W.; Feng, L.; Gao, W. Improved Transformer Net for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 2216. [CrossRef]
17. Sun, Z.; Cao, S.; Yang, Y.; Kitani, K.M. Rethinking transformer-based set prediction for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 3611–3620.
18. Yang, F.; Zhai, Q.; Li, X.; Huang, R.; Luo, A.; Cheng, H.; Fan, D.P. Uncertainty-Guided Transformer Reasoning for Camouflaged Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 4146–4155.
19. Wang, L.; Li, R.; Wang, D.; Duan, C.; Wang, T.; Meng, X. Transformer Meets Convolution: A Bilateral Awareness Network for Semantic Segmentation of Very Fine Resolution Urban Scene Images. *Remote Sens.* **2021**, *13*, 3065. [CrossRef]
20. Jin, Y.; Han, D.; Ko, H. TrSeg: Transformer for semantic segmentation. *Pattern Recognit. Lett.* **2021**, *148*, 29–35. [CrossRef]
21. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-trained image processing transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12299–12310.
22. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 12179–12188.
23. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 16519–16529.
24. Kim, K.; Wu, B.; Dai, X.; Zhang, P.; Yan, Z.; Vajda, P.; Kim, S.J. Rethinking the Self-Attention in Vision Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3071–3075.
25. Salvador, A.; Gundogdu, E.; Bazzani, L.; Donoser, M. Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 15475–15484.
26. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, PMLR, Online, 18–24 July 2021; pp. 10347–10357.
27. Lin, A.; Chen, B.; Xu, J.; Zhang, Z.; Lu, G. DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation. *arXiv* **2021**, arXiv:2106.06716.
28. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv* **2021**, arXiv:2102.12122.
29. Li, X.; Chen, H.; Qi, X.; Dou, Q.; Fu, C.W.; Heng, P.A. H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE Trans. Med Imaging* **2018**, *37*, 2663–2674. [CrossRef]
30. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
31. Kim, S.W.; Kook, H.K.; Sun, J.Y.; Kang, M.C.; Ko, S.J. Parallel feature pyramid network for object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 234–250.
32. International Society for Photogrammetry and Remote Sensing. 2D Semantic Labeling Challenge. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 9 September 2018).
33. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
34. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]

35. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder–decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
36. 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. [CrossRef]
37. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2021; pp. 6881–6890.
38. Purkait, P.; Zhao, C.; Zach, C. SPP-Net: Deep absolute pose regression with synthetic views. *arXiv* **2017**, arXiv:1712.03452.
39. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
40. Qin, Z.; Zhang, Z.; Chen, X.; Wang, C.; Peng, Y. Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1363–1367.
41. Cheng, B.; Wei, Y.; Shi, H.; Feris, R.; Xiong, J.; Huang, T. Revisiting rcnn: On awakening the classification power of faster rcnn. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 453–468.
42. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the OSDI, Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
43. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.

information

MDPI

*Article*

# Object Detection of Road Assets Using Transformer-Based YOLOX with Feature Pyramid Decoder on Thai Highway Panorama

**Teerapong Panboonyuen [1,*], Sittinun Thongbai [2], Weerachai Wongweeranimit [3], Phisan Santitamnont [3,4], Kittiwan Suphan [2] and Chaiyut Charoenphon [4,*]**

[1] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand
[2] InfraPlus Co., Ltd., Phayathai Rd, Pathumwan, Bangkok 10330, Thailand; sittinun2tb@gmail.com (S.T.); kittiwan.sati@gmail.com (K.S.)
[3] Center of Excellence in Infrastructure Management, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand; weerachai.w@chula.ac.th (W.W.); phisan.S@eng.chula.ac.th (P.S.)
[4] Department of Survey Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand
* Correspondence: teerapong.panboonyuen@gmail.com (T.P.); chaiyut.c@chula.ac.th (C.C.)

**Abstract:** Due to the various sizes of each object, such as kilometer stones, detection is still a challenge, and it directly impacts the accuracy of these object counts. Transformers have demonstrated impressive results in various natural language processing (NLP) and image processing tasks due to long-range modeling dependencies. This paper aims to propose an exceeding you only look once (YOLO) series with two contributions: (i) We propose to employ a pre-training objective to gain the original visual tokens based on the image patches on road asset images. By utilizing pre-training Vision Transformer (ViT) as a backbone, we immediately fine-tune the model weights on downstream tasks by joining task layers upon the pre-trained encoder. (ii) We apply Feature Pyramid Network (FPN) decoder designs to our deep learning network to learn the importance of different input features instead of simply summing up or concatenating, which may cause feature mismatch and performance degradation. Conclusively, our proposed method (Transformer-Based YOLOX with FPN) learns very general representations of objects. It significantly outperforms other state-of-the-art (SOTA) detectors, including YOLOv5S, YOLOv5M, and YOLOv5L. We boosted it to 61.5% AP on the Thailand highway corpus, surpassing the current best practice (YOLOv5L) by 2.56% AP for the test-dev data set.

**Keywords:** deep learning; YOLO; YOLOX; Vision Transformer; highway scenes



**Citation:** Panboonyuen, T.; Thongbai, S.; Wongweeranimit, W.; Santitamnont, P.; Suphan, K.; Charoenphon, C. Object Detection of Road Assets Using Transformer-Based YOLOX with Feature Pyramid Decoder on Thai Highway Panorama. *Information* **2022**, *13*, 5. https://doi.org/10.3390/info13010005

Academic Editor: Zoran H. Peric

Received: 15 November 2021
Accepted: 22 December 2021
Published: 25 December 2021



## 1. Introduction

Identifying road asset objects in Thailand highway monitoring image sequences is essential for intelligent traffic monitoring and administration of the highway. With the widespread use of traffic surveillance cameras, an extensive library of traffic video footage has been available for examination. A more distant road surface may usually be evaluated from an eye-observing angle. At this viewing angle, the vehicle's object size varies enormously, and the detection accuracy of a small item far away from the road is low. In the face of complicated camera scenarios, it's critical to address and implement the difficulties listed above successfully. We will focus on the challenges mentioned earlier in this post and provide a suitable solution. This study applies the object detection findings for multi-object tracking and asset object counting, including kilometer signs (marked as KM Sign) and kilometer stones (marked as KM Stone).

Among modern Convolutional Neural Networks (ConvNet/CNNs), there are many techniques, e.g., dynamic heads with attentions [1], dual attention [2], self-attention [3]

have gained increasing attention due to their capability. Still, all suffer from accuracy performance issues.

Nowadays, many works [4–8] extensively used architectures and applied in road object detection such as You Only Look Once version 3 (YOLOv3) [9], Mask R-CNN [10], BiseNet [11], YOLOv4 [12], YOLOv5 [13], and/or YOLOX [14]. They are created for image recognition, consist of stacked Conv blocks. Due to anxieties about the cost of computation, the purpose of kernel maps is decreased gradually. Furthermore, the encoder network can learn more semantic visual theories with a steadily increased receptive field. Consequently, it also inflates a primary restriction of studying long-range dependency knowledge, significant for image labeling in unconstrained scene images. It matures challenging due to still limited receptive fields. However, the previous architecture has not fully leveraged various feature maps from convolution or attention blocks conducive to image segmentation, and this has become a motivation for this work.

To defeat this limitation, as mentioned earlier, a completely new architecture known as Vision Transformer YOLO (ViT-YOLO) [15] with ViT [16] as major backbone has a tremendous capacity in long-range dependency acquisition and sequence-based picture modeling. It is a vision model-based that is built as closely as possible on the Transformer architecture, which was created for text-based jobs in the first area [17]. Furthermore, it has matured famous in several computer vision tasks, such as hyperspectral image classification [18,19], bounding-box detection [20,21], and image labeling [22,23]. ViT moves the window divider between successive levels of self-attention. The shifted windows provide links between the windows of the last layer, considerably increasing modeling capability. In terms of real-world precision, this method is also effective.

In this work, prompted by the preceding observation, we introduce transformer-based Feature Pyramid Network (FPN) [24] decoder designs. It learns the importance of different input features instead of simply summing up or concatenating, which may cause feature mismatch and performance degradation as demonstrated in Figure 4 This work points to further improving the SOTA on object detection in Thailand highway road images. For better performance, we inject the FPN style of decoder design into Transformer-based YOLOX reasoning. In this article, our main contributions are twofold:

- Utilizing a pre-training ViT to retrieve the virtual visual tokens based on the vision patches on images. We immediately fine-tune the model weights on downstream responsibilities by appropriating pre-training ViT as a backbone of YOLOX [14] by appending responsibility layers superimposing the pre-trained encoder.
- We apply the Feature Pyramid Network (FPN) [24] as decoder designs on our Transformer-Based YOLOX. It adds a different bottom-up path aggregation architecture. Notably, when the deep architecture is relatively shallow, and the feature map is more significant, the transformer layer is used prematurely to enforce regression boundaries which can lose some meaningful context information.

The experimental results on Thailand highway road demonstrate the effectiveness of the proposed scheme. The results proved that our Transformer-Based YOLOX with FPN decoder designs overcomes YOLOv5S, YOLOv5M, and YOLOv5L based architectures [25,26] in terms of $AP$, $AP50$, and $AP75$ score sequentially.

This article is organized as follows. Section 2 discusses related work, and our data set is detailed in Section 3. Next, Section 4 provides the detail of our methodology, and Section 5 presents our experimental results. Finally, conclusions are drawn in Section 6.

## 2. Related Work

Most relevant to our methodology is the Vision Transformer (ViT) [16] and their follow-ups [27–31]. Vision Transformer (ViT) [16] is a deep learning architecture that utilizes the mechanism of attention, and there are many works that follow-ups ViT [8,27–31]. Several works of ViT directly employ a Transformer model on non-overlapping medium-sized image patches for image classification. It reaches an exciting speed-performance trade-off on almost all computer vision tasks compared to previous deep learning networks.

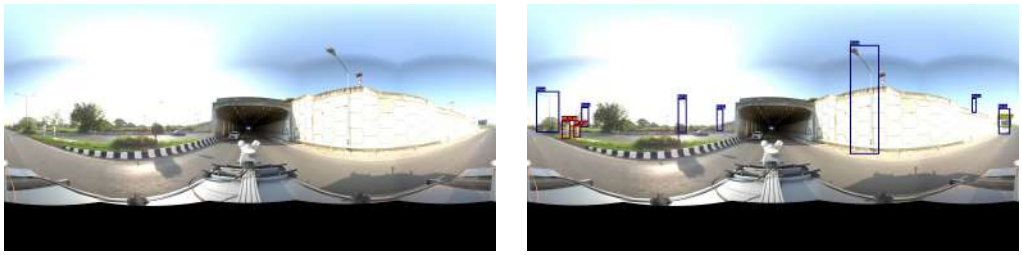
DeiT [32] introduces several training policies that allow it also to be efficient using the extra modest ImageNet-1K corpus. The effects of ViT on computer vision tasks are encouraging. Still, its model is inappropriate for profit as a general-purpose backbone network on dense image tasks due to its low-resolution kernel filter and the quadratic improvement in complexity with the image size. Some works utilize ViT models for the dense image tasks of image labeling and detection through transpose or upsampling layers yet comparatively lower precision. Unsurprisingly, we find ViT [33,34] models perform the best performance-accuracy trade-off among these methods on computer vision tasks, even though this work concentrates on general-purpose performance relatively than particularly on segmentation (labeling). It investigates a comparable line of studying to produce multi-resolution kernel features on ViT. Moreover, its complexity is still quadratic to the size of an image. At the same time, ours is linear and operates regionally, which has shown advantages in modeling the significant correlation in visual signals. Furthermore, it is efficient and effective, achieving SOTA performance, e.g., $MeanIoU$, $AveragePrecision(AP)$ on COCO object detection, and ADE20K image labeling.

## 3. Road Asset Data Set

Department of Highway (Thailand) has a highway road network of more than 52,000 km. To acquire information about road assets and roadside categories within the highway road network would take a lot of resource equipment and human resources. To solve data collection problems. A Mobile Mapping System (MMS.) and Artificial Intelligence (AI) were implemented. Its results from the current information, complete details, and highly efficient applications.

We have the process of collecting geospatial data from mobile vehicles (cars). Vehicles could be equipped with a range of sensors such as positioning (GNSS, GPS) and cameras.

The panoramic image type of The Ladybug-5 is a 360-degree spherical camera producing an image with a resolution of 8000 × 4000 pixels. It is required to calculate the position of objects on the pictures. Figure 1 depicts an example of the data set used in this study. A 360-degree spherical camera that can capture 8k30 or 4k60 footage is used to obtain the data set. The Ladybug5+ produces high-quality photographs with a 2 mm accuracy level at 10 m because of its proprietary calibration and better global shutter sensors. The Ladybug SDK offers a wide range of features that make it simple to capture, analyze, and spherical export material (shown as Figure 2). Furthermore, Figure 3 shows the sample size required for the detection task (number of images per class).



(**a**) (**b**)

**Figure 1.** The challenges in the Road Asset corpus. Sample of input image (**a**) and target image (**b**).

To survey and collect information on various types of highway assets. For use in the management of highway works in three main areas: (i) road asset management and maintenance, (ii) in terms of road safety to analyze the location, and (iii) in the planning of highway development projects. Therefore, a complete survey of the number of kilometer digits and the position is correct; accordingly, it is necessary to solve the duplication of construction work. The problem of calculating the amount is their construction work.

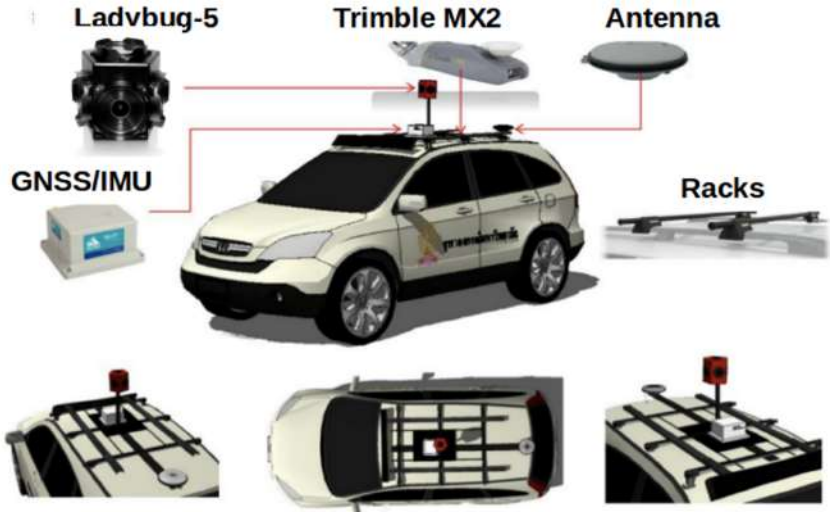
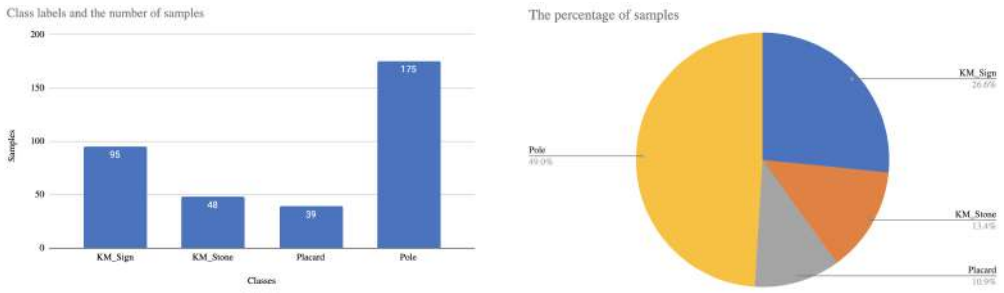



**Figure 2.** Our Ladybug5+ 360 degrees spherical camera.

As our deep learning environment setup, We use "TensorFlow Core v2.7.0 (TF)" [35] to create an end-to-end open-source platform for deep learning. The entire experiments were performed on servers with Intel® Xeon® Scalable 4210R (10 core, 2.4 GHz, 13.75 MB, 100 W), 256 GB of memory, and the NVIDIA RTX™ 1080Ti (11 GB) × 2 cards.




(**a**) (**b**)

**Figure 3.** The sample size of annotated class and its corresponding object class in the Road Asset corpus. Sample size for each class (**a**) and Percent of sample size for each class (**b**).

## 4. Proposed Method

### *4.1. Transformer Based YOLOX*

Although currently, YOLOv5 [13] is already performing well, some recent work on object detection has triggered the development of this new YOLOX algorithm [14]. The most important focus points in object detection are anchor-free detectors, advanced label assignment strategies, and end-to-end detectors. These new focal points are still not integrated into the YOLO algorithm, and YOLOv4 and YOLOv5 are still anchor-based detectors and use hand-crafted assigning rules for training. It is a fundamental reason for the development of the YOLOX algorithm.

Our Transformer-based YOLOX follows a sequence-to-sequence vector with transformers from [36] and the corresponding output vector with input vector fabrication as in Natural Language Processing (NLP), the capacity of a machine application to understand mortal language. The most famous image classification network simply employs the Transformer Encoder to convert the multiple input tokens. However, the decoder component of the conventional transformer network is also employed for other purposes.

The regular ViT-YOLO model [15] and its conversion for computer vision tasks where the relations between a token (image patches) and each other tokens are calculated. The global figure leads to quadratic complexity concerning the number of image patches,

addressing it unsuitable for many image problems requiring an immense set of tokens for the softmax layer.

A pure transformer-based encoder learns feature representations furnished the $1D$ vector of embedding sequence $E$ input. It means each ViT layer has a global receptive field, answering the insufficient receptive field problem of the existing encoder-decoder deep neural network already and for all. The ViT encoder consists of $L_e$ layers of multilayer perceptron (MLP) and multi-head self-attention (MSA) modules.

This distinct behavior appears to be due to the inclusion of some inductive biases in CNNs, which these networks can use to comprehend the particularities of the analyzed image more rapidly, even if they end up restricting them and making it more difficult to grasp global relationships. Input visual distortions such as adversarial patches or permutations were also significantly more resistant to Vision Transformers. In reality, CNNs generate outstanding outcomes even when trained on data sets that are not as huge as Vision Transformers trade.

In case that a conventional encoder designed for image labeling would downsample a $2D$ image $x \in R^{HW3}$ into a grid of into a featuremap $x_f \in R^{\frac{H}{16} \times \frac{W}{16} \times C}$ we thus decide to set the transformer input sequence length $L$ as $\frac{H}{16} \times \frac{W}{16} = \frac{W}{256}$. This means the output of the vector sequence of the ViT can be clearly reshaped to the point kernel map $x_f$.

To recover the $\frac{HW}{256}$-long vector sequence of our input, we divide an image $x \in R^{H \times W \times 3}$ into a grid of as $\frac{H}{16} \times \frac{W}{16}$ patches uniformly, several ViT modules with modified self-attention calculation (SwinTF modules) are adapted on these image patch tokens. The ViT module maintain the number of patches $\frac{H}{4} \times \frac{W}{4}$ and then make a series out of this grid. Each vectorized patch p is mapped into a latent C-dimensional embedding space using a linear projection function. $f : p \to e \in R^C$, for a patch $x$, we obtain a 1D series of vector embeddings. We get a unique embedding $p_i$ for each position $i$ to encode the patch spatial information, which is then added to $e_i$ to generate the final sequence input $E = \{e_1 + p_1, e_2 + p_2, ..., e_L + p_L\}$. In this process, spatial data is kept notwithstanding the order-less attention type of transformers.

A classical transformer-based encoder accepts feature representations when given the 1D embedding sequence $E$ as input. It means that each ViT layer has a global receptive field, resolving the problem of the standard deep learning encoder's restricted sensory area once and for all. The encoder of SwinTF consists of $L_e$ vector of MLP and MSA modules (Figure 4). At each layer $l$, the input to self-attention is in a triplet of $(query, key, value)$ calculated from the input $Z^{l-1} \in R^{L \times C}$ as:

$$query = Z^{l-1}W_Q, key = Z^{l-1}W_K, value = Z^{l-1}W_V \tag{1}$$

where $W_Q/W_K/W_V \in R^{C \times d}$ are the learnable weights of three linear projection vectors and $d$ is the dimension of $(query, key, value)$. Self-attention (SA) is then expressed as:

$$SA(Z^{l-1}) = Z^{l-1} + softmax(\frac{Z^{l-1}W_Q(ZW_K)^T}{\sqrt{d}})(Z^{l-1}W_V) \tag{2}$$

MSA is a reckoning with m self-supporting SA actions and projects their concatenated outputs: $MSA(Z^{l-1}) = [SA_1(Z_l - 1); SA_2(Z_l - 1); ...; SA_m(Z_l - 1)]W_O$, where $W_O \in R^{md} \times C$. $d$ is typically set to $C/m$. The output of MSA is then transformed by an MLP module with residual skip as the output layer as:

$$Z^l = MSA(Z_{l-1}) + MLP(MSA(Z^{l-1})) \in R^{L \times C}. \tag{3}$$

Lastly, a normalized layer is employed before MLP and MSA modules which are omitted for clearness. We express $Z^1, Z^2, Z^3, ..., Z^{L_e}$ as the weights of transformer vectors.

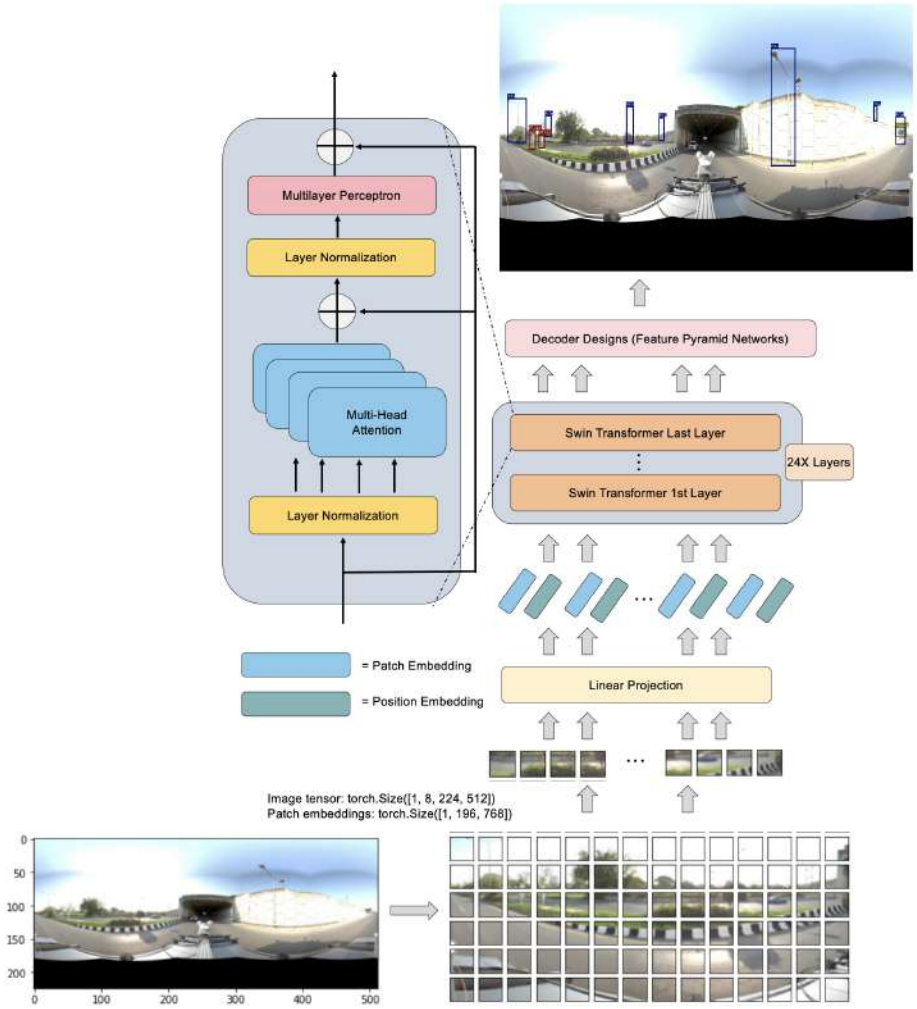



**Figure 4.** Overall architecture of our Transformer-Based YOLOX.

### *4.2. Feature Pyramid Network (FPN) Decoder Design*

Objects from the road captured images vary a lot in sizes while the feature map from a single layer of the convolutional neural network has limited capacity of representation, so its crucial to effectively represent and process multi-scale features. FPN decoder designs as portrayed in Figure 5 are set up to achieve pixel-level labeling.

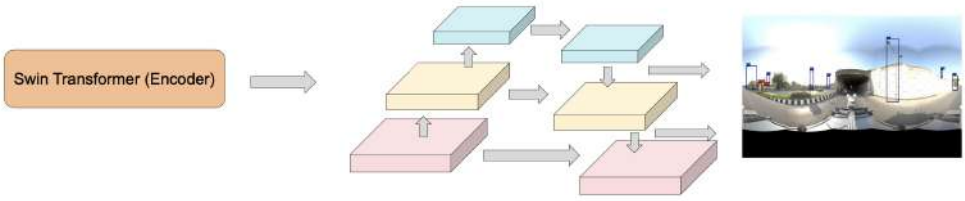



**Figure 5.** Our feature pyramid decoder design.

FPN [24] is a characteristic extractor created with accuracy and speed in mind for such a pyramid idea. It takes the place of detectors like Faster R-feature CNN's extractor [37]. Image recognition generates many feature map layers (multi-scale feature maps) with superior quality information than the traditional feature pyramid. It also utilizes specifically constructed transformers in a self-level, top-down, and bottom-up interactive pattern to change any feature pyramid into another feature pyramid of the same size but with richer contexts. It features a simple query, key, and value operation (Equation (1)) that is demonstrated to be important in choosing informative long-range interaction, which fits

our objective of non-local interaction at appropriate sizes. We depict the higher-level "idea" using the visual qualities of the lower-level "pixels" intuitively. Each level's altered feature maps (red, yellow, and blue) are resized to their matching map size and concatenated with the original map before being sent into the convolution layer, which resizes them to the accurate "thickness." Higher-resolution features are upsampled from higher-pyramid-level feature maps, which are spatially coarser but semantically more robust. The spatial resolution is upsampled by a factor of two, with the nearest neighbor being used for simplicity. Each lateral link combines feature maps from the bottom-up and top-down paths of the same spatial size. To minimize the channel dimensions, the feature maps from the bottom-up course are convolutional 11 times. In addition, element-wise addition is used to combine the feature maps from the bottom-up and top-down pathways. Finally, a 33 convolution is applied to each merged map to form the final feature map to reduce the aliasing impact of upsampling. This last collection of feature maps corresponds to the precise spatial dimensions. Because all layers of the pyramid, like a standard featured picture pyramid, employ joint classifiers/regressors, the feature dimension at output d is fixed at $d = 256$. As a result, the outputs of all further convolutional layers are 256-channel.

## 5. Experimental Results

Our proposed YOLOX with Vision Transformer and FPN method reaches the highest performance on Average Precision (AP) rating at 61.15% in the testing set. At the same time, the YOLOv5L is the best baseline, including with its fixed backbone as modification of CSP-v5, rated lower in terms of average AP at 58.94%, which this baseline is less average AP than the proposed method at 2.21% as shown in Table 1. In the AP 50 precision, the YOLOX with Transformer with FPN outperforms the highest precision rating at 69.34%. In comparison, the YOLOv5L model is significantly less precise in terms of AP75 than the proposed method at 10.19%, as shown in Table 1. Moving to the most significant AP at 75, the YOLOX with Transformer and FPN reaches the highest measurement at 55.23%, while the best combination of YOLOv5L reaches the AP 75 at 53.66%, which this baseline is less than the proposed method by 1.57%. There is a trade-off between performance as precision (AP) and the complexity of the deep convolution neural networks architecture. The YOLOX method achieves more the Average AP than the YOLOv5L, around 2%. Still, the proposed method takes a long time to train when comparing YOLOv5L because the trainable parameters of YOLOX are significantly higher than YOLOv5L, about 20M, as shown in Table 1. Table 2 displays numbers of training, validation, and testing sets on our road asset data set.

**Table 1.** Comparison of the average precision of different object classifiers on Road Asset test-corpus. We select all the models trained on 100 epochs for a fair comparison.

| | **Backbone** | **Model** | ***AP*(%)** | ***Param*** | ***F*1** | ***FPS*** |
|---|---|---|---|---|---|---|
| **Baseline** | Modified CSP v5 | YOLOv5-S | 49.2 | 7.8M | 63.01 | 22 |
| | Modified CSP v5 | YOLOv5-M | 52.33 | 21.8M | 66.22 | 23 |
| | Modified CSP v5 | YOLOv5-L | 58.94 | 47.8M | 67.88 | 26 |
| | ResNext | Faster R-CNN | 56.32 | 43.12M | 65.12 | 15 |
| | ResNext | CentreNet | 58.11 | 12.2M | 62.33 | 31 |
| **Proposed** | ViT + FPN | YOLOX | 61.15 | 87.3M | 71.11 | 11 |

The YOLOX, with the combination of the modern image classification front-end, namely Vision Transformer with FPN, achieves the highest AP in the large-scale object classes such as *KMSign* and *Placard*. The release of YOLOv5 includes five different models sizes: YOLOv5s (smallest), YOLOv5m, YOLOv5l, YOLOv5x (largest).

Furthermore, it reaches the highest average precision (AP) at 51.32% and 57.63%, as shown in Table 3, in those mentioned classes. Our proposed outperforms when comparing the performance of YOLO-v5-L on these large-scale object classes in terms of AP. The YOLOX contains higher AP than YOLO-v5-L in categories such as *KMSign* and *Placard* by

6.47% and 2.82%, as shown in Table 3, respectively. Turning to the smaller object classes such as *KMStone* and *Pole*, our proposed method is still the winner of AP's *KMStone* and *Pole* rating at 61.22% 60.88% as shown in Table 3, respectively. Compared to the YOLO-v5-L method, the proposed method outperforms AP on the smaller object classes, both *KMStone* and *Pole* by 3.79% and 2.45%, as shown in Table 3, respectively.

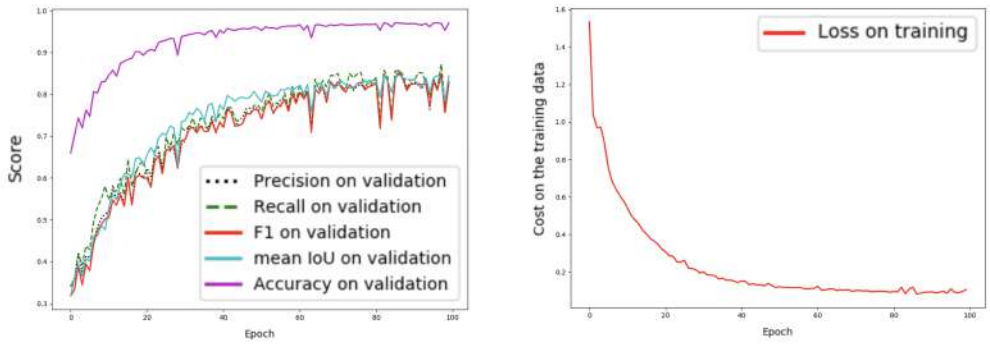
In the learning curve analysis, the cost function of YOLOX coupling with the Vision transformer with FPN, representing the line graph on the right-hand side, constantly learns into the local optima for all 100 epochs of the training set shown in Figure 6. In addition, the line graphs represent the performances of our proposed method in terms of Precision, Recall, F1, mean IoU, and Accuracy, which are evaluated on the validation set. On the left-hand side, the line graphs exhibit the upper tendency. Precision, Recall, F1, and mean IoU line graphs reach their highest performances, about 70% at the epoch of 100 on the validation set, while the line graph of Accuracy goes almost 100% at the end of 100 training epoch. Furthermore, these performances of line graphs drop at the epoch of 80, as shown in Figure 6.

**Table 2.** Numbers of training, validation, and testing sets.

| Data Set | Total Images | Training Set | Validation Set | Testing Set |
|---|---|---|---|---|
| Road Asset Corpus | 1300 | 800 | 300 | 200 |

**Table 3.** Comparison of the average precision between the results of four categories after subsequent operations on Road Asset test-dev corpus.

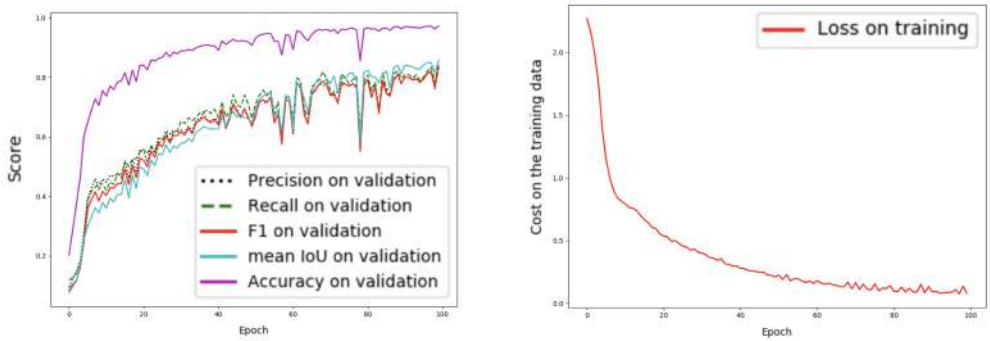
| | Backbone | Model | *KMSign* | *KMStone* | *Pole* | *Placard* |
|---|---|---|---|---|---|---|
| **Baseline** | Modified CSP v5 | YOLOv5-S | 34.57 | 44.23 | 51.12 | 46.13 |
| | Modified CSP v5 | YOLOv5-M | 40.12 | 53.12 | 54.73 | 47.23 |
| | Modified CSP v5 | YOLOv5-L | 44.85 | 57.43 | 58.43 | 54.81 |
| | ResNext | Faster R-CNN | 46.75 | 54.22 | 57.34 | 56.22 |
| | ResNext | CentreNet | 37.625 | 54.35 | 56.66 | 53.21 |
| **Proposed** | ViT + FPN | YOLOX | 51.32 | 61.22 | 60.88 | 57.63 |




(**a**) (**b**)

**Figure 6.** Graph (learning curves) of the Road Asset corpus of the proposed approach, "YOLOX-based Vision Transformer with FPN"; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.

On the other hand, learning curves of both accuracy and loss graph of our best baseline (YOLOv5L) have shown via Figure 7. It shows their charts are smooth less than our proposed method.




(**a**) (**b**)

**Figure 7.** Graph (learning curves) of the Road Asset corpus of the baseline approach, "YOLOv5L"; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.

As shown in Figure 8, it provides qualitative object detection results of our YOLOX-Transformer with FPN model on an arbitrary image from Road Asset corpus. Lastly, we trained (finetuning) our YOLOX-Transformer with FPN model again via Pascal VOC data set [38] and prediction results shown in Figure 9.

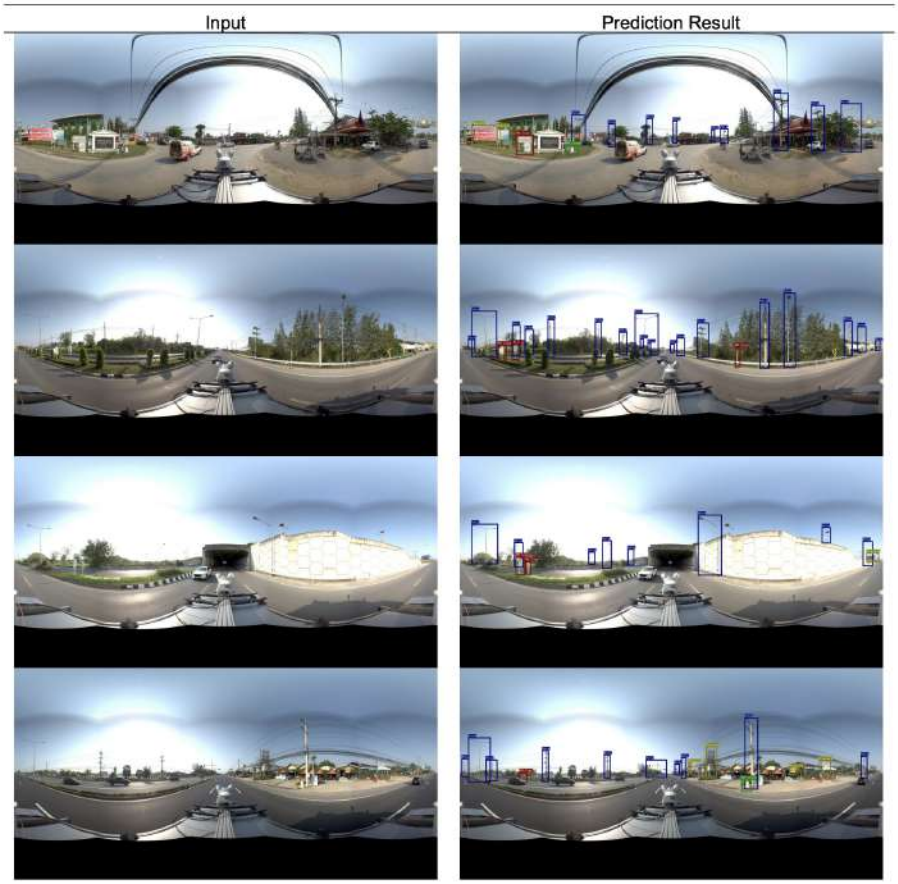



**Figure 8.** Visualized detection results of YOLOX-Transformer with FPN on an arbitrary image from Road Asset corpus.

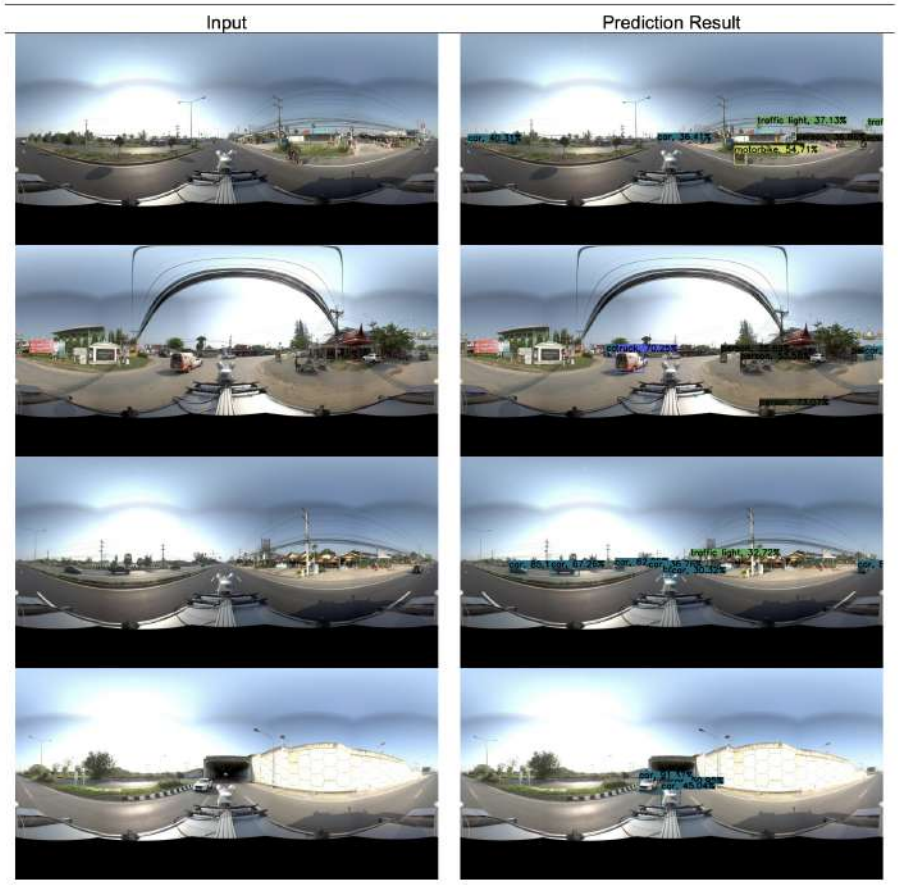



**Figure 9.** Visualized detection results of YOLOX-Transformer with FPN on Pascal VOC data set.

## 6. Conclusions

This paper proposes a novel Transformer-Based YOLOX with FPN, high-performance anchor-free YOLO for object detection. Our model can globally focus on dependencies between image feature patches and retain sufficient spatial information for object detection via multi-head self-attention. Furthermore, other effective techniques are adopted to achieve better accuracy and robustness. Furthermore, we apply FPN as learnable weights of decoder design to learn the importance of different input features instead of simply summing up or concatenating, which may cause feature mismatch and performance degradation. In particular, our Transformer-Based YOLOX with FPN achieves a new record of 61.15% box AP, 69.34% box AP50, and 55.23% box AP75 on Thailand highway test-dev, outperforming the prior SOTA model, including the following: YOLOv5S, YOLOv5M, and YOLOv5L.

**Author Contributions:** Conceptualization, T.P.; Formal analysis, T.P.; Investigation, T.P.; Methodology, T.P.; Project administration, T.P.; Resources, T.P.; Software, T.P.; Supervision, T.P.; Validation, T.P.; Visualization, T.P.; Writing–original draft, T.P.; Writing–review and editing, T.P.; Stand up and cheer, C.C., P.S., S.T., W.W., K.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Center of Excellence in Infrastructure Management, Chulalongkorn University and the Ratchadapisek Somphot Fund for Postdoctoral Fellowship, Chulalongkorn University.

**Acknowledgments:** Teerapong Panboonyuen, also known as Kao Panboonyuen appreciates (thanks) and acknowledges the scholarship from Ratchadapisek Somphot Fund for Postdoctoral Fellowship, Chulalongkorn University, Thailand.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| FPN | Feature Pyramid Network |
| Param | Parameters |
| SwinTF | Swin Transformer |
| ViT | Vision Transformer |

## References


1. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7373–7382.
2. Liu, S.; Zhang, L.; Lu, H.; He, Y. Center-Boundary Dual Attention for Oriented Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5603914. [CrossRef]
3. Gu, Y.; Wang, L.; Wang, Z.; Liu, Y.; Cheng, M.M.; Lu, S.P. Pyramid constrained self-attention network for fast video salient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10869–10876.
4. Haris, M.; Glowacz, A. Road Object Detection: A Comparative Study of Deep Learning-Based Algorithms. *Electronics* **2021**, *10*, 1932. [CrossRef]
5. Chen, G.; Chen, K.; Zhang, L.; Zhang, L.; Knoll, A. VCANet: Vanishing-Point-Guided Context-Aware Network for Small Road Object Detection. *Automot. Innov.* **2021**, *4*, 400–412. [CrossRef]
6. Wang, K.; Liu, M.; Ye, Z. An advanced YOLOv3 method for small-scale road object detection. *Appl. Soft Comput.* **2021**, *112*, 107846. [CrossRef]
7. Li, G.; Xie, H.; Yan, W.; Chang, Y.; Qu, X. Detection of road objects with small appearance in images for autonomous driving in various traffic situations using a deep learning based approach. *IEEE Access* **2020**, *8*, 211164–211172. [CrossRef]
8. Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; Gao, J. Focal self-attention for local-global interactions in vision transformers. *arXiv* **2021**, arXiv:2107.00641.
9. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 687–694.
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
11. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
12. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [CrossRef]
13. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef] [PubMed]
14. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
15. Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; Liu, F. ViT-YOLO: Transformer-Based YOLO for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 2799–2808.
16. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 5998–6008.
18. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
19. Qing, Y.; Liu, W.; Feng, L.; Gao, W. Improved Transformer Net for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 2216. [CrossRef]
20. Sun, Z.; Cao, S.; Yang, Y.; Kitani, K.M. Rethinking transformer-based set prediction for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 3611–3620.
21. Yang, F.; Zhai, Q.; Li, X.; Huang, R.; Luo, A.; Cheng, H.; Fan, D.P. Uncertainty-Guided Transformer Reasoning for Camouflaged Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 4146–4155.
22. Wang, L.; Li, R.; Wang, D.; Duan, C.; Wang, T.; Meng, X. Transformer Meets Convolution: A Bilateral Awareness Network for Semantic Segmentation of Very Fine Resolution Urban Scene Images. *Remote Sens.* **2021**, *13*, 3065. [CrossRef]
23. Jin, Y.; Han, D.; Ko, H. TrSeg: Transformer for semantic segmentation. *Pattern Recognit. Lett.* **2021**, *148*, 29–35. [CrossRef]
24. Kim, S.W.; Kook, H.K.; Sun, J.Y.; Kang, M.C.; Ko, S.J. Parallel feature pyramid network for object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 234–250.

25. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 2778–2788.
26. Thuan, D. Evolution of Yolo Algorithm and Yolov5: The State-of-the-Art Object Detection Algorithm. 2021. Available online: https://www.theseus.fi/handle/10024/452552 (accessed on 12 November 2021).
27. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-trained image processing transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12299–12310.
28. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 12179–12188.
29. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 16519–16529.
30. Kim, K.; Wu, B.; Dai, X.; Zhang, P.; Yan, Z.; Vajda, P.; Kim, S.J. Rethinking the Self-Attention in Vision Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3071–3075.
31. Salvador, A.; Gundogdu, E.; Bazzani, L.; Donoser, M. Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 15475–15484.
32. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*; PMLR: London, UK, 2021; pp. 10347–10357.
33. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
34. Lin, A.; Chen, B.; Xu, J.; Zhang, Z.; Lu, G. DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation. *arXiv* **2021**, arXiv:2106.06716.
35. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the OSDI, Savannah GA USA, 2–4 November 2016; Volume 16, pp. 265–283.
36. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 6881–6890.
37. Cheng, B.; Wei, Y.; Shi, H.; Feris, R.; Xiong, J.; Huang, T. Revisiting rcnn: On awakening the classification power of faster rcnn. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 453–468.
38. Vicente, S.; Carreira, J.; Agapito, L.; Batista, J. Reconstructing pascal voc. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 41–48.

*remote sensing*

MDPI

*Article*

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

**Teerapong Panboonyuen [1], Kulsawasd Jitkajornwanich [2], Siam Lawawirojwong [3], Panu Srestasathiern [3] and Peerapon Vateekul [1,*]**

1 Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand; teerapong.panboonyuen@gmail.com

2 Data Science and Computational Intelligence (DSCI) Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd, Ladkrabang, Bangkok 10520, Thailand; kulsawasd.ji@kmitl.ac.th

3 Geo-Informatics and Space Technology Development Agency (Public Organization), 120, The Government Complex, Chaeng Wattana Rd, Lak Si, Bangkok 10210, Thailand; siam@gistda.or.th (S.L.); panu@gistda.or.th (P.S.)

* Correspondence: peerapon.v@chula.ac.th

Received: 5 March 2020; Accepted: 9 April 2020; Published: 12 April 2020


**Abstract:** One of the fundamental tasks in remote sensing is the semantic segmentation on the aerial and satellite images. It plays a vital role in applications, such as agriculture planning, map updates, route optimization, and navigation. The state-of-the-art model is the Enhanced Global Convolutional Network (GCN152-TL-A) from our previous work. It composes two main components: (*i*) the backbone network to extract features and (*ii*) the segmentation network to annotate labels. However, the accuracy can be further improved, since the deep learning network is not designed for recovering low-level features (e.g., river, low vegetation). In this paper, we aim to improve the semantic segmentation network in three aspects, designed explicitly for the remotely sensed domain. First, we propose to employ a modern backbone network called "High-Resolution Representation (HR)" to extract features with higher quality. It repeatedly fuses the representations generated by the high-to-low subnetworks with the restoration of the low-resolution representations to the same depth and level. Second, "Feature Fusion (FF)" is added to our network to capture low-level features (e.g., lines, dots, or gradient orientation). It fuses between the features from the backbone and the segmentation models, which helps to prevent the loss of these low-level features. Finally, "Depthwise Atrous Convolution (DA)" is introduced to refine the extracted features by using four multi-resolution layers in collaboration with a dilated convolution strategy. The experiment was conducted on three data sets: two private corpora from Landsat-8 satellite and one public benchmark from the "ISPRS Vaihingen" challenge. There are two baseline models: the Deep Encoder-Decoder Network (DCED) and our previous model. The results show that the proposed model significantly outperforms all baselines. It is the winner in all data sets and exceeds more than 90% of *F*1: 0.9114, 0.9362, and 0.9111 in two Landsat-8 and ISPRS Vaihingen data sets, respectively. Furthermore, it achieves an accuracy beyond 90% on almost all classes.

**Keywords:** deep learning; convolutional neural network; global convolution network; feature fusion; depthwise atrous convolution; high-resolution representations; ISPRS vaihingen; Landsat-8

## 1. Introduction

Semantic segmentation in a medium resolution (MR) image, e.g., a Landsat-8 (LS-8) image, and very high resolution (VHR) images, e.g., aerial images, is a long-standing issue and problem in the domains of remote sensing-based information. Natural objects such as roads, water, forests, urban, and agriculture fields regions are operated in various tasks such as route optimization to create imperative remotely sensed applications.

Deep learning, especially the Deep Convolutional Neural Network (CNN), is an acclaimed approach for automatic feature learning. In previous research, CNN-based segmentation approaches are proposed to perform semantic labeling [1–5]. To achieve such a challenging task, features from various levels are fused together [5–7]. Specifically, a lot of approaches fuse low-level and high-level features together [5–9]. In remote sensing corpora, ambiguous human-made objects need high-level features for a more well-defined recognition (e.g., roads, building roofs, and bicycle runways), while fine-structured objects (e.g., low vegetations, cars, and trees) could benefit from comprehensive low-level features [10]. Consequently, the performance will be affected by the different numbers of layers and/or different fusion techniques of the deep learning model.

In recent years, the Global Convolutional Network (GCN) [11], the modern CNN, has been introduced, in which the valid receptive field and large filter enable dense connections between pixel-based classifiers and activation maps, which enhances the capability to cope with different transformations. The GCN is aimed at addressing both the localization and segmentation problems for image labeling and presents Boundary Refinement (BR) to refine the object boundaries further as well. Our previous work [12] extended the GCN by enhancing three approaches as illustrated in Figures 1 and 2. First, "Transfer Learning" [13–15] was employed to relieve the shortage problem. Next, we varied the backbone network using ResNet152, ResNet101, and ResNet50. Last, "Channel Attention Block" [16,17] was applied to allocate CNN parameters for the output of each layer in the front-end of the deep learning architecture.

Nevertheless, our previous work still disregards the local context, such as low-level features in each stage. Moreover, most feature fusion methods are just a summation of the features from adjacent stages and they do not consider the representations of diversity (critical for the performance of the CNN). This leads to unpredictable results that suffer from measuring the performance such as the $F1$ score. This, in fact, is the inspiration for this work.

In summary, although the current enhanced Global Convolutional Network (GCN152-TL-A) method [12] has achieved significant breakthroughs in semantic segmentation on remote sensing corpora, it is still laborious to manually label the MR images in river and pineapple areas and the VHR images in low vegetation and car areas. The two reasons are as follows: $(i)$ previous approaches are less efficient to recover low-level features for accurate labeling, and $(ii)$ they ignore the low-level features learned by the backbone network's shallow layers with long-span connections, which is caused by semantic gaps in different-level contexts and features.

In this paper, motivated by the above observation, we propose a novel Global Convolutional Network ("HR-GCN-FF-DA") for segmenting multi-objects from satellite and aerial images, as illustrated in Figure 3. This paper aims to further improve the state-of-the-art on semantic segmentation in MR and VHR images. In this paper, there are three contributions, as follows:

- Applying a new backbone called "High-Resolution Representation (HR)" to GCN for the restoration of the low-resolution representations of the same depth and similar level.
- Proposing the "Feature Fusion (FF)" block into our network to fuse each level feature from the backbone model and the global model of GCN to enrich local and global features.
- Proposing "Depthwise Atrous Convolution (DA)" to bridge the semantic gap and implement durable multi-level feature aggregation to extract complementary information from very shallow features.

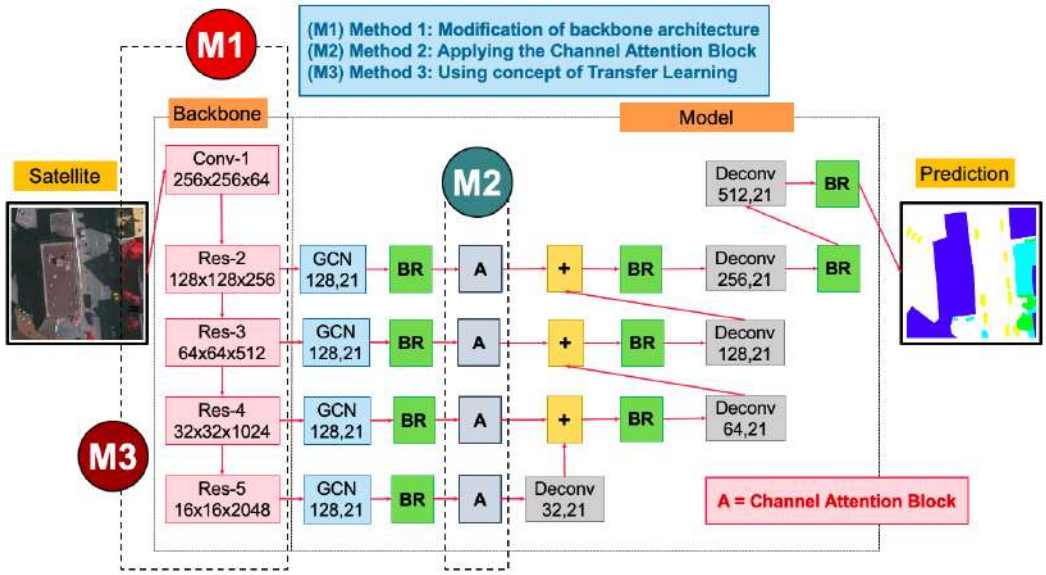



**Figure 1.** An overview of enhanced GCN (Global Convolution Network [11]) with transfer learning and attention mechanism (GCN152-TL-A) [12].

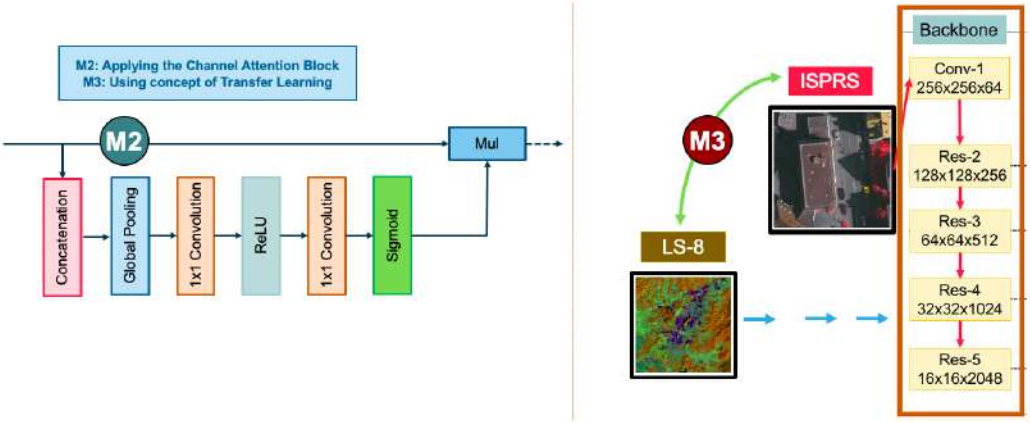



**Figure 2.** An Attention Mechanism (A) block (left) and the Transfer Learning (TL) approach (right) transfer knowledge (from pre-trained weights) across two corpora—medium and very high resolution images from [12].

The experiments were conducted using the widespread aerial imagery, ISPRS (Stuttgart) Vaihingen [18] data set and GISTDA (Geo-Informatics and Space Technology Development Agency (Public Organization)), organized by the government in our country, data sets (captured by the Landsat-8 satellite). The results revealed that our proposed method surpasses the two baselines: Deep Convolutional Encoder-Decoder Network (DCED) [19–21] and the enhanced Global Convolutional Network (GCN152-TL-A) method [12] in terms of $F1$ score.

The remainder of this paper is organized as follows: Section 2 discusses related work. Our proposed methods are detailed in Section 3. Next, Section 4 provides the details on remote sensing corpora. Section 5 presents our performance evaluation. Then, Section 6 reports the experimental results, and Section 7 is the discussion. Last, we close with the conclusions in Section 8.

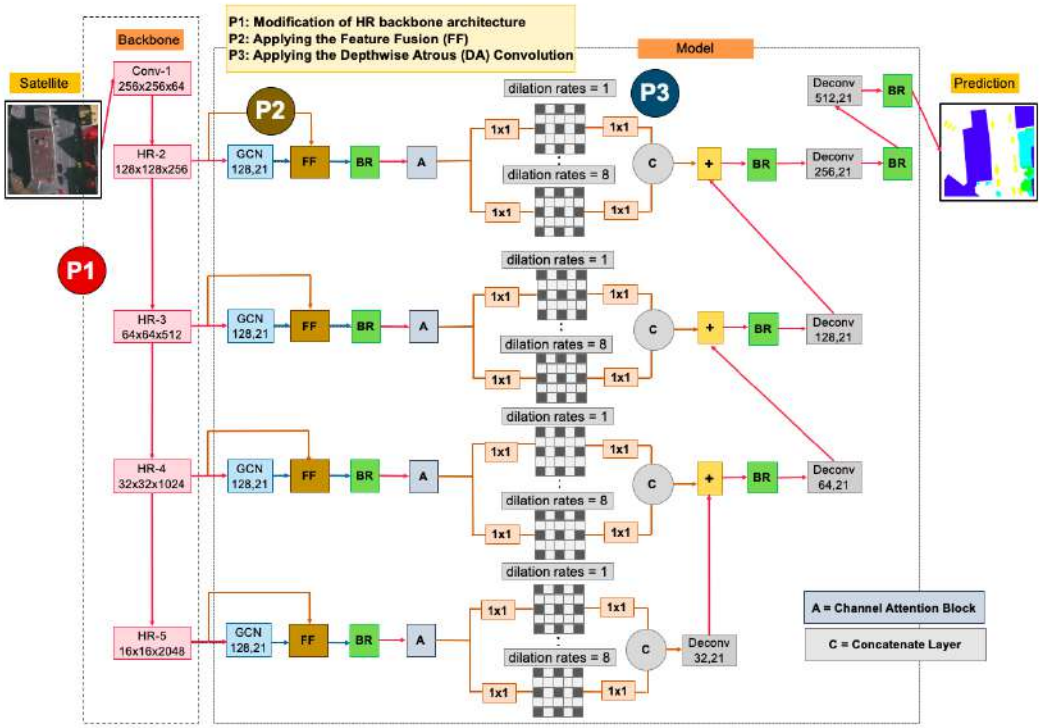



**Figure 3.** The HR-GCN-FF-DA: an enhanced GCN architecture with feature fusion and depthwise atrous convolution.

## 2. Related Work

The CNN has been outstandingly utilized for the data analysis of remote sensing domains, in particular, land cover classification or segmentation of agriculture or forest districts [10,12,22–26]. It has rapidly become a successful method for accelerating the process of computer vision tasks, e.g., image classification, object detection, or semantic segmentation with high precision results [4,27–33] and is a fast-growing area.

It is separated into two subsections: $(i)$ we demonstrate modern CNN architectures for semantic labeling on both traditional computer vision and remote sensing tasks and $(ii)$ the novel techniques of deep learning, especially playing with images, are discussed.

### *2.1. Modern CNN Architecture for Semantic Labeling*

In early research, several DCED-based approaches have obtained a high performance in the various baseline corpora [16,19–21,26,34–36]. Nevertheless, most of them also struggle with issues with performance accuracy. Consequently, much research on novel CNN architectures has been introduced, such as a high-resolution representation [37,38] network that supports high-resolution representations in all processes by connecting high-to-low and low-to-high-resolution convolutions to keep high and low-resolution representations. CSRNet [8] proposed an atrous (dilated) CNN to comprehend highly congested scenes through crowd counting and generating high-quality density maps. They deployed the first ten layers from VGG-16 as the backbone convolutional models and dilated convolution layers as the backend to enlarge receptive fields and extract deeper features without losing resolutions. SeENet [6] enhanced shallow features to alleviate the semantic gap between deep features and shallow features and presented feature attention, which involves discovering complementary information from low-level features to enhance high-level features for precise segmentation. It also was constructed with the parallel pyramid to implement precise semantic segmentation. ExFuse [7] proposed to boost the feature fusion by bridging the semantic and resolution gap between low-level and high-level feature maps. They proposed more semantic information into low-level features with three aspects:

$(i)$ semantic supervision, $(ii)$ semantic embedding branch, and $(iii)$ layer rearrangement. They also embeded spatial information into high-level features. In the remote sensing corpus, ResUNet [25] proposed a trustworthy structure for performance effects for the job of image labeling of aerial images. They used a VGG16 network as a backbone, combined with the pyramid scene parsing pooling and dilated deep neural network. They also proposed a new generalized dice loss for semantic segmentation. TreeUNet (also known as adaptive tree convolutional neural networks) [24] proposed a tree-cutting algorithm and an adequate deep neural network with inadequate binary links to increase the classification percentage at the pixel level for subdecimeter aerial imagery segmentation, by sending kernel maps within concatenating connections and fusing multi-scale features. From the ISPRS Vaihingen Challenge and Landsat-8 corpus, the enhanced Global Convolutional Network (also known as "GCN152-TL-A"), illustrated in Figure 1, Panboonyuen et al. (2019) [12] presented an enhanced GCN for semantic labeling with three main contributions. First, "Domain-Specific Transfer Learning" (TL) [13–15], illustrated in Figure 2 (right), aims to restate the weights obtained from distinct fields' inputs. It is currently prevalent in various tasks, such as Natural Language Processing (NLP), and has also become popular in Computer Vision (CV) in the past few years. It allows you to reach a deep learning model with comparatively inadequate data. They prefaced to relieve the lack of issue on the training set by appropriating other remote sensing data sets with various satellites with an essentially pre-trained weight. Next, "Channel Attention", shown in Figure 2 (left), proposed with their network to select the most discriminative kernels (feature maps). Finally, they enhanced the GCN network by improving its backbone by using "ResNet152". "GCN152-TL-A" has surpassed state-of-the-art (SOTA) approaches and become the new SOTA. Hence, "GCN152-TL-A" is selected as our baseline in this work.

### *2.2. Modern Technique of Deep Learning*

A novel technique of deep learning is an essential agent for improving the precision of deep learning, especially the CNN. While the most prevalent contemporary designs tick all the boxes for image labeling responsibilities, e.g., the atrous convolution (also known as dilated convolution), channel attention mechanism, refinement residual block, and feature fusion, and have been utilized to boost the performance of the deep learning model.

Atrous convolution [5,6,9,39,40], also known as multi-scale context aggregation, is proposed to regularly aggregate multi-scale contextual information devoid of losing resolution. In this paper, we use the technique of "Depthwise Atrous Convolution (DA)" [6] to extract complementary information from very shallow features and enhance the deep features for improving feature fusion from our feature fusion step.

The channel attention mechanism [16,17] generates a one-dimensional tensor for allowed feature maps, which is activated by the softmax function. It focuses on global features found in some feature maps and has attracted broad interest in extracting rich features in the computer vision domain and offers great potential in improving the performance of the CNN. In previous work, GCN152-TL-A [12], the self attention and utilize channel attention modules are applied to pick the features similar to [16].

Refinement residual block [16] is part of the enhanced CNN model with ResNet-backbones, e.g., ResNet101 or ResNet152. This block is used after the GCN module and during the deconvolution layer. It is used to refine the object boundaries further. In our previous work, GCN152-TL-A [12], we employed the boundary refinement block (BR) that is based on the"Refinement Residual Block" from [11].

Feature fusion [7,41–44] is regularly manipulated in semantic labeling for different purposes and concepts. It presents a concept that combines multiplied, added, or concatenate CNN layers for improving a process of dimensionality reduction to recover and/or prevent the loss of some important features such as low-level features (e.g., lines, dots, or gradient orientation with the content of an image scene). In another way, it can also recover high-level features by using the technique of "high-to-low and low-to-high" [37,38] to produce high-resolution representations.

## 3. Proposed Method

Our proposed deep learning architecture, "HR-GCN-FF-DA", is demonstrated in an overview architecture in Figure 3. The network, based on GCN152-TL-A [12], consists primarily of three parts: (*i*) changing the backbone architecture (the P1 block in Figure 3), (*ii*) implementing the "Feature Fusion" (the P2 block in Figure 3), and (*iii*) using the concept of "Depthwise Atrous Convolution" (the P3 block in Figure 3).

### *3.1. Data Preprocessing and Augmentation*

In this work, three benchmarks were used with the experiments, these were the (*i*) Landsat-8w3c, (*ii*) Landsat-8w5c, and (*iii*) ISPRS Vaihingen (Stuttgart) Challenge data sets. Before a discussion about the model, it is important to deploy a data preprocessing, e.g., pixel standardization, scale pixel values (to have unit variance), and a zero mean into the data sets. In the image domain, the mean subtraction, calculated by the per-channel mean from the training set, is executed in order to improve the model convergence.

Furthermore, a data augmentation (also known as the "ImageDataGenerator" function in TensorFlow/Keras library) is employed, since it can help the model to avoid an overfitting issue and somewhat enlarge the training data—a strategy used to increase the amount of data. To augment the data, each image is width and height-shifted and flipped horizontally and vertically. Then, unwanted outer areas are removed into $512 \times 512$ pixels with a resolution of 81 $cm^2$/pixel in the ISPRS and 900 $m^2$/pixel in the Landsat-8 data set.

### *3.2. The GCN with High-Resolution Representations (HR) Front-End*

The GCN152-TL-A [12], as shown in Figure 1, is our prior attempt that surpasses a traditional semantic segmentation model, e.g., deep convolutional encoder-decoder (DCED) networks [19–21]. By using GCN as our core model, our previous work was improved in three aspects. First, its backbone was revised by varying ResNet-50, ResNet-101, and ResNet-152 networks, as shown in M1 in Figure 1. Second, the "Channel Attention Mechanism" was employed (shown in M2 in Figure 1). Third, the "Domain-Specific Transfer Learning" (TL) was employed to reuse the pre-trained weights obtained from training on other data sets in the remote sensing domain. This strategy is important in the deep learning domain to overcome the limited amount of training data. In our work, there are two main data sets: Landsat-8 and ISPRS. To train the Landsat-8 model, the pre-trained network is obtained by utilizing the ISPRS data. This can also be explained conversely—the pre-trained network can be obtained by Landsat-8 data.

Although the GCN152-TL-A network has determined an encouraging forecast performance, it can still be possible to improve it further through changing the frontend using high-resolution representation (HR) [37,38] instead of ResNet-152 [25,45]. HR has surpassed all existing deep learning methods on semantic segmentation, multi-person pose estimation, object detection, and pose estimation tasks in the COCO, which is large-scale object detection, segmentation, and captioning corpora. It is a parallel structure to enable the deep learning model to link multi-resolution subnetworks in an effective and modern way. HR connects high-to-low subnetworks in parallel. It maintains high-resolution representations through the whole process for a spatially precise heatmap estimation. It creates reliable high-resolution representations through repeatedly fusing the representations generated by the high-to-low subnetworks. It introduces "exchange units" which shuttle across different subnetworks, enabling each one to receive information from other parallel subnetworks. Representations of HR can be obtained by repeating this process. There are four stages as the 2nd, 3rd, 4th, and 5th stages are formed by repeating modularized multi-resolution blocks. A multi-resolution block consists of a multi-resolution group convolution and a multi-resolution convolution, which is illustrated as P1 in Figure 3 (backbone model) and this proposed method is named the "HR-GCN" method.

### *3.3. Feature Fusion*

Inspired by the idea of feature fusion [41–44] that integrates multiplication, additional, or concatenate layers. Convolution with $1 \times 1$ filters is used to transform features with different dimensions into the shape, which can be fused. The fusion method contains an addition process. Each layer of the backbone network such as VGG, Inception, ResNet, or HR creates the feature map for specific. We proposed to combine output with low-level features (front-end network) with the deep model and refine the feature information.

As shown in Figure 4, the kernel maps after fusing will be calculated as Equation (1) :

$$Z_{add} = X_1 \oplus X_2 \oplus X_3 \cdots \oplus X_i \cdots \oplus X_j \tag{1}$$

where $j$ adverts to the index of the layer, $X_k$ is a set of output activation maps of one layer and $\oplus$ adverts to element-wise addition.

Hence, the nature of the addition process encourages essential information to build classifiers to comprehend the feature details. It denotes all bands of $Z_{add}$ to hold more feature information.

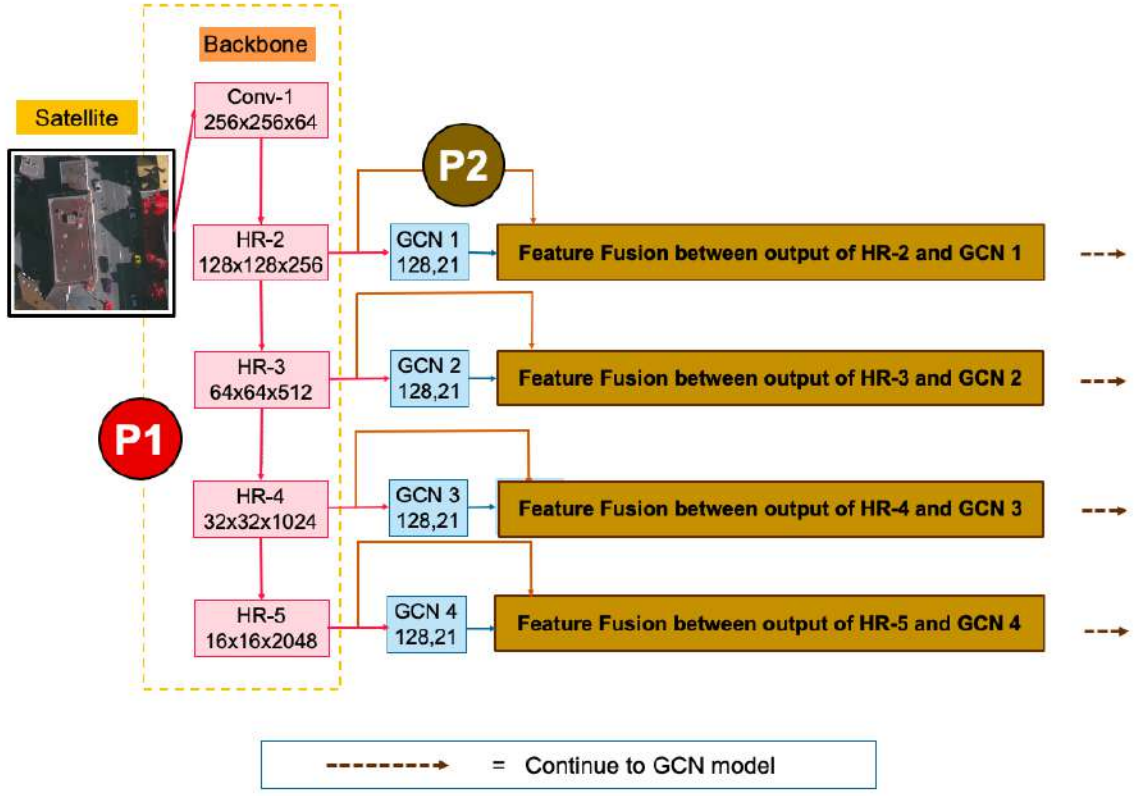



**Figure 4.** The framework of our feature fusion strategy.

Equation (2) shows the relationship between input and output. Thus, we take the fusion activation map into the model again, it can be performed as Equation (4):

$$\bar{y}^i = ReLU(w^T x^i + b) \tag{2}$$

where $x$ is the input and output of layer of the convolution recorded as $y^i$; $b$ and $w$ refer to bias and weight. The cost function in this work is demonstrated via Equation (3).

$$J(w,b) = -\frac{1}{m} \times [(1 - y^{(i)}) log(1 - \bar{y}^{(i)} + (y^{(i)} log(\bar{y}^{(i)})] \tag{3}$$

where $y$ refers to segmentation target of input (each image) and $J$, $w$, and $b$ are the loss, weight, and bias value, respectively.

$$Y_{add} = f(W_k Z_{add} + B_k) \tag{4}$$

The feature fusion procedure always transforms into the same thing when using additional procedures. In this work, we use addition fusion elements, as shown in Figure 4.

### *3.4. Depthwise Atrous Convolution (DA)*

Depthwise Atrous Convolution (DA) [6,9,39] is presented to settle the contradictory requirements between the larger region of the input space that affects a particular unit of the deep network (receptive fields) and activation map resolution.

DA is a robust operation to reduce the number of parameters (weights) in the layer of the CNN while maintaining a similar performance that includes the computation cost and tunes the kernel's field-of-view in order to capture a generalized standard convolution operation and multi-scale information. An atrous filter can be a dilated kernel in varied rates, e.g., rate = 1, 2, 4, 8, by inserting zeros into appropriate positions in the kernel mask.

Basically, the DA module uses atrous convolutions to aggregate multi-scale contextual information without dissipating resolution orderly in each layer. It generalizes "Kronecker-factored" convolutional kernels, and it allows for broad receptive fields, while only expanding the number of weights logarithmically. In other words, DA can apply the same kernel at distinct scales using various atrous factors.

Compared to the ordinary convolution operator, atrous (dilated) convolution is able to achieve a larger receptive field size without increasing the numbers of kernel parameters.

Our motivation is to apply DA to solve challenging scale variations and to trade off precision in aerial and satellite images, as shown in Figure 5.

In a one-dimensional (1D) case, let $x[i]$ denote input signal, and $y[i]$ denote output signal. The dilated convolution is formulated as Equation (5):

$$y[i] = \sum_{j=1}^{J} x[i + a \cdot k] \cdot w[j] \tag{5}$$

where $a$ is the atrous (dilated) rate, $w[j]$ denotes the $j$-th parameter of the kernel, and $J$ is the filter size. This equation reduces to a standard convolution when d = 1, 2, 4, and 8, respectively.

In the cascading mode from DeepLabV3 [46,47] and Atrous Spatial Pyramid Pooling (ASPP) [9], multi-scale contextual information can be encoded by probing the incoming features with dilated convolution to capture sharper object boundaries by continuously recovering the spatial characteristic. DA has been applied to increase the computational ability and achieve the performance by factorizing a traditional convolution into a depth-wise convolution followed by a point-wise convolution, such as $1 \times 1$ convolution (it is often applied on the low-level attributes to decrease the whole of the bands (kernel maps)).

To simplify notations, $H_{J,a}(x)$ is term of a dilated convolution, and ASPP can be performed as Equation (6).

$$y = H_{3,1}(x) + H_{3,2}(x) + H_{3,4}(x) + H_{3,8}(x) \tag{6}$$

To improve the semantics of shallow features, we apply the idea of multiple dilated convolution with different sampling rates to the input kernel map before continuing with the decoder network and adjusting the dilation rates (1, 2, 4, and 8) to configure the whole process of our proposed method called "HR-GCN-FF-DA", shown in P3 in Figures 3 and 5.

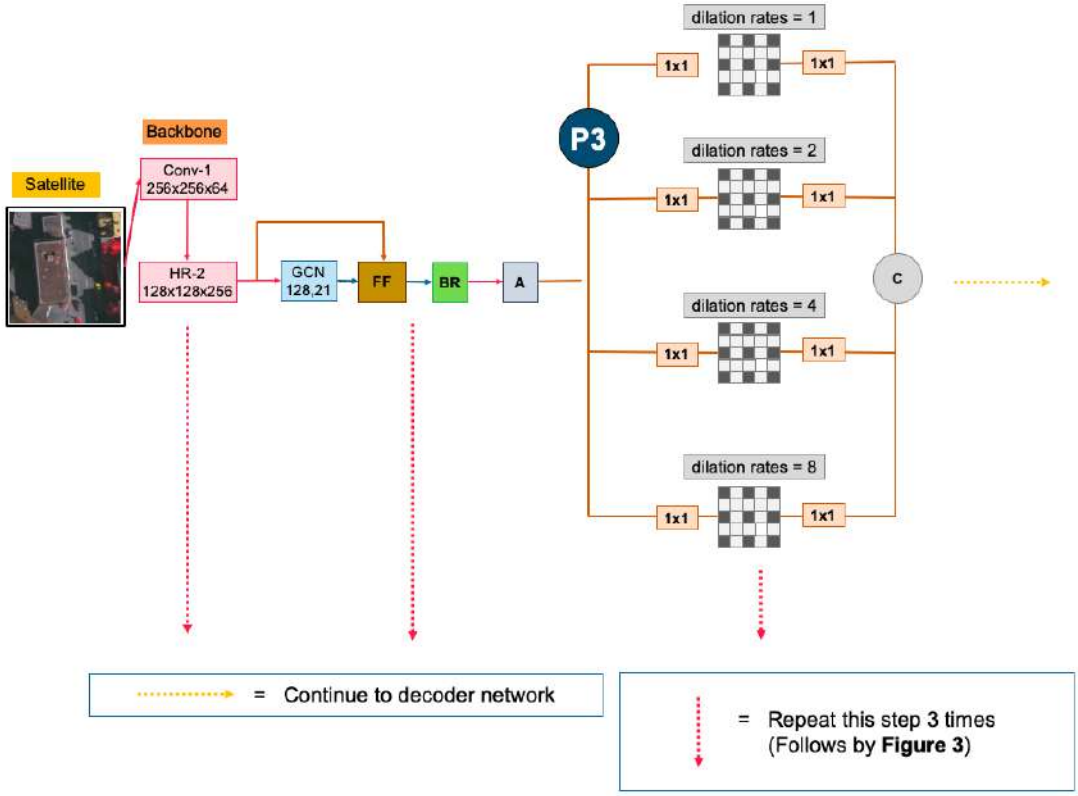



**Figure 5.** The Depthwise Atrous Convolution (DA) module in the proposed parallel pyramid method for improving feature fusion.

## 4. Remote Sensing Corpora

In our experiments, there are two main sources of data: public and private corpora. The private corpora is the medium resolution imagery received from the satellite "Landsat-8" used by the government organization in Thailand called GISTDA. Since there are two variations of annotations, the Landsat-8 data is considered as two data sets: one with three classes and the other with five classes, as shown in Table 1. The public corpora is very high-resolution imagery from the standard benchmark called "ISPRS Vaihingen (Stuttgart)". Evaluations based on classification/segmentation metrics, e.g., *F*1 *Score*, *Precision*, *Recall* and *Average Accuracy* are deployed with all experiments.

**Table 1.** Abbreviations on our Landsat-8 corpora.

| Abbreviation | Description |
|---|---|
| Landsat-8w3c corpus | Landsat-8 corpus with 3 classes |
| Landsat-8w5c corpus | Landsat-8 corpus with 5 classes |

### *4.1. Landsat-8w3c Corpus*

For this corpus, there is a new benchmark that differs from our previous work. All images are taken in the area of the northern provinces (Changwat) of Thailand. The data set is made from the Landsat-8 satellite consisting of 1420 satellite images, some samples are shown in Figure 6. This data set contains a massive collection of medium resolution imagery of (20,921 $\times$ 17,472) pixels. There are three classes: para rubber (red), pineapple (green), and corn (yellow). From a total of 1390 images, the images are separated into 1000 training and 230 validation images, as well as 190 test images to compare with other baseline methods.

### *4.2. Landsat-8w5c Corpus*

This data set is the same corpus from Landsat-8, but it is annotated with five class labels: agriculture, forest, miscellaneous (misc), urban, and water as shown in Figure 7. There are 1012 medium resolution satellite images of 17,200 × 16,300 pixels. From the total 1039 images, the images are separated into 700 training and 239 validation images, as well as 100 test images to comparison to other baseline methods.

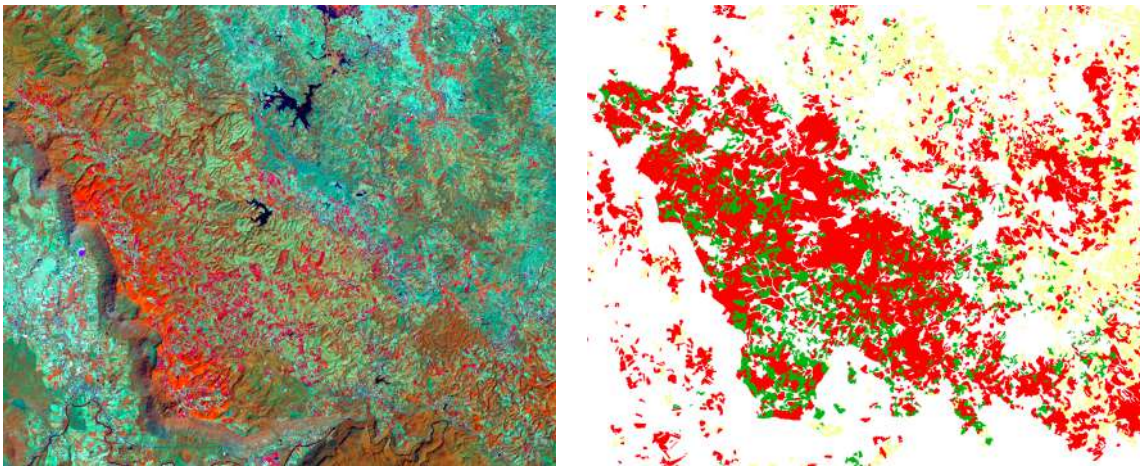


**Figure 6.** The example of satellite images from the Landsat-8w3c corpus, northern province (left) and target image (right). The ground-truth of the medium resolution data set includes three classes: para rubber (red), pineapple (green), and corn (Yellow).

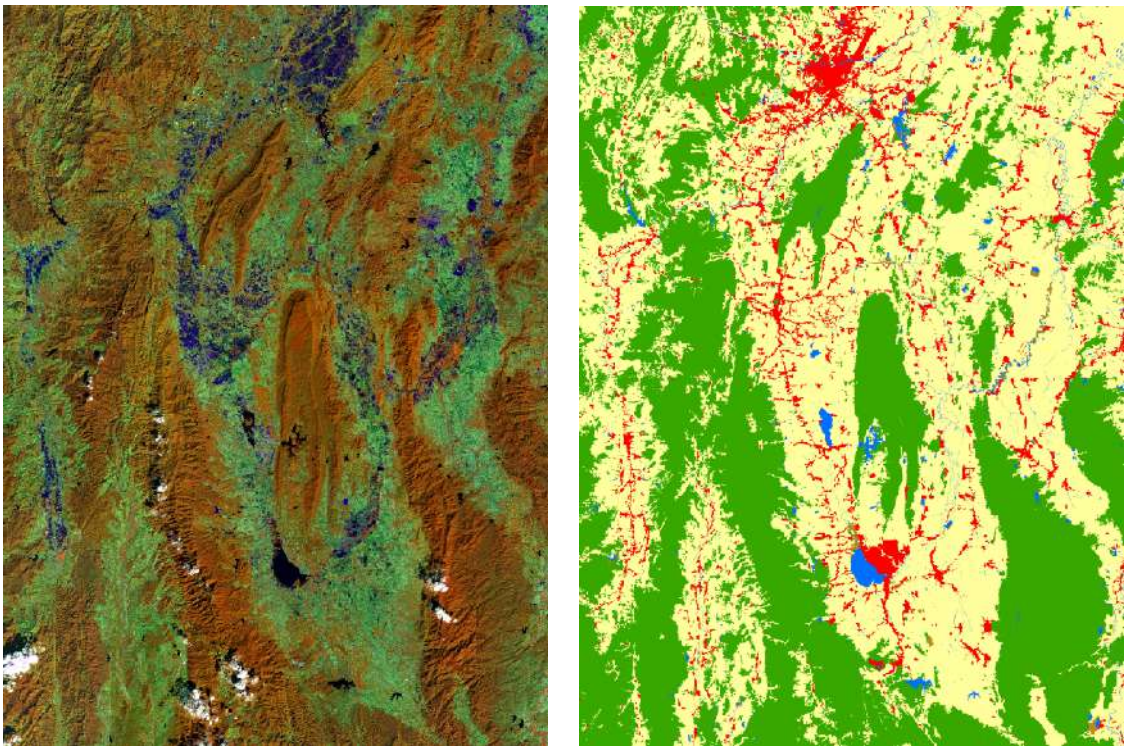


**Figure 7.** The example of satellite images from Landsat-8w5c corpus, northern province (left) and target image (right). The ground-truth of medium resolution data set includes five classes: urban (red), forest (green), water (blue), agriculture or harvested area (yellow), and miscellaneous or misc (brown).

### *4.3. ISPRS Vaihingen Corpus*

The challenge of ISPRS semantic segmentation at Vaihingen (Stuttgart) [18] (Figures 8 and 9) is used to be our standard corpus. They were captured over Vaihingen in Germany. The data set is a subset of the data used for the test of digital aerial cameras carried out by the German Association of Photogrammetry and Remote Sensing (DGPF).


**Figure 8.** ISPRS 2D Vaihingen segmentation corpus (33 scenes).

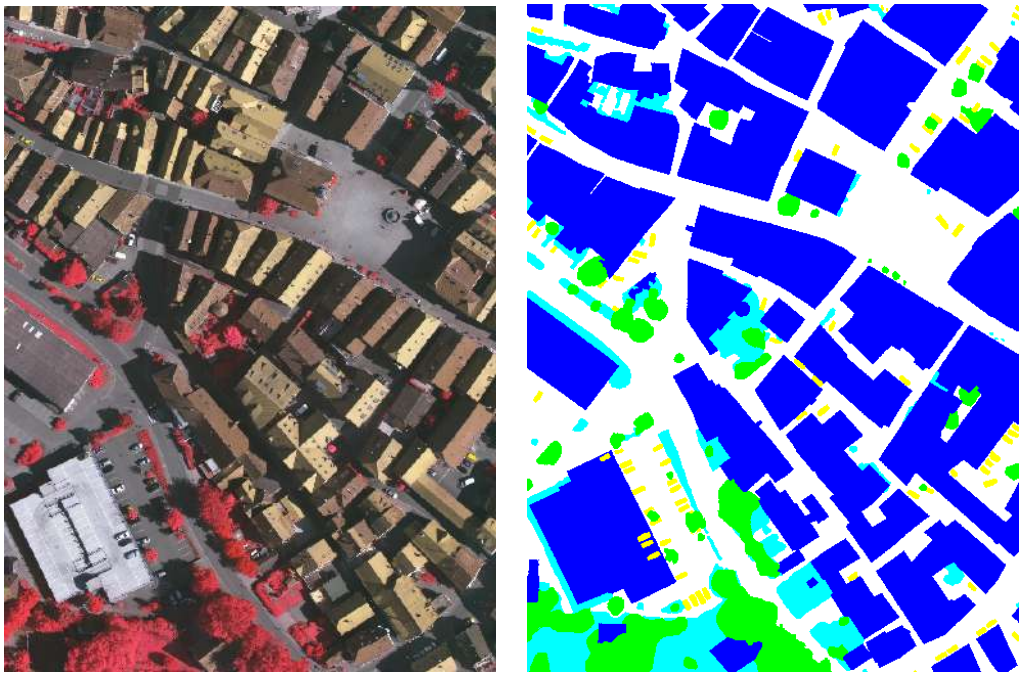


**Figure 9.** Sample of input scene from Figure 8 (left) and target image (right). The annotated Vaihingen corpus has five categories: tree (green), building (blue), clutter/background (red), low vegetation or LV (greenish-blue), and impervious surface or imp surf (white)

It consists of three spectral bands such as NDSM, DSM, near-infrared bands, red, and green data. For our work, NDSM and DSM data are not used in this corpus. They provide 33 images of about 2500 $\times$ 2000 pixels of about 9 cm of resolution. Following other methods, four scenes such as scene 5, 7, 23, and 30 are removed from the training set as a validation set. All experimental results are announced on the validation set if not specified.

## 5. Performance Evaluation

The performance of "HR-GCN-FF-DA" is evaluated in all corpora for $F1$ and $AverageAccuracy$. To assess class-specific performance, the $F1$, $precision$, $recall$, and $AverageAccuracy$ metric are used. It is computed as the symphonious average between recall and precision. We carry $precision$, $recall$, and $F1$ as fundamental metrics and also incorporate the $AverageAccuracy$, which calculates the number of correctly classified positions and divides it by the total number of the reference positions. The $AverageAccuracy$ and $F1$ metrics can be assessed using Equations (7)–(10).

The confusion matrix for pixel-level classification [18] and the false positive (denoted as FP) are computed from the summation of the column. In contrast, the false negative (denoted as FN) is the summation of the horizontal axis, excluding the principal diagonal factor. Next, the true positive (denoted as TP) is the value of the identical oblique elements, and the true negative (denote as TN) contrasts TP.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{9}$$

$$AverageAccuracy = \frac{TP + TN}{TP + + FP + FN + TN} \tag{10}$$

## 6. Experimental Results

For a python deep learning framework, we use "Tensorflow (TF)" [48], an end-to-end open source platform for deep learning. The whole experiment was implemented on servers with Intel® 2066 core I9-10900X, 128 GB of memory, and the NVIDIA RTX™ 2080Ti (11 GB) x 4 cards.

For the training phrase, the adaptive learning rate optimization algorithm (extension to the stochastic gradient descent (SGD)) [49] and batch normalization [50], a technique for improving the performance, stability, and speed of deep learning, were applied and standardized to ease the training in every experiment.

For the learning rate schedules tasks, [9,16,32], we selected the polylearning rate policy. As shown in Equation (11), the learning rate is scheduled by multiplying a decaying factor to the initial learning rate ($4 \times 10^{-3}$).

$$learning\ rate = init\_learning\ rate \times (1 - \frac{epoch}{MaxEpoch})^{0.9} \tag{11}$$

All deep CNN models are trained for 30 epochs on the Landsat-8w3c corpus and ISPRS Vaihingen data sets. It is increased to be 50 epochs for the Landsat-8w5c data set. Each image is resized to 521 $\times$ 521 pixels along with augmented data using a randomly cropping strategy. Weights are updated using the mini-batch of 4.

This section explains the elements of our experiments. The proposed CNN architecture is based on the victor from our previous work called "GCN152-TL-A" [12]. In our work, there are three proposed improvements: (*i*) adjusting backbones using high-resolution representations, (*ii*) the feature fusion module, and (*iii*) depthwise atrous convolution. From all proposed policies, there are four acronyms of procedures, as shown in Table 2.

**Table 2.** Acronyms of our proposed deep learning approaches.

| Acronym | Representation |
|---|---|
| A | Channel Attention Block |
| DA | Depthwise Atrous Convolution |
| FF | Feature Fusion |
| HR | High-Resolution Representations |

There are three subsections to discuss the experimental results of each data set: (*i*) Landsat-8w3c, (*ii*) Landsat-8w5c, and (*iii*) ISPRS Vaihingen data sets.

There are two baseline models of the semantic labeling task in the domains of remote sensing-based information on the computer vision. The first baseline is DCED, which is commonly used in much segmentation work [19–21]. The second baseline is the winner of our previous work called "GCN152-A-TL" [12]. Note that "GCN-A-TL" is abbreviated using just "GCN", since we always employ the attention and transfer-learning strategies into our proposed models.

Each proposed tactic can elevate the completion of the baseline approach shown via the whole experiment. First, the effect of our new backbone (HRNET) is investigated by using HRNET on the GCN framework called "HR-GCN". Second, the effect of our feature fusion is shown by adding it into our model, called "HR-GCN-FF". Third, the effect of the depthwise atrous convolution is explained by using it on top of a traditional convolution mechanism, called "HR-GCN-FF-DA".

### *6.1. The Results of Landsat-8w3c Data Set*

The Landsat-8w3c corpus was used in all experiments. We distinguished between the alterations of the proposed approaches and CNN baselines. "HR-GCN-FF-DA", the full proposed method, is the winner with $F1$ of 0.9114. Furthermore, it is also the winner of all classes. More detailed results are given in the next subsection. Presented in Tables 3 and 4 are the results of this corpus, Landsat-8w3c.

#### 6.1.1. HR-GCN Model: Effect of Heightened GCN with High-Resolution Representations on Landsat-8w3c

The previous enhanced GCN network is improved to increase the $F1$ score by using the High-Resolution Representations (HR) backbone instead of the ResNet-152 backbone (best frontend network from our previous work). $F1$ of HR-GCN (0.8763) outperforms that of the baseline methods. DCED (0.8114) and GCN152-TL-A (0.8727) refer to Tables 3 and 4. The result returns a higher $F1$ at 6.50% and 0.36%, respectively. Hence, it means the features extracted from HRNET are better than those from ResNet-152.

For the analysis of each class, HR-GCN achieved an average accuracy on para rubber, pineapple, and corn for 0.8371, 0.8147, and 0.8621, consecutively. Compared to DCED, it won in two classes: para rubber and corn. However, it won against our previous work (GCN152-TL-A) only in the pineapple class.

#### 6.1.2. HR-GCN-FF Model: Effect of Using "Feature Fusion" on Landsat-8w3c

Next, we apply "Feature Fusion" to capture low-level features to decorate the feature information of CNN. HR-GCN-FF (0.8852) is higher than that of HR-GCN (0.8763), GCN152-TL-A (0.8727), and DCED (0.8113), shown in Tables 3 and 4. It gives a higher $F1$ score at 0.89%, 1.26%, and 7.39%, consecutively.

It is interesting that the FF module can really improve the performance in all classes, especially in the para rubber and pineapple classes. It outperforms both HR-GCN and all baselines in all classes. To further investigate the results, Figures 10e and 11e show that the model with FF can capture pineapple (green area) surrounded in para rubber (red area).

**Table 3.** Effects on the testing set of the Landsat-8w3c data set.

| | Pretrained | Frontend | Model | *Precision* | *Recall* | *F*1 |
|---|---|---|---|---|---|---|
| **Baseline** | - | VGG16 | DCED [19–21] | 0.8546 | 0.7723 | 0.8114 |
| | TL | Res152 | GCN-A [12] | 0.8732 | 0.8722 | 0.8727 |
| **Proposed Method** | TL | HRNET | GCN-A | 0.8693 | 0.8836 | 0.8764 |
| | TL | HRNET | GCN-A-FF | 0.8797 | 0.8910 | 0.8853 |
| | TL | HRNET | GCN-A-FF-DA | **0.8999** | **0.9233** | **0.9114** |

**Table 4.** Effects on the testing set of the Landsat-8w3c data set among each class with our proposed procedures in terms of *AverageAccuracy*.

| | Model | Para Rubber | Pineapple | Corn |
|---|---|---|---|---|
| **Baseline** | DCED [19–21] | 0.8218 | 0.8618 | 0.8084 |
| | GCN152-TL-A [12] | 0.9127 | 0.7778 | 0.8878 |
| **Proposed Method** | HR-GCN | 0.8371 | 0.8147 | 0.8621 |
| | HR-GCN-FF | 0.9179 | 0.8689 | 0.8989 |
| | HR-GCN-FF-DA | **0.9386** | **0.8881** | **0.9184** |

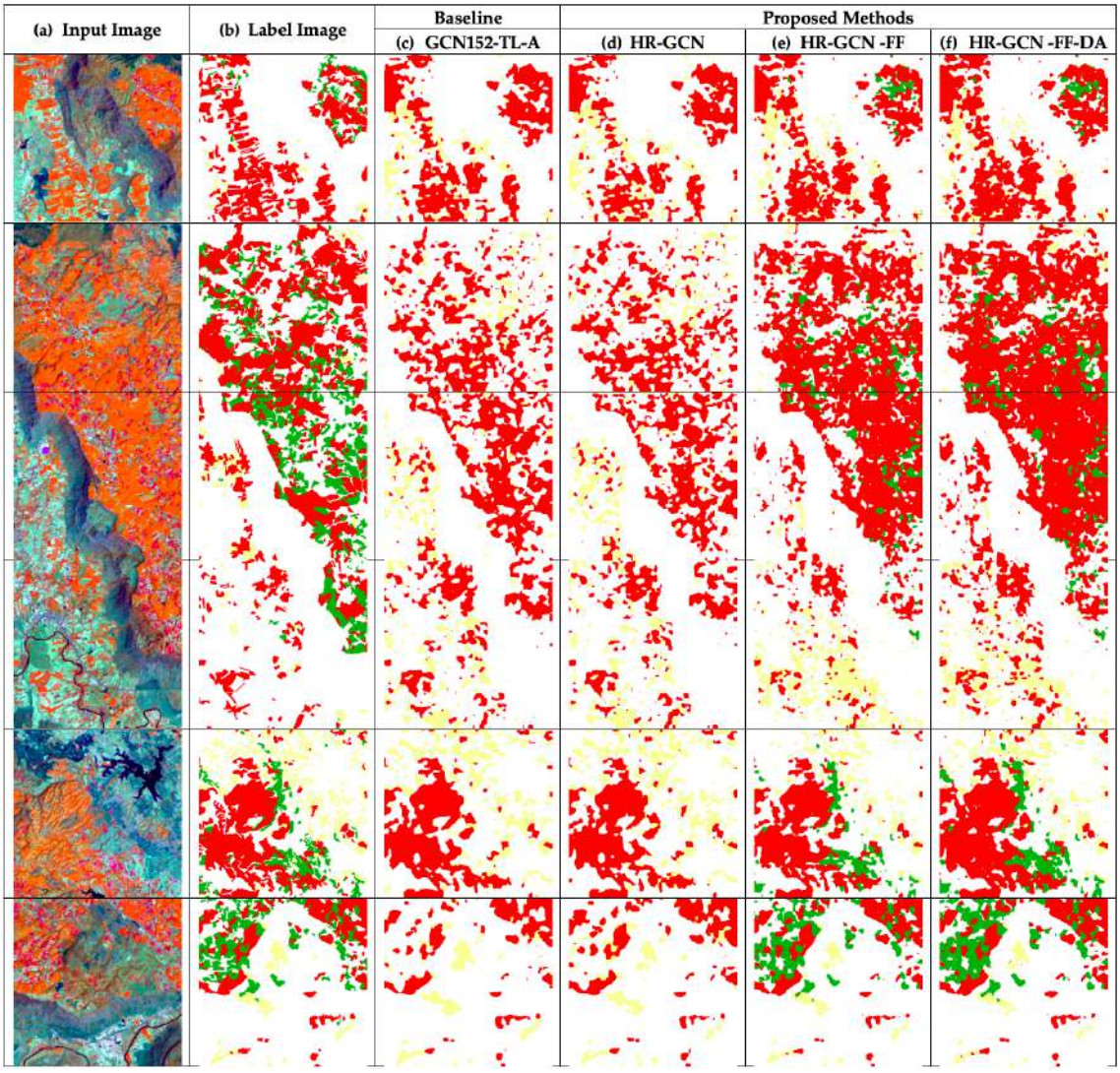



**Figure 10.** Comparisons between "HR-GCN-FF-DA" and other published methods of the Landsat-8w3c corpus testing set.

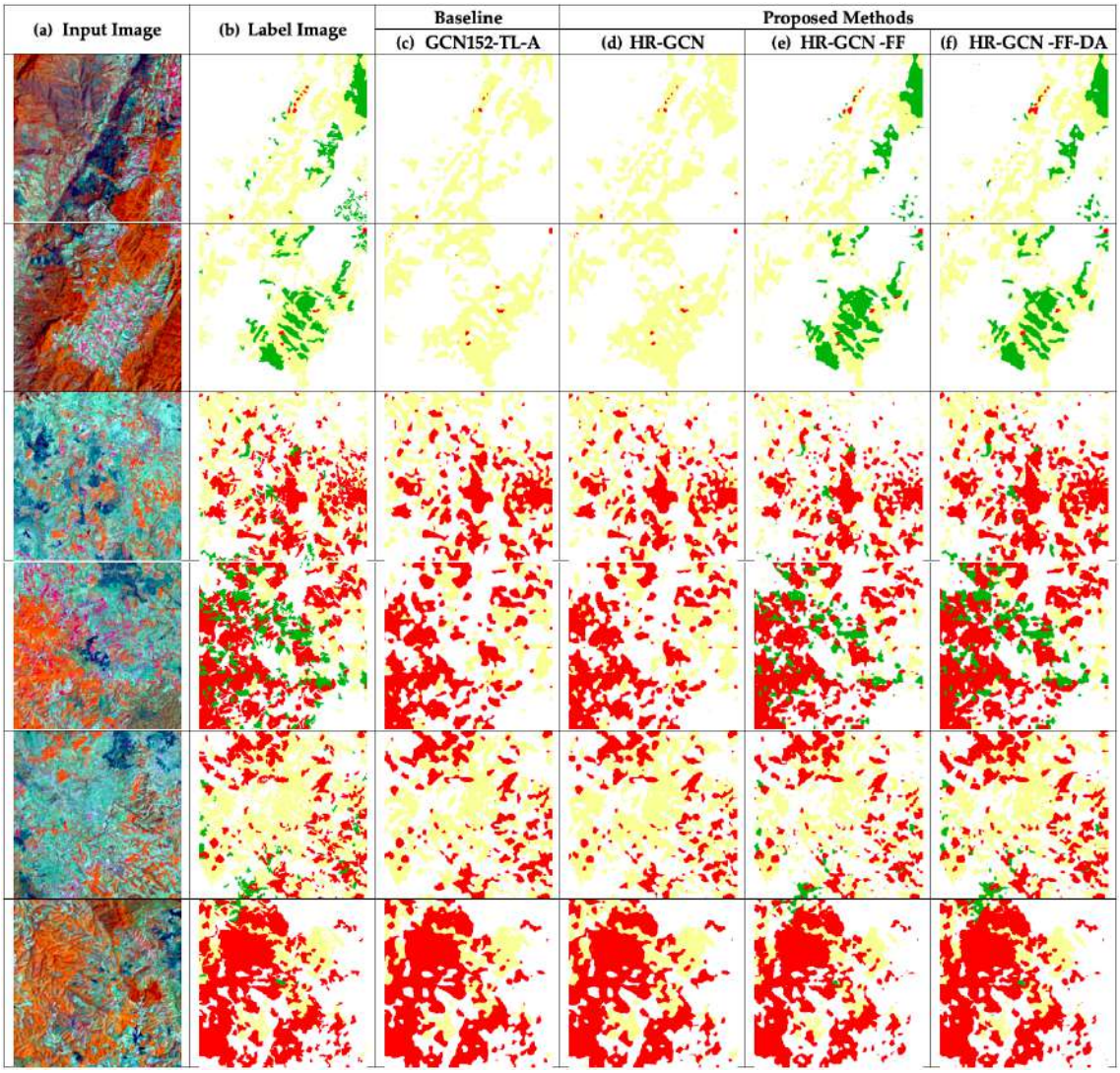


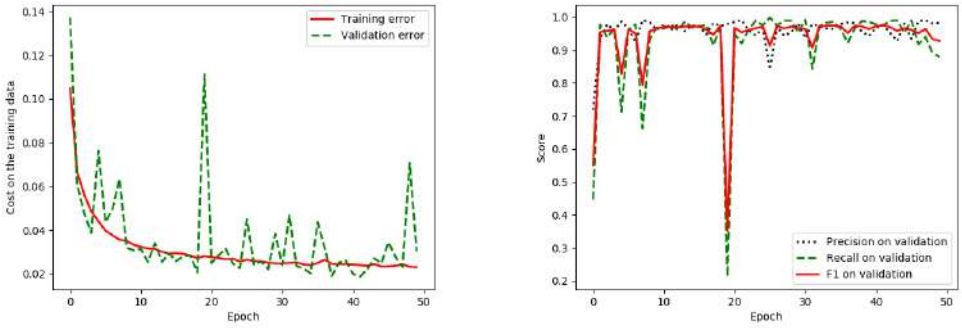

**Figure 11.** Comparisons between "HR-GCN-FF-DA" and beyond baseline methods of the Landsat-8w3c corpus testing set.

#### 6.1.3. HR-GCN-FF-DA Model: Effect of Using "Depthwise Atrous Convolution" on Landsat-8w3c

The last strategy aims to use an approach of "Depthwise Atrous Convolution" (details in Section 3.4) by extracting complementary information from very shallow features and enhancing the deep features for improving feature fusion of the Landsat-8w3c corpus. The "HR-GCN-FF-DA" method is the victor. $F1$ is obviously more distinguished than DCED at 10.00% and GCN152-TL-A (the best benchmark) at 3.87%, as shown in Tables 3 and 4.

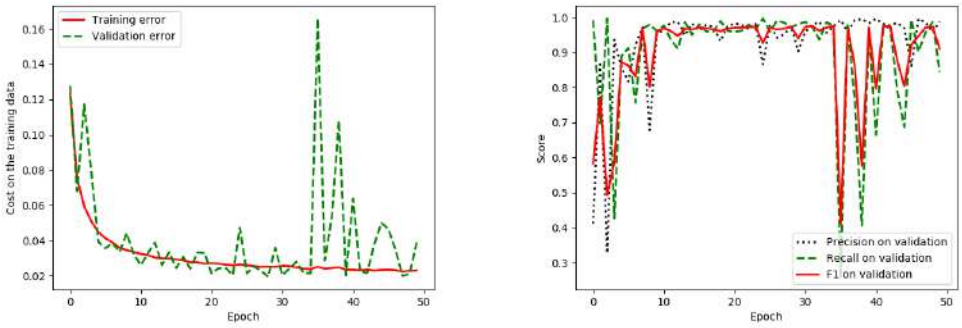
For an analysis of each class, our model is clearly the winner in all classes with an accuracy beyond 90% in two classes: para rubber and corn. Figures 10 and 11 show twelve sample outputs from our proposed methods (column $(d\ tof)$) compared to the baseline (column $(c)$) to expose improvements in its results. From our investigation, we found that the dilated convolutional concept can make our model have better overview information, so it can capture larger areas of data.

There is a lower discrepancy (peak) in the validation data of "HR-GCN-FF-DA", Figure 12a, than that in the baseline, Figure 13a. Moreover, Figures 13b and 12b show three learning graphs such as precision, recall, and F1 lines. The loss graph of the "HR-GCN-FF-DA" model seems flatter (very smooth) than the baseline in Figure 13a. The epoch at number 27 was selected to be a pre-trained model for testing and transfer learning procedures.




(**a**) (**b**)

**Figure 12.** Graph (learning curves) of the Landsat-8w3c data set of the proposed approach, "HR-GCN-FF-DA"; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.




(**a**) (**b**)

**Figure 13.** Graph (learning curves) of the Landsat-8w3c data set of the baseline approach, GCN152-TL-A; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus. [12].

### *6.2. The Results on Landsat-8w5c Data Set*

In this subsection, the Landsat-8w5c corpus was conducted on all experiments. We compare "HR-GCN-FF-DA" network (column $(f)$) to CNN baselines via Tables 5 and 6. "HR-GCN-FF-DA" is the winner with a $F1$ of 0.9111. Furthermore, it is also the winner in all classes especially water and urban class that are composed with low-level features. More detailed results are described in the next subsection and are presented in Tables 5 and 6 for the results of this data set, Landsat-8w5c.

#### 6.2.1. HR-GCN Model: Effect of Heightened GCN with High-Resolution Representations on Landsat-8w5c

The $F1$ score of HR-GCN (0.8897) outperforms that of baseline methods: DCED (0.8505) and GCN152-TL-A (0.8791); $F1$ at 3.92% and 1.07% respectively. The main reason is due to both higher recall and precision. This can imply that features extracted from HRNET are also better than those from ResNet-152 on Landsat-8 images as well, shown in Tables 5 and 6.

**Table 5.** Effects on the testing set of Landsat-8w5c data set.

| | Pre-trained | Frontend | Model | *Precision* | *Recall* | *F*1 |
|---|---|---|---|---|---|---|
| **Baseline** | - | VGG16 | DCED [19–21] | 0.8571 | 0.8441 | 0.8506 |
| | TL | Res152 | GCN-A [12] | 0.8616 | 0.8973 | 0.8791 |
| **Proposed Method** | TL | HRNET | GCN-A | 0.8918 | 0.8877 | 0.8898 |
| | TL | HRNET | GCN-A-FF | 0.9209 | 0.9181 | 0.9195 |
| | TL | HRNET | GCN-A-FF-DA | **0.9338** | **0.9385** | **0.9362** |

**Table 6.** Effects on the testing set of Landsat-8w5c data set among each class with our proposed procedures in terms of *AverageAccuracy*.

| | Model | Agriculture | Forest | Misc | Urban | Water |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [19–21] | 0.9819 | **0.9619** | 0.7628 | 0.8538 | 0.7250 |
| | GCN152-TL-A [12] | 0.9757 | 0.9294 | 0.6847 | 0.9288 | 0.7846 |
| **Proposed Method** | HR-GCN | 0.9755 | 0.9501 | 0.8231 | 0.9133 | 0.7972 |
| | HR-GCN-FF | 0.9741 | 0.9526 | 0.8641 | 0.9335 | 0.8282 |
| | HR-GCN-FF-DA | **0.9856** | 0.9531 | **0.9176** | **0.9561** | **0.8437** |

For the analysis on each class, HR-GCN achieved an averaging accuracy in agriculture, forest, miscellaneous, urban, and water for 0.9755, 0.9501, 0.8231, 0.9133, and 0.7972, consecutively. Compared to DCED, it won in three classes: forest, miscellaneous and water. However, it won against our previous work (GCN152-TL-A) in the pineapple class, and it showed about the same performance in the agriculture class.

#### 6.2.2. HR-GCN-FF Model: Effect of Using "Feature Fusion" on Landsat-8w5c

The second mechanism focuses on utilizing 'Feature Fusion" to fuse each level feature for enriching the feature information. From Tables 5 and 6, the $F1$ of HR-GCN-FF (0.9195) is greater than that of HR-GCN (0.8897), GCN152-TL-A (0.8791), and DCED (0.8505). It produces a more precise $F1$ score at 2.97%, 4.04%, and 6.89%.

To further analyze the results, Figures 14e and 15e show that the FF module can better capture low-level details. Especially in the water class, it can recover the missing water area, resulting in an improvement of accuracy from 0.7972 to 0.8282 (3.1%).

#### 6.2.3. HR-GCN-FF-DA Model: Effect of Using "Depthwise Atrous Convolution" on Landsat-8w5c

The last policy points to the performance of the method of "Depthwise Atrous Convolution" by enhancing the features of CNN for improving the previous step. The $F1$ score of the "HR-GCN-FF-DA" approach is the conqueror. It is more eminent than DCED and GCN152-TL at 8.56% and 5.71%, consecutively, shown in Tables 5 and 6.

In the dilated convolution, filters are boarder, which can capture better overview details resulting in ($i$) larger coverage areas and ($ii$) connected small areas together.

For an analysis of each class, our final model is clearly the winner in all classes with an accuracy beyond 95% in two classes: agriculture and urban classes. Figures 14 and 15 show twelve sample outputs from our proposed methods (column $(d\ to\ f)$) compared to the baseline (column $(c)$) to expose improvements in its results and that founds that Figures 14f and 15f are likewise to the ground images. From our investigation, we found that the dilated convolutional concept can make our model have better overview information, so it can capture larger areas of data.

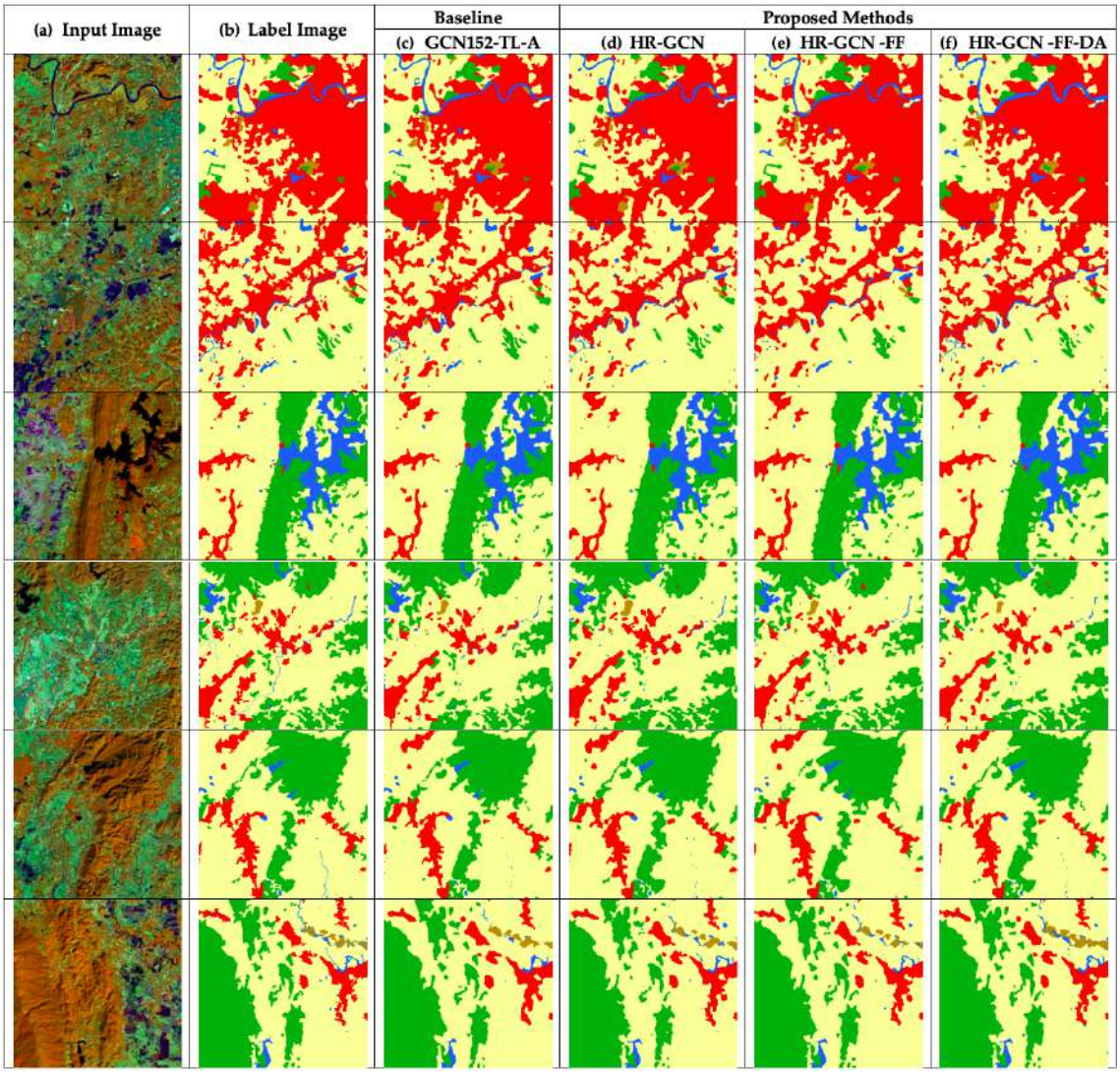



**Figure 14.** Comparisons between "HR-GCN-FF-DA" and beyond baseline methods on the Landsat-8w5c corpus testing set.

Considering the loss graphs, our model in Figure 16a can learn smoother than the baseline (our previous work) in Figure 17a, since the discrepancy (peak) in the validation error (green line) is lower in our model. There is a lower discrepancy (peak) in the validation data of "HR-GCN-FF-DA", Figure 16a, than that in the baseline, Figure 17a. Moreover, Figures 17b and 16b show three learning graphs such as precision, recall, and F1 lines. The loss graph of the "HR-GCN-FF-DA" model seems flatter (very smooth) than the baseline in Figure 17a and the epoch at number 40 out of 50 was selected to be a pre-trained model for testing and transfer learning procedures.

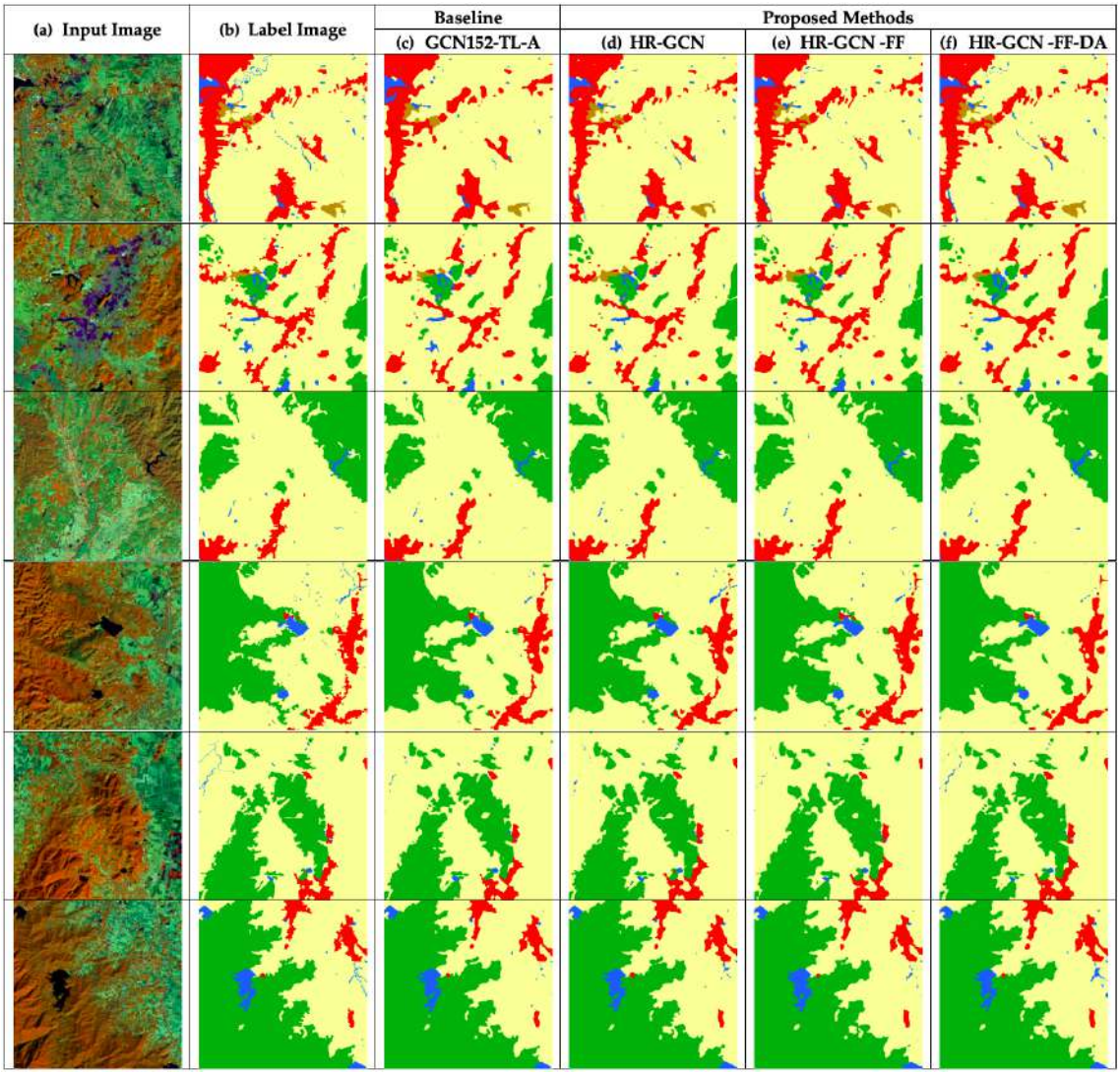
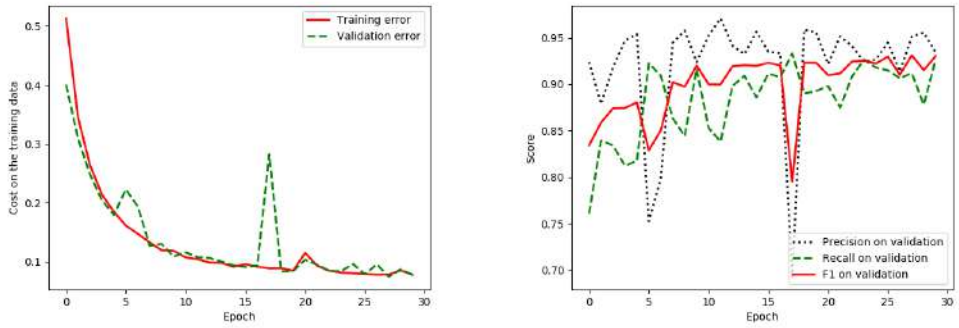
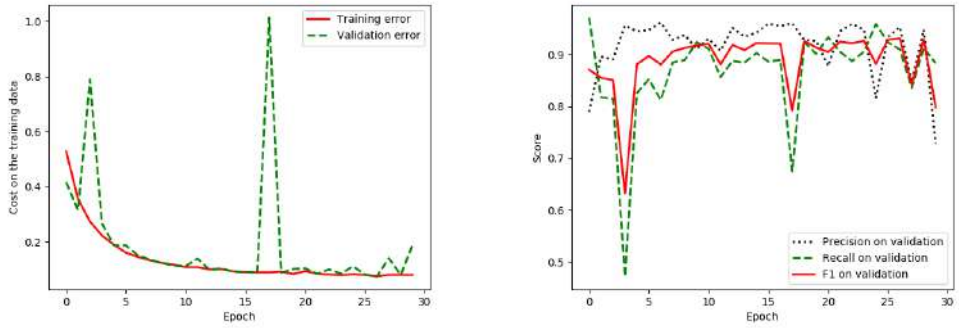



**Figure 15.** Comparisons between "HR-GCN-FF-DA" and beyond baseline methods on the Landsat-8w5c corpus testing set.




(**a**) (**b**)

**Figure 16.** Graph (learning curves) on Landsat-8w5c data set of the proposed approach, "HR-GCN-FF-DA"; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.




(**a**) (**b**)

**Figure 17.** Graph (learning curves) on Landsat-8w5c data set of the baseline approach, GCN152-TL-A [12]; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.

### *6.3. The Results in ISPRS Vaihingen Challenge Data Set*

In this subsection, the ISPRS Vaihingen (Stuttgart) Challenge corpus was used in all experiments. The "HR-GCN-FF-DA" is the winner with *F*1 of 0.9111. Furthermore, it is also the winner of all classes. More detailed results will be provided in the next subsection, and the consequences of our proposed method with CNN baselines for this data set are shown in Tables 7 and 8.

#### 6.3.1. HR-GCN Model: Effect of Heightened GCN with High-Resolution Representations in ISPRS Vaihingen

The *F*1 score of HR-GCN (0.8701) exceeds that of the baseline methods: DCED (0.8580) and GCN152-TL-A (0.8620). It complies a higher *F*1 at 1.21% and 0.81%, respectively. This shows that the enhanced GCN with HR backbone is also more significantly streamlined than the GCN152-TL-A style, shown in Tables 7 and 8.

The goal of the HR module is to help prevent the loss of some important features, such as low-level features, so it can significantly improve the accuracy of the car class from 0.8034 to 0.8202 (1.68%) and the building class from 0.8725 to 0.9282 (5.57%).

**Table 7.** Effects on the testing set of ISPRS Vaihingen (Stuttgart) challenge data set.

| | **Pre-trained** | **Frontend** | **Model** | ***Precision*** | ***Recall*** | ***F*1** |
|---|---|---|---|---|---|---|
| **Baseline** | - | VGG16 | DCED [19–21] | 0.8672 | 0.8490 | 0.8580 |
| | TL | Res152 | GCN-A [12] | 0.8724 | 0.8520 | 0.8620 |
| **Proposed Method** | TL | HRNET | GCN-A | 0.8717 | 0.8686 | 0.8701 |
| | TL | HRNET | GCN-A-FF | 0.8981 | 0.8812 | 0.8896 |
| | TL | HRNET | GCN-A-FF-DA | **0.9228** | **0.8997** | **0.9111** |

**Table 8.** Effects on the testing set of ISPRS Vaihingen (Stuttgart) challenge data set among each class with our proposed procedures in terms of *AverageAccuracy*.

| | **Model** | **IS** | **Buildings** | **LV** | **Tree** | **Car** |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [19–21] | 0.8721 | 0.8932 | 0.8410 | 0.9144 | 0.8153 |
| | GCN152-TL-A [12] | 0.8758 | 0.8725 | 0.8567 | **0.9534** | 0.8034 |
| **Proposed Method** | HR-GCN | 0.8864 | 0.9282 | 0.8114 | 0.8945 | 0.8202 |
| | HR-GCN-FF | 0.8279 | 0.9458 | 0.9264 | 0.9475 | 0.8502 |
| | HR-GCN-FF-DA | **0.9075** | **0.9589** | **0.9266** | 0.9299 | **0.8710** |

6.3.2. HR-GCN-FF Model: Effect of Using "Feature Fusion" on ISPRS Vaihingen

Next, we propose "Feature Fusion" to fuse each level feature for enriching the feature information. From Tables 7 and 8, the *F*1 of HR-GCN-FF (0.8895) is greater than that of HR-GCN (0.8701), GCN152-TL-A (0.8620), and DCED (0.8580). It also returns a higher *F*1 score at 1.95%, 2.76%, and 3.16%, respectively.

The goal of the FF module is to capture low-level features, so it can significantly improve the accuracy of the low vegetation class (LV) from 0.8114 to 0.9264 (11.5%), the accuracy of the tree class from 0.8945 to 0.9475 (5.3%), and the accuracy of the car class from 0.8202 to 0.8502 (3%). This finding is shown in Figures 18e and 19e.

6.3.3. HR-GCN-FF-DA Model: Effect of Using "Depthwise Atrous Convolution" on ISPRS Vaihingen

Finally, our last approach is to apply "Depthwise Atrous Convolution" to intensify the deep features from the previous step. From Tables 7 and 8 we see that the *F*1 of the "HR-GCN-FF-DA" method is also the conqueror in this data set. The *F*1 score of "HR-GCN-FF-DA" is also more precise than the DCED and GCN152-TL-A at 5.31% and 4.96%, consecutively.

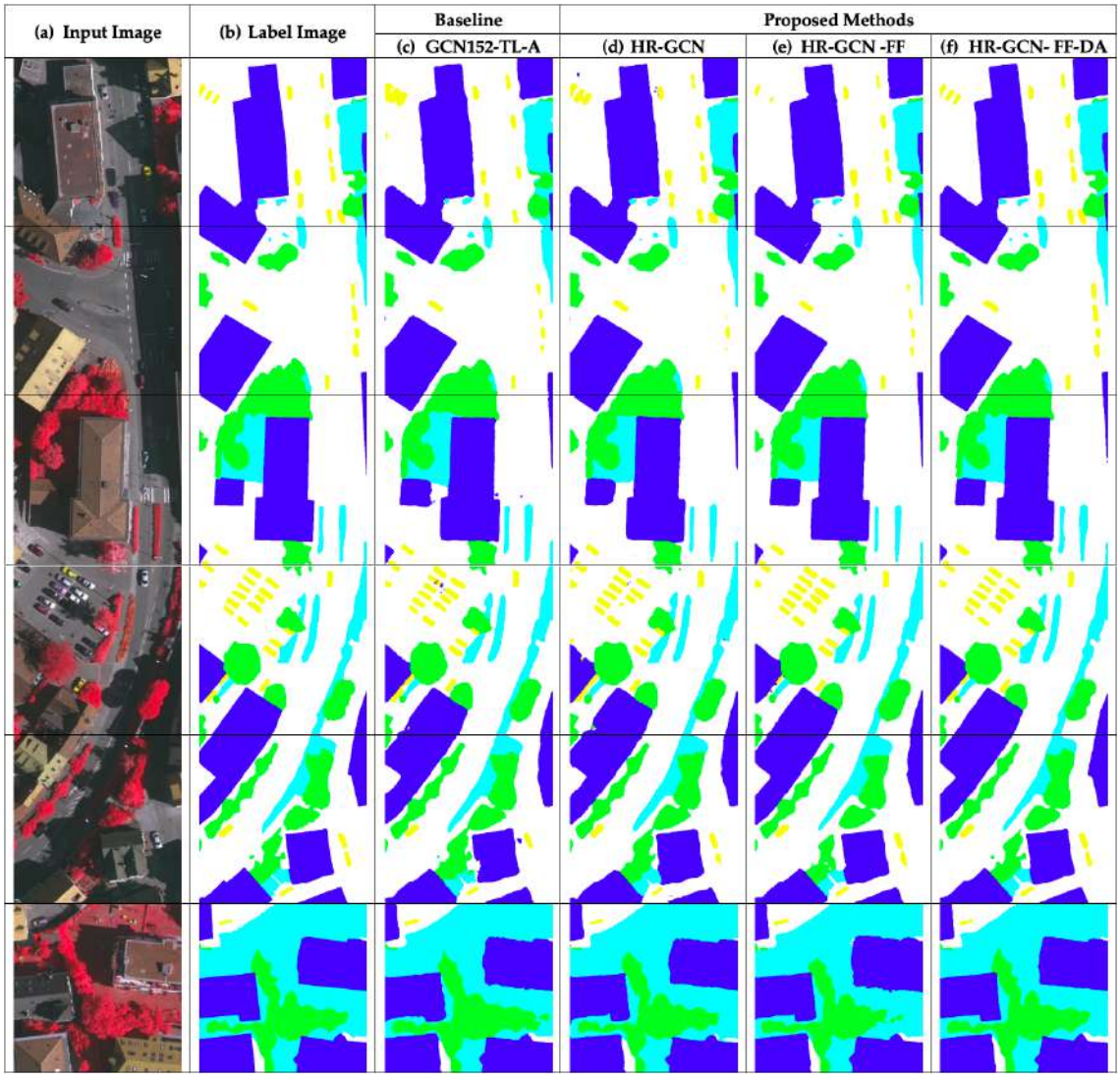



**Figure 18.** Comparisons between "HR-GCN-FF-DA" and beyond baseline methods on the ISPRS Vaihingen (Stuttgart) challenge corpus testing set.

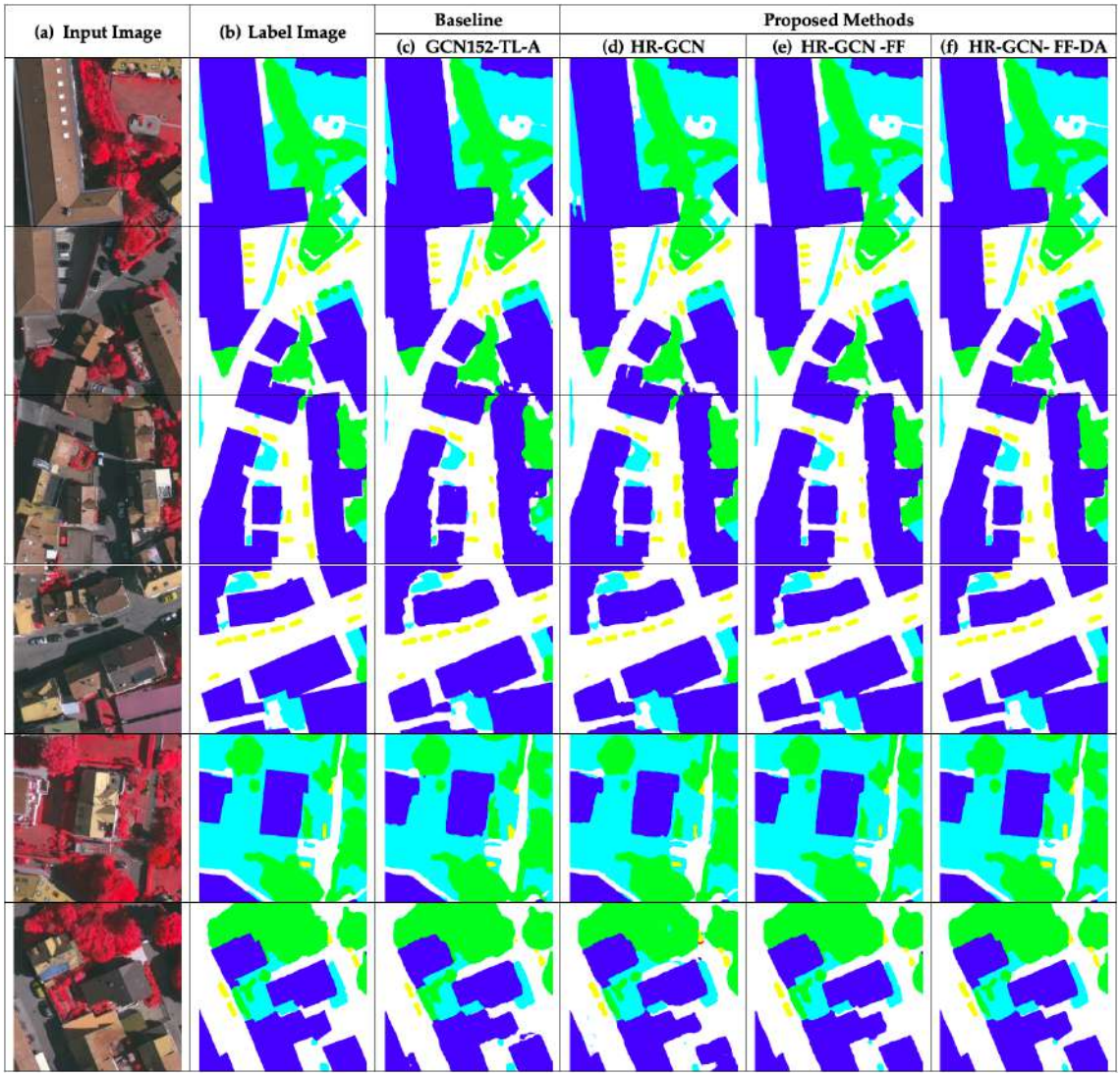



**Figure 19.** Comparisons between "HR-GCN-FF-DA" and beyond baseline methods on the ISPRS Vaihingen (Stuttgart) challenge corpus testing set.

It is very impressive that our model with all its strategies can improve the accuracy in almost all classes to be greater than 90%. Although the accuracy of car is 0.8710, it improves on the baseline (0.8034) by 6.66%.

## 7. Discussion

In the Landsat-8w3c corpus, for an analysis of each class, our model is clearly the winner in all classes with an accuracy beyond 90% in two classes: para rubber and corn. Figures 10 and 11 show twelve sample outputs from our proposed methods (column $(d\ to f)$) compared to the baseline (column $(c)$) to expose improvements in its results and shows that Figures 10f and 11f are similar to the target images. From our investigation, we found that the dilated convolutional concept can make our model have better overview information, so it can capture larger areas of data.

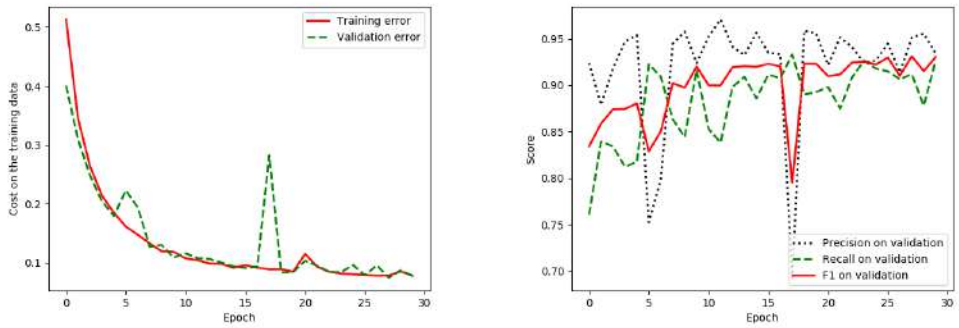
There is a lower discrepancy (peak) in the validation data of "HR-GCN-FF-DA" Figure 12a than that in the baseline Figure 13a. Moreover, Figures 13b and 12b show three learning graphs such as precision, recall, and F1 lines. The loss graph of the "HR-GCN-FF-DA" model seems flatter (very smooth) than the baseline in Figure 13a. The epoch at number 27 was selected to be a pre-trained model for testing and transfer learning procedures.

In the Landsat-8w5c corpus, for an analysis of each class, our final model is clearly the winner in all classes with an accuracy beyond 95% in two classes: agriculture and urban classes. Figures 14 and 15 show twelve sample outputs from our proposed methods (column $(d\ tof)$) compared to the baseline (column $(c)$) to expose improvements in its results and shows that Figures 14f and 15f are similar to the ground images. From our investigation, we found that the dilated convolutional concept can make our model have better overview information, so it can capture larger areas of data.

Considering the loss graphs, our model in Figure 16a can learn smoother than the baseline (our previous work) in Figure 17a, since the discrepancy (peak) in the validation error (green line) is lower in our model. There is a lower discrepancy (peak) in the validation data of "HR-GCN-FF-DA", Figure 16a, than that in the baseline Figure 17a. Moreover, Figures 17b and 16b show three learning graphs such as precision, recall, and F1 lines. The loss graph of the "HR-GCN-FF-DA" model seems flatter (very smooth) than the baseline in Figure 17a. The epoch at number 40 out of 50 was selected to be the pre-trained model for testing and transfer-learning procedures.

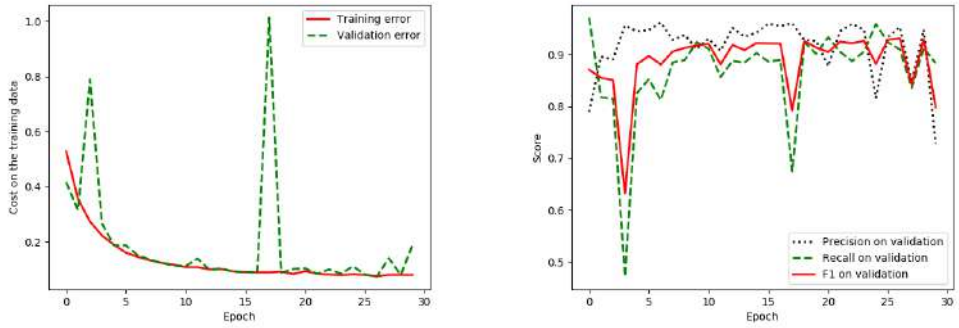
In the ISPRS Vaihingen corpus, for an analysis of each class, our model is clearly the winner in all classes with an accuracy beyond 90% in four classes: impervious surface, building, low vegetation, and trees. Figure 18 shows twelve sample outputs from our proposed methods (column $(d\ tof)$) compared to the baseline (column $(c)$) to expose improvements in its results and shows that Figures 18f and 19f are similar to the target images. From our investigation, we found that the dilated (atrous) convolutional idea can make our deep CNN model have better overview learning, so that it can capture more ubiquitous areas of data.

For the loss graph, it is similar to the results in our previous experiments. There is a lower discrepancy (peak) in the validation data of our model (Figure 20a) than that in the baseline (Figure 21a). Moreover, Figures 21b and 20b explicate a trend that represents a high-grade model performance. Lastly, the epoch at number 26 (out of 30) was selected to be a pre-trained model for testing and transfer learning procedures.




(**a**) (**b**)

**Figure 20.** Graph (learning curves) on ISPRS Vaihingen data set of the proposed approach, "HR-GCN-FF-DA"; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.




(**a**) (**b**)

**Figure 21.** Graph (learning curves) in ISPRS data set of the baseline approach, GCN152-TL-A [12]; x refers to epochs, and y refers to different measures (**a**) Plot of model loss (cross-entropy) on training and validation corpora; (**b**) performance plot on the validation corpus.

## 8. Conclusions

We propose a novel CNN architecture to achieve image labeling on remote-sensed images. Our best-proposed method, "HR-GCN-FF-DA", delivers an excellent performance in regards to three aspects: (i) modifying the backbone architecture with "High-Resolution Representations (HR)", (ii) applying the "Feature Fusion (FF)", and (iii) using the concept of "Depthwise Atrous Convolution (DA)". Each proposed strategy can really improve $F1$-results by 4.82%, 4.08%, and 2.14% by adding HR, FF, and DA modules, consecutively. The FF module can really capture low-level features, resulting in a higher accuracy of river and low-vegetation classes. The DA module can refine the features and provide more coverage areas, resulting in a higher accuracy of pineapple and miscellaneous classes. The results demonstrate that the "HR-GCN-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 corpora, respectively. Moreover, it reaches an accuracy surpassing 90% in almost all classes.

**Author Contributions:** Conceptualization, T.P.; Data curation, K.J., S.L. and P.S.; Formal analysis, T.P.; Investigation, T.P.; Methodology, T.P.; Project administration, T.P.; Resources, T.P.; Software, T.P.; Supervision, T.P. and P.V.; Validation, T.P.; Visualization, T.P.; Writing–original draft, T.P.; Writing–review and editing, T.P. and P.V. All authors have read and agreed to the published version of the manuscript.

**Acknowledgments:** Teerapong Panboonyuen, also known as Kao Panboonyuen appreciates and thanks to the scholarship from the 100th Anniversary Chulalongkorn University Fund for the Doctoral Scholarship and the 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund. Teerapong Panboonyuen greatly acknowledges the Geo-Informatics and Space Technology Development Agency (GISTDA), Thailand, and Kao thanks to the staff from the GISTDA (Thanwarat Anan, Bussakon Satta, and Suwalak Nakya) for providing the remote sensing corpora used in this study.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following acronyms are used in this article:

| | |
|---|---|
| A | Channel Attention |
| BR | Boundary Refinement |
| DA | Depthwise Atrous Convolution |
| DSM | Digital Surface Model |
| FF | Feature Fusion |
| HR | High-Resolution Representations |
| IS | Impervious Surfaces |
| Misc | Miscellaneous |
| NDSM | Normalized Digital Surface Mode |
| LV | Low Vegetation |
| TL | Transfer Learning |

## References


1. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
2. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
3. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
4. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2359–2367.
5. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
6. Pang, Y.; Li, Y.; Shen, J.; Shao, L. Towards bridging semantic gap to improve semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4230–4239.
7. Zhang, Z.; Zhang, X.; Peng, C.; Xue, X.; Sun, J. Exfuse: Enhancing feature fusion for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 269–284.
8. Li, Y.; Zhang, X.; Chen, D. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1091–1100.
9. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018, pp. 3684–3692.
10. Liu, Y.; Fan, B.; Wang, L.; Bai, J.; Xiang, S.; Pan, C. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote. Sens.* **2018**, *145*, 78–95. [CrossRef]
11. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1743–1751.
12. 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. [CrossRef]
13. Xie, M.; Jean, N.; Burke, M.; Lobell, D.; Ermon, S. Transfer learning from deep features for remote sensing and poverty mapping. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.

14. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
15. Liu, J.; Wang, Y.; Qiao, Y. Sparse Deep Transfer Learning for Convolutional Neural Network. In Proceedings of the AAAI, San Francisco, CA, USA, 4–9 February 2017; pp. 2245–2251.
16. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a discriminative feature network for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1857–1866.
17. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
18. International Society for Photogrammetry and Remote Sensing. 2D Semantic Labeling Challenge. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 9 September 2018).
19. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
20. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
21. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
22. Wang, H.; Wang, Y.; Zhang, Q.; Xiang, S.; Pan, C. Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sens.* **2017**, *9*, 446. [CrossRef]
23. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote. Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
24. Yue, K.; Yang, L.; Li, R.; Hu, W.; Zhang, F.; Li, W. TreeUNet: Adaptive Tree convolutional neural networks for subdecimeter aerial image segmentation. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *156*, 1–13. [CrossRef]
25. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote. Sens.* **2020**, *162*, 94–114. [CrossRef]
26. Sun, T.; Chen, Z.; Yang, W.; Wang, Y. Stacked U-Nets With Multi-Output for Road Extraction. In Proceedings of the CVPR Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 202–206.
27. Lateef, F.; Ruichek, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing* **2019**, *338*, 321–348. [CrossRef]
28. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
29. Ghosh, S.; Das, N.; Das, I.; Maulik, U. Understanding deep learning techniques for image segmentation. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–35. [CrossRef]
30. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 405–420.
31. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
33. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1264. [CrossRef]
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
35. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.

36. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October2015; Springer: Berlin, Germany, 2015; pp. 234–241.
37. Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; Wang, J. High-resolution representations for labeling pixels and regions. *arXiv* **2019**, arXiv:1904.04514.
38. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5693–5703.
39. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
40. Kampffmeyer, M.; Jenssen, R.; Salberg, A.B. Dense Dilated Convolutions Merging Network for Semantic Mapping of Remote Sensing Images. In Proceedings of the 2019 Joint Urban Remote Sensing Event (JURSE), Vannes, France, 22–24 May 2019; pp. 1–4.
41. Yang, W.; Wang, W.; Zhang, X.; Sun, S.; Liao, Q. Lightweight feature fusion network for single image super-resolution. *IEEE Signal Process. Lett.* **2019**, *26*, 538–542. [CrossRef]
42. Ma, C.; Mu, X.; Sha, D. Multi-Layers Feature Fusion of Convolutional Neural Network for Scene Classification of Remote Sensing. *IEEE Access* **2019**, *7*, 121685–121694. [CrossRef]
43. Du, Y.; Song, W.; He, Q.; Huang, D.; Liotta, A.; Su, C. Deep learning with multi-scale feature fusion in remote sensing for automatic oceanic eddy detection. *Inf. Fusion* **2019**, *49*, 89–99. [CrossRef]
44. Duarte, D.; Nex, F.; Kerle, N.; Vosselman, G. Multi-resolution feature fusion for image classification of building damages with convolutional neural networks. *Remote Sens.* **2018**, *10*, 1636. [CrossRef]
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
47. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]
48. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the OSDI, Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.


*remote sensing*

MDPI

*Article*

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

**Teerapong Panboonyuen [1,*], Kulsawasd Jitkajornwanich [2], Siam Lawawirojwong [3], Panu Srestasathiern [3] and Peerapon Vateekul [1,*]**

1 Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand

2 Data Science and Computational Intelligence (DSCI) Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd, Ladkrabang, Bangkok 10520, Thailand; kulsawasd.ji@kmitl.ac.th

3 Geo-Informatics and Space Technology Development Agency (Public Organization), 120, The Government Complex, Chaeng Wattana Rd, Lak Si, Bangkok 10210, Thailand; siam@gistda.or.th (S.L.); panu@gistda.or.th (P.S.)

* Correspondence: teerapong.panboonyuen@gmail.com (T.P.); peerapon.v@chula.ac.th (P.V.)

Received: 5 December 2018; Accepted: 1 January 2019; Published: 4 January 2019


**Abstract:** In the remote sensing domain, it is crucial to complete semantic segmentation on the raster images, e.g., river, building, forest, etc., on raster images. A deep convolutional encoder–decoder (DCED) network is the state-of-the-art semantic segmentation method for remotely sensed images. However, the accuracy is still limited, since the network is not designed for remotely sensed images and the training data in this domain is deficient. In this paper, we aim to propose a novel CNN for semantic segmentation particularly for remote sensing corpora with three main contributions. First, we propose applying a recent CNN called a global convolutional network (GCN), since it can capture different resolutions by extracting multi-scale features from different stages of the network. Additionally, we further enhance the network by improving its backbone using larger numbers of layers, which is suitable for medium resolution remotely sensed images. Second, "channel attention" is presented in our network in order to select the most discriminative filters (features). Third, "domain-specific transfer learning" is introduced to alleviate the scarcity issue by utilizing other remotely sensed corpora with different resolutions as pre-trained data. The experiment was then conducted on two given datasets: (i) medium resolution data collected from Landsat-8 satellite and (ii) very high resolution data called the ISPRS Vaihingen Challenge Dataset. The results show that our networks outperformed DCED in terms of $F1$ for 17.48% and 2.49% on medium and very high resolution corpora, respectively.

**Keywords:** deep convolutional neural networks; multi-class segmentation; global convolutional network; channel attention; transfer learning; ISPRS Vaihingen; Landsat-8

## 1. Introduction

Semantic segmentation of earthly objects such as agriculture fields, forests, roads, and urban and water areas from remotely sensed images has been manipulated in many applications in various domains, e.g., urban planning, map updates, route optimization, and navigation [1–5], allowing us to better understand the domain's images and create important real-world applications.

A deep convolutional neural network (CNN) is a well-known method for automatic feature learning. It can mechanically learn features at different levels and abstractions from raw images by multiple hierarchical stacking convolution and pooling layers [4–14]. To accomplish such a challenging task, features at different levels are required. Specifically, abstract high-level features are more suitable for the recognition of confusing manmade objects, while the labeling of finely structured objects could benefit from detailed low-level features [1]. Therefore, different numbers of layers will affect the performance of deep learning models.

In the past few years, the modern CNNs have been extensively proposed including Global Convolutional Network (GCN) [15] in which the large kernel and effective receptive field play an important role in performing classification and localization tasks simultaneously. The GCN is proposed to address the classification and localization issues for semantic segmentation and to suggest a residual-based boundary refinement for further refining object boundaries. However, this type of architecture ignores the global context such as weights of the features in each stage. Furthermore, most methods of this type are just summed up the features of adjacent stages without considering their diverse representations. 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.

In this paper, we present a novel global convolutional network for segmenting multi-objects from aerial and satellite images. To this end, it is focused on three aspects: (i) varying backbones using ResNet50, ResNet101, and ResNet152, (ii) applying a "channel attention block" [16,17] to assign weights for feature maps in each stage of the backbone architecture, and (iii) employing "domain-specific transfer learning" [18–20] to relieve scarcity. Experiments were conducted using satellite imagery (from the Landsat-8 satellite), which was provided by a government organization in Thailand, and using well-known aerial imagery from the ISPRS Vaihingen Challenge corpus [21], which is publicly available. The results showed that our method outperforms the baseline including deep convolutional encoder–decoder (DCED) in terms of $F1$ and by mean of class-wise intersection over union (*mean IoU*).

The remainder of this paper is arranged as follows. The related work is discussed in Section 2. Section 3 describes our proposed methodology. Experimental datasets and evaluations are described in Section 4. Experimental results and discussions are presented in Section 5. Finally, we conclude our work and discuss future work in Section 6.

## 2. Related Work

Deep learning has been successfully applied for remotely sensed data analysis, notably land cover mapping on urban areas [1–3], and has increasingly become a promising tool for accelerating the image recognition process with high accuracy [4–14,22–30]. It is a fast-growing field, and new architectures appear every few days. This section is divided into three subsections: we discuss deep learning concepts for semantic segmentation, a set of multi-objects segmentation techniques using modern deep learning architectures, and modern techniques of deep learning.

### *2.1. Deep Learning Concepts for Semantic Segmentation*

Semantic segmentation algorithms are often formulated to solve structured pixel-wise labeling problems based on a deep CNN. Noh et al. [13] proposed a novel semantic segmentation technique utilizing a deconvolutional neural network (DCNN) and the top layer from the DCNN adopted from

VGG16 [4,8]. The DCNN structure is composed of upsampling layers and deconvolution layers, describing pixel-wise class labels and predicting segmentation masks, respectively. Their proposed deep learning methods yield high performance in PASCAL VOC 2012 corpus, with the 72.5% accuracy in the best-case scenario (the highest accuracy—as of the time of the writing of this paper—compared to other methods that were trained without requiring additional or external data). Long et al. [12] proposed adapted contemporary classification networks incorporating Alex, VGG, and GoogLe networks into a fully CNN. In this method, some of the pooling layers were skipped: Layer 3 (FCN-8s), Layer 4 (FCN-16s), and Layer 5 (FCN-32s). The skip architecture reduces the potential over-fitting problem and has shown improvements in performance, ranging from 20% to 62.2% in the experiments tested on PASCAL VOC 2012 data. Ronneberger et al. [14] proposed U-Net, a DCNN for biomedical image segmentation. The architecture consists of a contracting path and a symmetric expanding path that captures context and consequently enables precise localization. The proposed network claimed to be capable of learning despite the limited number of training images and performed better than the prior best method (a sliding-window DCNN) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Vijay Badrinarayanan [31–33] proposed a deep convolutional encoder–decoder network (DCED), called "SegNet," that consists of two main networks, encoder and decoder, and some outer layers. The two outer layers of the decoder network are responsible for feature extraction, the results of which are transmitted to the layer adjacent to the last layer of the decoder network. This layer is responsible for pixel-wise classification (determining which pixel belongs to which class). There is no fully connected layer in between feature extraction layers. In the upsampling layer of the decoder, pool indices from the encoder are distributed to the decoder, where the kernel will be trained in each epoch (the training round) at the convolution layer. In the last layer (classification), softmax was used as a classifier for pixel-wise classification. The DCED is one of the deep learning models that exceeds the state of the art on many remote sensing corpus.

In this work, the DCED method was selected as our baseline since it is the most popular architecture used in various networks for semantic segmentation.

### *2.2. Modern Deep Learning Architectures For Semantic Segmentation*

Recently, many approaches based on the DCED have achieved high performance on different benchmarks [16,31–33]. However, most of them still suffer from accuracy performance issues. Therefore, many works of modern deep learning architectures have been proposed, such as instance-aware semantic segmentation [34], which is slightly different from semantic segmentation. Instead of labeling all pixels, it focuses on the target objects and labels only pixels of those objects. FCIS [28] is based on techniques based on fully convolutional networks (FCNs). The mask R-CNN [9] was built around the FCN and is incorporated with a proposed joint formulation. Peng [15] presented the concept of large kernel matters to improve semantic segmentation with a global convolutional network (GCN) as shown in Figure 1. They proposed a GCN to address both the classification and localization issues for semantic segmentation. Large separable kernels were used to expand the receptive field, and a boundary refinement block was added to further improve localization performance near the boundaries. From the Cityscapes challenge, the GCN outperforms methods of all previous publications (all modern deep learning baselines) and has become the new state of the art. Therefore, the GCN was selected as our proposed method and as the main model of our work.

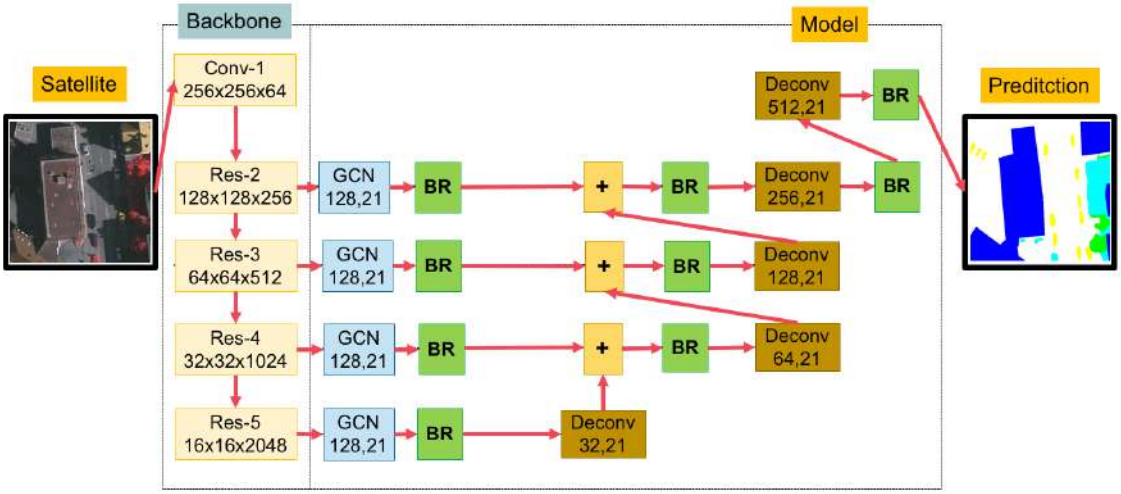



**Figure 1.** An overview of the original global convolutional network (GCN) and boundary refinement (BR) [15].

### *2.3. Modern Techniques of Deep Learning*

Modern techniques of deep learning are important for the accuracy of a CNN. The most popular modern ideas used for semantic segmentation tasks, such as global context, the attention module, and semantic boundary detection, have been used for boosting accuracy.

Global context [16] is a modern method that has proven the effectiveness of global average pooling in the semantic segmentation task. For example, PSPNet [30] and Deeplab v3 [5] respectively extend it to spatial pyramid pooling [30] and atrous spatial pyramid pooling [5], resulting in great performance at different benchmarks. However, to take advantage of the pyramid pooling module sufficiently, these two methods adopt the base feature network to downsample with atrous convolution eight times [5], which is time-consuming and memory-intensive.

Attention Module [16]: Attention is helpful to focus on what we want. Recently, the attention module has increasingly become a powerful tool for deep neural networks [16,17]. The method in [16,17] pays attention to different scale information. In this work, we utilize a channel attention block to select features, similar to learning a discriminative feature network [16].

Refinement Residual Block [16]: The feature maps of each stage in the feature network all go through the refinement residual block. For our work, we use the boundary refinement block (BR) to be a concept of "refinement residual block" from [15]. The first component of the block is a $1 \times 1$ convolution layer. We use it to unify the number of channels to 21. Meanwhile, it can combine the information across all channels. Then the following is a basic residual block [7], which can refine the feature map. Furthermore, this block can strengthen the recognition ability of each stage, inspired from the architecture of ResNet.

## 3. The Proposed Method

In this section, the details of our proposed network are explained (shown in Figure 2). The network is based on the GCN with three aspects of improvements: (i) the modification of backbone architecture (shown in P1 in Figure 2), (ii) applying the channel attention block (shown in P2 in Figure 2), and (iii) using the concept of domain-specific transfer learning (shown in P3 in Figure 2).

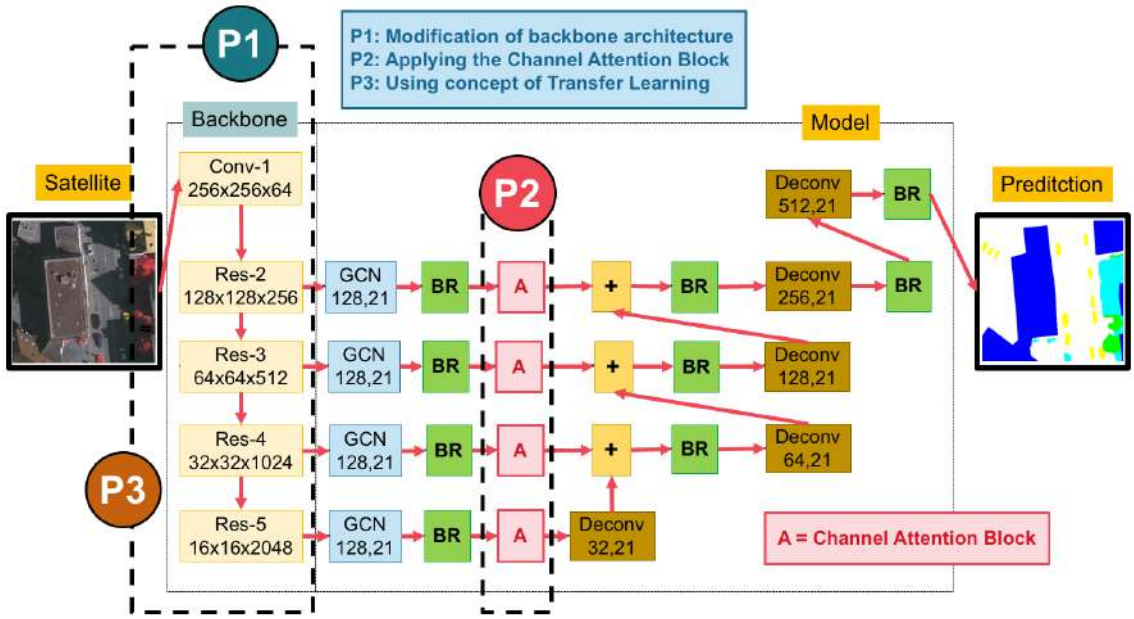



**Figure 2.** An overview of our proposed network.

### *3.1. Data Preprocessing*

In this paper, there are two benchmark corpuses, including (i) the ISPRS Vaihingen Challenge corpus and (ii) the Landsat-8 dataset. They are comprised of very high and medium resolution images, respectively. More details of the datasets will be explained in Sections 4.1 and 4.2. Before a discussion of the model, it is worth explaining our data preprocessing procedure, since it is required when working with neural network and deep learning models. Thus, the mean subtraction is executed.

In addition, data augmentation is often required on more complex object recognition tasks. Therefore, a random horizontal flip is generated to increase the training data. For the ISPRS corpus, all images are standardized and cropped into $512 \times 512$ pixels with a resolution of 9 $cm^2$/pixel. For the Landsat-8 corpus, each image is also flipped horizontally and scaled to $512 \times 512$ with a resolution of 30 $m^2$/pixel from the original images ($16{,}800 \times 15{,}800$ pixels).

### *3.2. A Global Convolutional Network (GCN) with Variations of Backbones*

GCN [15] as shown in Figure 1 is a modern architecture that surpasses the drawbacks of a traditional semantic segmentation network, such as deep convolutional encoder–decoder (DCED) networks. A traditional network usually cascades convolutional layers in order to generate sophisticated features; they can be considered as local features that are specialized for a specific task. However, it is not necessary to employ only specialized features; the general features are also important. Thus, a GCN overcomes this issue by introducing a multi-level architecture, where each level aims to capture a different resolution of features, so both local and global features are considered in the model.

As shown in Figure 1, there are two main blocks in the GCN: a localization block and a classification block. From the localization view in the left block, the structure is a stack of classical fully convolutional layers called "levels." Each level aims to construct features with different resolutions. From the classification view, there are two modules: the GCN and the boundary refinement (BR). For the GCN module, the kernel size of the convolutional structure should be as large as possible, which is motivated by the densely connected structure of the classification models. If the kernel size increases to the spatial size of the feature map (named the global convolution), the network will share the same benefits with the pure classification models. The BR module is added to further improve localization performance near the boundaries.

Although the GCN architecture has shown promising prediction performance, it is still possible to further improve by varying backbones using ResNet [7] with different numbers of layers as ResNet50, ResNet101, and ResNet152, as shown in Figure 3. Additionally, the GCN is suggested to work on a large kernel size. In this paper, we set the large kernel size as 9 (this previous work [15]).

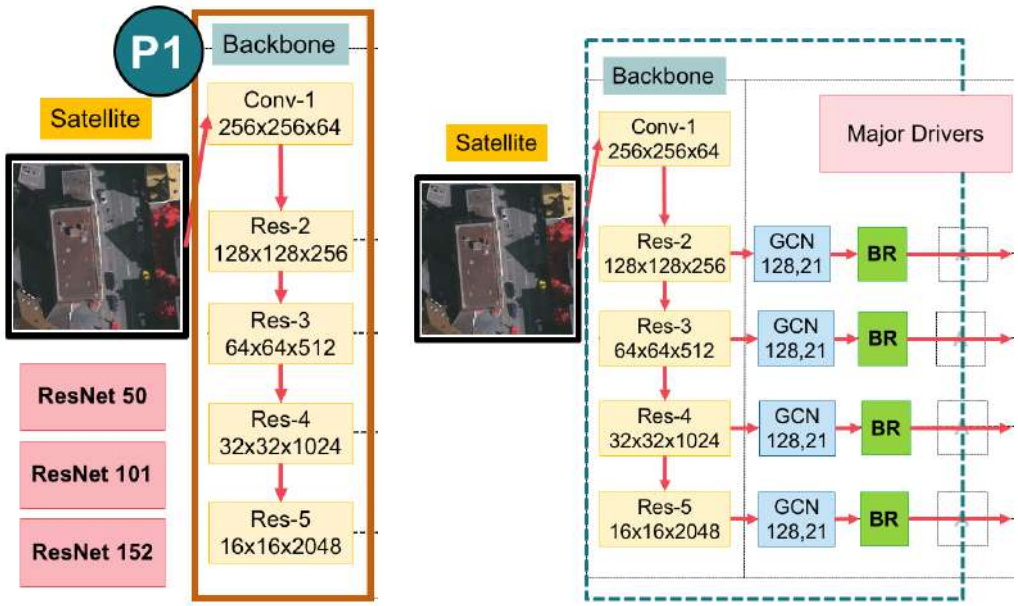



**Figure 3.** An overview of the whole backbone pipeline in (**left**) the main backbone with varying by ResNet50, ResNet101, and ResNet152; (**right**) the major drivers of our main classification network (composed of a global convolutional network (GCN) and a boundary refinement (BR) block [15]).

### *3.3. The Channel Attention Block*

Attention mechanisms [16,17] in neural networks are very loosely based on the visual attention mechanism found in humans and equips a neural network with the ability to focus on a subset of its inputs (or features): it selects specific inputs. Human visual attention is well-studied, and while there are different models, all of them essentially come down to being able to focus on a certain region of an image with a very high resolution, perceiving the surrounding image in a medium resolution, and then adjusting the focal point over time.

To apply this atttentional layer to our network, the channel attention block is shown in Block A in Figure 2 and its detailed architecture is shown in Figure 4. It is designed to change the weights of the remote sensing features on each stage (level), so that the weights are assigned more values on important features adaptively.

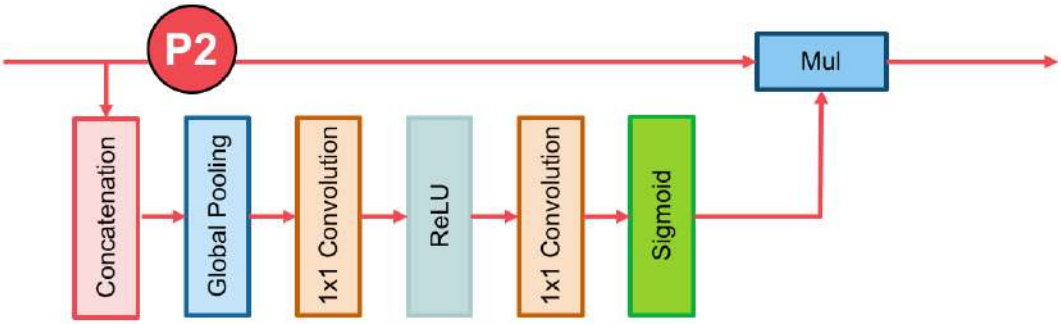



**Figure 4.** Components of the channel attention block. The red lines represent the downsample operators, respectively. The red line cannot change the size of feature maps. It is only a path for information passing.

In the proposed architecture, a convolution operator gives the probability of each class at each pixel. In Equation (1), the final score is summed over all channels of the feature maps.

$$y_k = F(x; w) = \sum_{i=1,j=1}^{D} w_{i,j} x_{i,j} \tag{1}$$

where $x$ is the output feature of network. $w$ represents the convolution's kernel, and $k \in 1, 2, 3, 4, 5, 6, 7, \ldots, K$. The number of channels is represented by $K$, and $D$ is the set of pixel positions.

$$\delta_i(y_k) = \frac{exp(y_k)}{\sum_{j=1}^{k} exp(y_j)} \tag{2}$$

where $\delta$ is the prediction probability. $y$ is the output of the network. As shown in Equations (1) and (2), the final predicted label is the category with the highest probability. Therefore, we suppose that the prediction result is $y_0$ of a certain patch, while its true label is $y_1$. Therefore, we can introduce a parameter $\alpha$ to change the highest probability value from $y_0$ to $y_1$, as Equation (3) shows.

$$\bar{y} = \alpha y = \begin{bmatrix} \alpha_1 \\ . \\ . \\ . \\ \alpha_k \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_k \end{bmatrix} = \begin{bmatrix} \alpha_1 w_1 \\ . \\ . \\ . \\ \alpha_k w_k \end{bmatrix} \times \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_k \end{bmatrix} \tag{3}$$

where $\bar{y}$ is the new prediction of the network, and $\alpha = Sigmoid(x; w)$.

Based on the above formulation of the Channel Attention Block, we can explore its practical significance. In Equation (1), it implicitly indicates that the weights of different channels are equal. However, the features in different stages have different degrees of discrimination, which results in different consistency of prediction. Consequently, in Equation (3), the $\alpha$ value applies the feature maps $x$, which represents the feature selection with the channel attention block.

### *3.4. Domain-Specific Transfer Learning*

The overall idea of transfer learning is to use knowledge learned from tasks for which many labeled data are usable in settings where only little-labeled data are available. Creating labeled data is expensive, so optimally leveraging an existing dataset is key. Certain low-level features, such as edges, shapes, corners, and intensity, can be shared across tasks, and new high-level features specific to the target problem can be learned [18]. Additionally, knowledge from an existing task acts as an additional input when learning a new target task.

Although the deep learning approach often performs promising prediction performance, it requires a large amount of training data. Since it is difficult to obtain annotated satellite images, the performance in prior works has been limited.

Fortunately, there is a recent concept called domain-specific transfer learning [18–20] that allows one to reuse the weights obtained from other domains' inputs. It is currently very popular in the field of deep learning because it enables one to train deep neural networks with comparatively insufficient data. This is very useful since most real-world problems typically do not have millions of labeled data points to train such complex models.

In terms of inadequacy, we propose an effective transfer deep neural network to perform knowledge transfer between a very high resolution (VHR) corpus and a medium resolution (MR) corpus. It is shown in Figure 5.

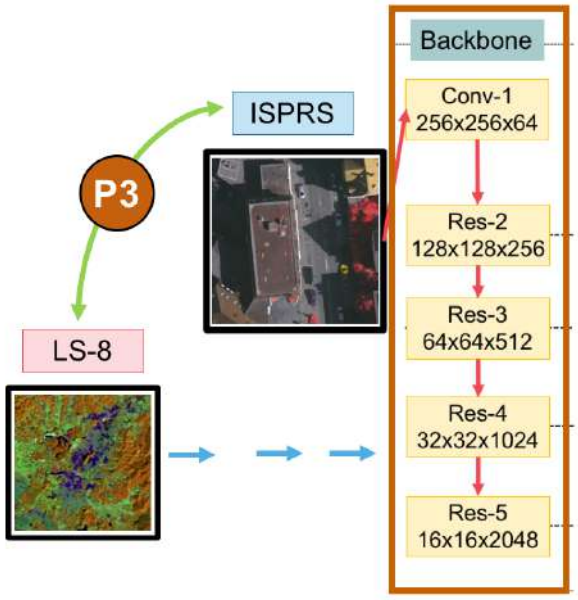



**Figure 5.** The domain-specific transfer learning strategy reuses pre-trained weights of models between two datasets—very high (ISPRS) and medium (Landsat-8; LS-8) resolution images.

## 4. Experimental Datasets and Evaluation

In our experiments, two types of datasets were used: (i) medium resolution imagery (satellite images; Landsat-8 dataset) made by the government organization in Thailand, named GISTDA (Geo-Informatics and Space Technology Development Agency (Public Organization)), and (ii) very high resolution imagery (aerial images; ISPRS Vaihingen dataset). All experiments were evaluated based on major metrics, such as *average accuracy*, *F*1 score, and *mean IoU* score.

### *4.1. Landsat-8 Dataset*

Landsat-8 is an American earth observation satellite and it collects and archive medium resolution (30-m spatial resolution) multispectral image data affording seasonal coverage of the global landmasses for a period of no less than 5 years. Landsat-8 [35] images consist of nine spectral bands with a spatial resolution of 30 m for Bands 1–7 and 9. The ultra blue Band 1 is useful for coastal and aerosol studies. Band 9 is useful for cirrus cloud detection. The resolution for Band 8 (panchromatic) is 15 m. Thermal Bands 10 and 11 are useful in providing more accurate surface temperatures and are collected at 100 m. The approximate scene size is 170 km north–south by 183 km east–west (106 mi by 114 mi). Since Landsat-8 data includes additional bands, the combinations used to create RGB composites differ from Landsat 7 and Landsat 5. For instance, Bands 4, 3, and 2 are used to create a color infrared (CIR) image using Landsat 7 or Landsat 5. To create a CIR composite using Landsat 8 data, Bands 5, 4, and 3 are used.

In this type of data, the satellite images are from Nan, a province in Thailand. The dataset is obtained from Landsat-8 satellite consisting of 1012 satellite images as shown by some samples in Figure 6.

This corpus is comprised of a large, diverse set of medium resolution (16,800 × 15,800) pixels, where 1012 of these images have high quality pixel-level labels of five classes: agriculture, forest, miscellaneous, urban, and water. The 1012 images were split into 800 training and 112 validation images with publicly available annotation, as well as 100 test images with annotations withheld, and comparison to other methods were performed via a dedicated evaluation server. For quantitative evaluation, mean of class-wise intersection over union (*mean IoU*) and *F*1 score are used.

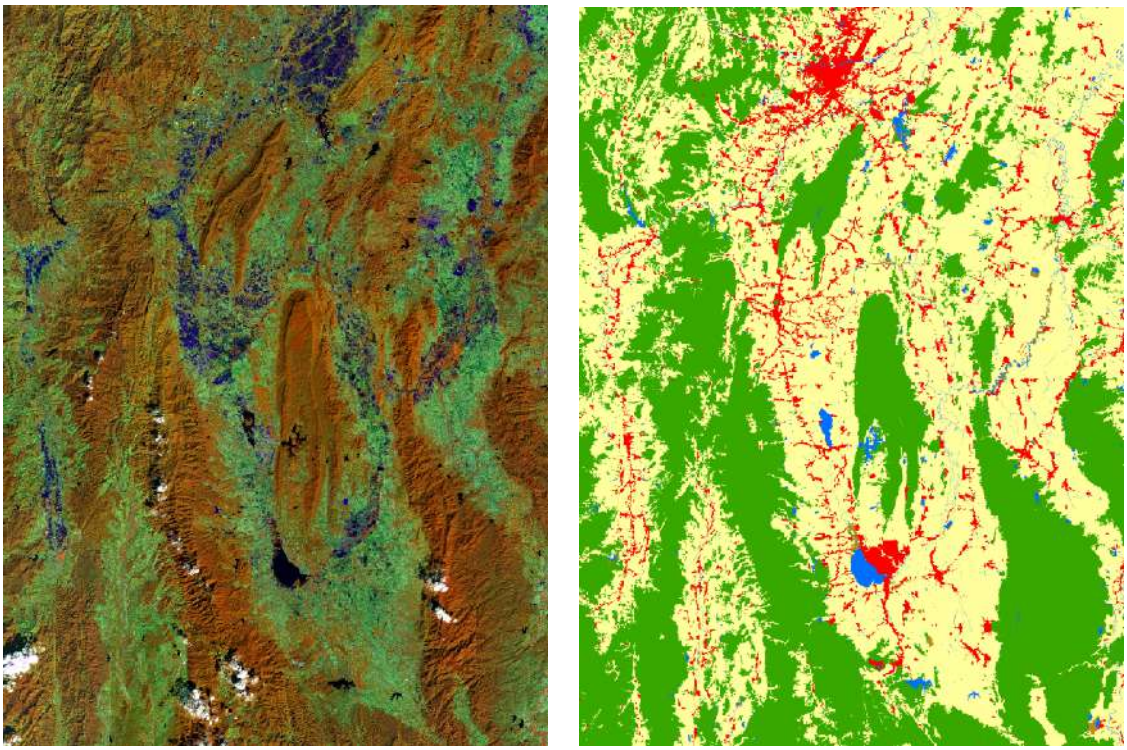


**Figure 6.** Sample satellite images from Nan, a province in Thailand (**left**), and corresponding ground truth (**right**). The label of medium resolution dataset includes five categories: agriculture (yellow), forest (green), miscellaneous (brown), urban (red), and water (blue).

### *4.2. ISPRS Vaihingen Dataset*

One of the major challenges in remote sensing is the automated extraction of urban objects from data acquired by airborne sensors. The Semantic Labeling Contest provides two state-of-the-art airborne image corpora. The Vaihingen corpus shows a relatively small village with many detached buildings and small multi-story buildings, and the Potsdam corpus shows a typical historic city with large building blocks, narrow streets, and dense settlement structure. In our experiments, the Vaihingen corpus was selected and used.

The ISPRS 2D Semantic labeling challenge in Vaihingen [21] (Figures 7 and 8) was used as our benchmark dataset. It consists of three spectral bands (i.e., red, green, and near-infrared bands), the corresponding DSM (digital surface model) and the NDSM (normalized digital surface model) data. Overall, there are 33 images of about 2500 $\times$ 2000 pixels at a ground sampling distance (GSD) of about 9 cm in the image data. Among them, the ground truth of only 16 images are available, and those of the remaining 17 images are withheld by the challenge organizer for the online test. For offline validation, we randomly split the 16 images with ground truth available into a training set of 10 images and a validation set of 6 images. For this work, DSM and NDSM data in all experiments on this dataset were not used. Following other methods, four tiles (Image Numbers 5, 7, 23, and 30) were removed from the training set as the validation set. Experimental results are reported on the validation set if not specified.


**Figure 7.** Overview of the ISPRS 2D Vaihingen Labeling corpus. There are 33 tiles. Numbers in the figure refer to the individual tile flag.

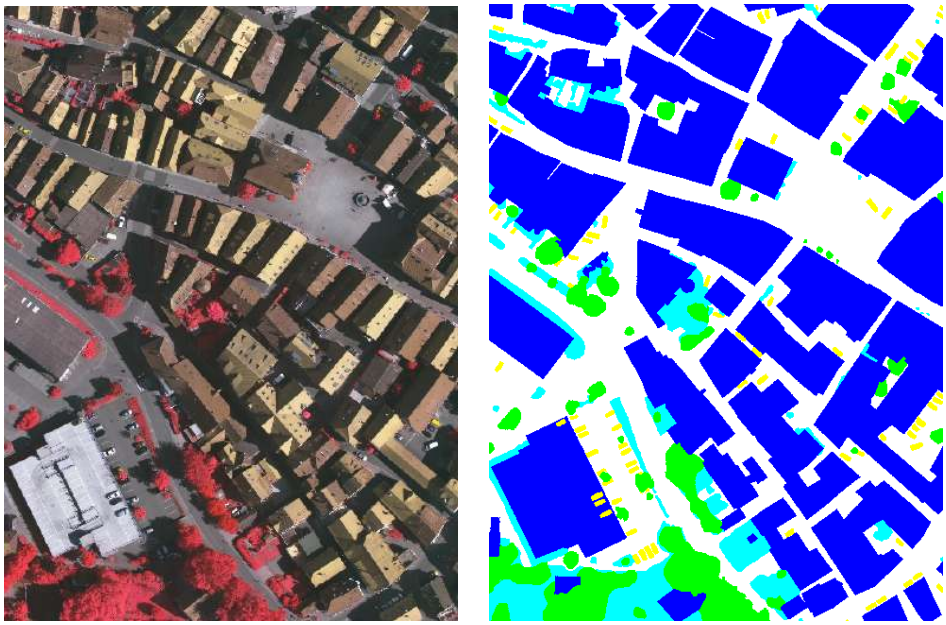


**Figure 8.** The sample input tile from Figure 7 (**left**) and corresponding ground truth (**right**). The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow), and clutter/background (red).

### *4.3. Evaluation*

The multi-class classification task can be considered as multi-segmentation, where class pixels are positives and the remaining non-spotlight pixels are negatives. Let $TP$ denote the number of true positives, $TN$ denote the number of true negatives, $FP$ denote the number of false positives, and $FN$ denote the number of false negatives.

*Precision*, *recall*, $F1$, and *mean IoU* are shown in Equations (4)–(8). Precision is the percentage of correctly classified main pixels among all predicted pixels by the classifier. Recall is the percentage of correctly classified main pixels among all actual main pixels. $F1$ is a combination of *precision* and *recall*.

To evaluate the performance of different deep models, we will discuss the above two major metrics ($F1$), the mean of class-wise intersection over union (*mean IoU*)) on each category, and the mean value of metrics to assess the average performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{7}$$

$$Mean\ IoU = \frac{TP}{TP + FP + FN}. \tag{8}$$

## 5. Experimental Results and Discussion

The implementation is based on a deep learning framework, called "Tensorflow-Slim" [36], which is extended from Tensorflow. All experiments were conducted on servers with an Intel® Xeon® Processor E5-2660 v3 (25M Cache, 2.60 GHz), 32 GB of memory (RAM), an Nvidia GeForce GTX 1070 (8 GB), an Nvidia GeForce GTX 1080 (8 GB), and an Nvidia GeForce GTX 1080 Ti (11 GB). Instead of using the whole image (1500 × 1500 pixels) to train the network, we randomly cropped all images to be 512 × 512 as inputs of each epoch.

For training, the Adam optimizer [11] was chosen with an initial learning rate of 0.004 and the weight decay of 0.00001. Batch normalization [10] is used before each convolutional layer in our implementation to ease the training and make it be able to concatenate feature maps from different layers. To avoid overfitting, common data augmentations are used as details in Section 3.1. For measurements, we use the mean pixel intersection-over-union (*mean IoU*) and the *F*1 score as the metric.

Inspired by [16,27,37], we use the "poly" learning rate policy where the learning rate is multiplied by Equation (9) with a power of 0.9 and an initial learning rate as $4 \times 10^{-3}$. The learning rate is scheduled by multiplying the initial as seen in Equation (9).

$$\text{learning rate} = (1 - \frac{epoch}{MaxEpoch})^{0.9}. \tag{9}$$

All models are trained for 50 epochs with a mini-batch size of 4, and each batch contains the cropped images that are randomly selected from training patches. These patches are resized to 521 × 521 pixels. The statistics of BN is updated on the whole mini-batch.

This section illustrates the details of our experiments. The proposed deep learning network is based on the GCN with three improvements: (i) varying the backbones using ResNet, (ii) channel attention and global average pooling, and (iii) domain-specific transfer learning. From all proposed strategies, there are six acronyms of strategies as shown in Table 1.

**Table 1.** Abbreviations on our proposed deep learning methods.

| 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 |

For the experimental setup, there were three experiments on two remotely sensed datasets: the Landsat-8 dataset and the ISPRS Vaihingen Challenge dataset (details in Sections 4.1 and 4.2). The experiments aimed to illustrate that each proposed strategy can improve the performance. First, the GCN152 method was compared to the GCN50 method and the GCN101 method for the varying backbones using ResNet with different numbers of layers on the GCN network strategy. Second, the GCN152-A method was compared to the GCN152 method for the channel attention strategy. Third, the full proposed technique GCN152-TL-A method was compared to existing methods for the concept of domain-specific transfer learning.

### *5.1. Results of the Landsat-8 Corpus with Discussion*

An experiment was conducted on the Landsat-8 corpus, and the result is shown in Tables 2 and 3 by comparing between baseline and variations of the proposed techniques. It is shown that our network with all strategies, GCN152-TL-A, outperforms other methods. More details will be discussed to show that each of the proposed techniques can improve accuracy. Only in this experiment is there a state-of-the-art baseline, including a deep convolutional encoder–decoder (DCED) [31–33].

**Table 2.** Results of the testing data of the Landsat-8 corpus between baseline and five variations of our proposed techniques in terms of *precision*, *recall*, *F*1, and *mean IoU*.

| | Pretrained | Backbone | Model | *Precision* | *Recall* | *F*1 | *mean IoU* |
|---|---|---|---|---|---|---|---|
| **Baseline** | - | - | DCED [31–33] | 0.6137 | 0.7209 | 0.6495 | 0.5384 |
| **Proposed Method** | - | Res50 | GCN [15] | 0.6678 | 0.7333 | 0.6847 | 0.5734 |
| | - | **Res101** | GCN | 0.6899 | 0.8031 | 0.7290 | 0.6154 |
| | - | **Res152** | GCN | 0.7115 | 0.8131 | 0.7563 | 0.6364 |
| | - | **Res152** | **GCN-A** | 0.7997 | 0.7937 | 0.7897 | 0.6726 |
| | **TL** | **Res152** | **GCN-A** | **0.8293** | **0.8476** | **0.8275** | **0.7178** |

**Table 3.** Results of the testing data of Landsat-8 corpus between each class with our proposed techniques in terms of *averageaccuracy*.

| | Model | Agriculture | Forest | Misc | Urban | Water |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [31–33] | 0.9616 | 0.7472 | 0.0976 | 0.7878 | 0.4742 |
| **Proposed Method** | GCN50 [15] | 0.9407 | 0.8258 | 0.1470 | **0.8828** | 0.5426 |
| | GCN101 | 0.9677 | 0.8806 | 0.2561 | 0.7971 | 0.5480 |
| | GCN152 | 0.9780 | 0.8444 | 0.4256 | 0.7158 | 0.5937 |
| | **GCN152-A** | 0.9502 | **0.9118** | 0.6689 | 0.8675 | 0.6001 |
| | **GCN152-TL-A** | **0.9781** | 0.8472 | **0.8732** | 0.7988 | **0.6493** |

#### 5.1.1. The Effect of an Enhanced GCN on the Landsat-8 Corpus

Our first strategy aims to increase an *F*1 and *mean IoU* score of the network by varying backbones using ResNet 50, ResNet 101, and ResNet 152 rather than the traditional one, the DCED method. From Tables 2 and 3, the *F*1 of GCN152 (0.7563) outperforms that of GCN50 (0.6847), GCN101 (0.7290), and the baseline method, DCED (0.6495); this yields a higher *F*1 at 2.74%, 3.52%, and 4.43%, respectively. The *mean IoU* of GCN152 (0.6364) outperforms that of GCN50 (0.5734), GCN101 (0.6154), and the baseline method, DCED (0.5384); this yields a higher *mean IoU* at 2.10%, 3.50%, and 4.20%, consecutively. The main reason is due to higher precision, but a slightly lower recall. This can imply that enhanced GCN is more significantly efficient than the DCED method (baseline) for this medium resolution corpus and ResNet with a large number of layers is more robust than the small number of layers.

When comparing the results between the original GCN method and the enhanced GCN methods on the Landsat-8 corpus (Table 2), it is clearly shown that a GCN with a larger layer of backbone can improve network performance in terms of *F*1 and *mean IoU*.

#### 5.1.2. The Effect of Using Channel Attention on the Landsat-8 Corpus

Our second mechanism focused on applying the channel attention block (details in Section 3.4) to change the weights of the features on each stage to enhance consistency. In Tables 2 and 3, the $F1$ of GCN152-A (0.7897) is greater than that of GCN152 (0.7563); this yields a higher $F1$ score at 3.34%. The *mean IoU* of GCN152-A (0.6726) is superior to that of GCN152 (0.6364); this yields a higher *mean IoU* score at 3.62%. The result (Figures 9e and 10e) shows that can make the network to obtain discriminative features stage-wise to make the prediction intra-class consistent. This is based on the consideration that we re-weighted all feature maps of each layer.

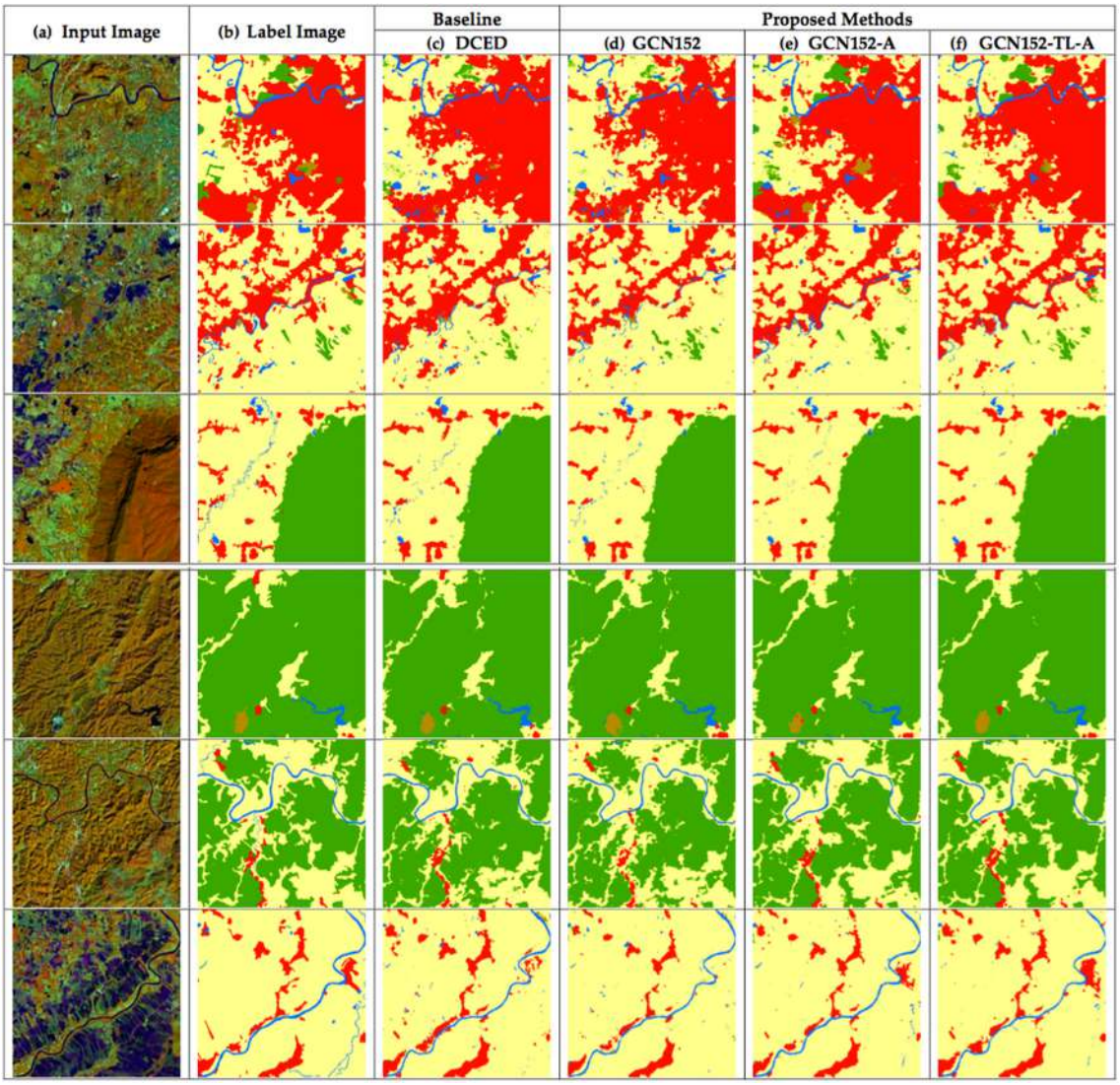



**Figure 9.** Six testing sample inputs and output satellite images on Landsat-8 in the Nan province in Thailand, where rows refer to different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of medium resolution dataset includes five categories: Agriculture (yellow), Forest (green), Miscellaneous (Misc, brown), Urban (red) and Water (blue).

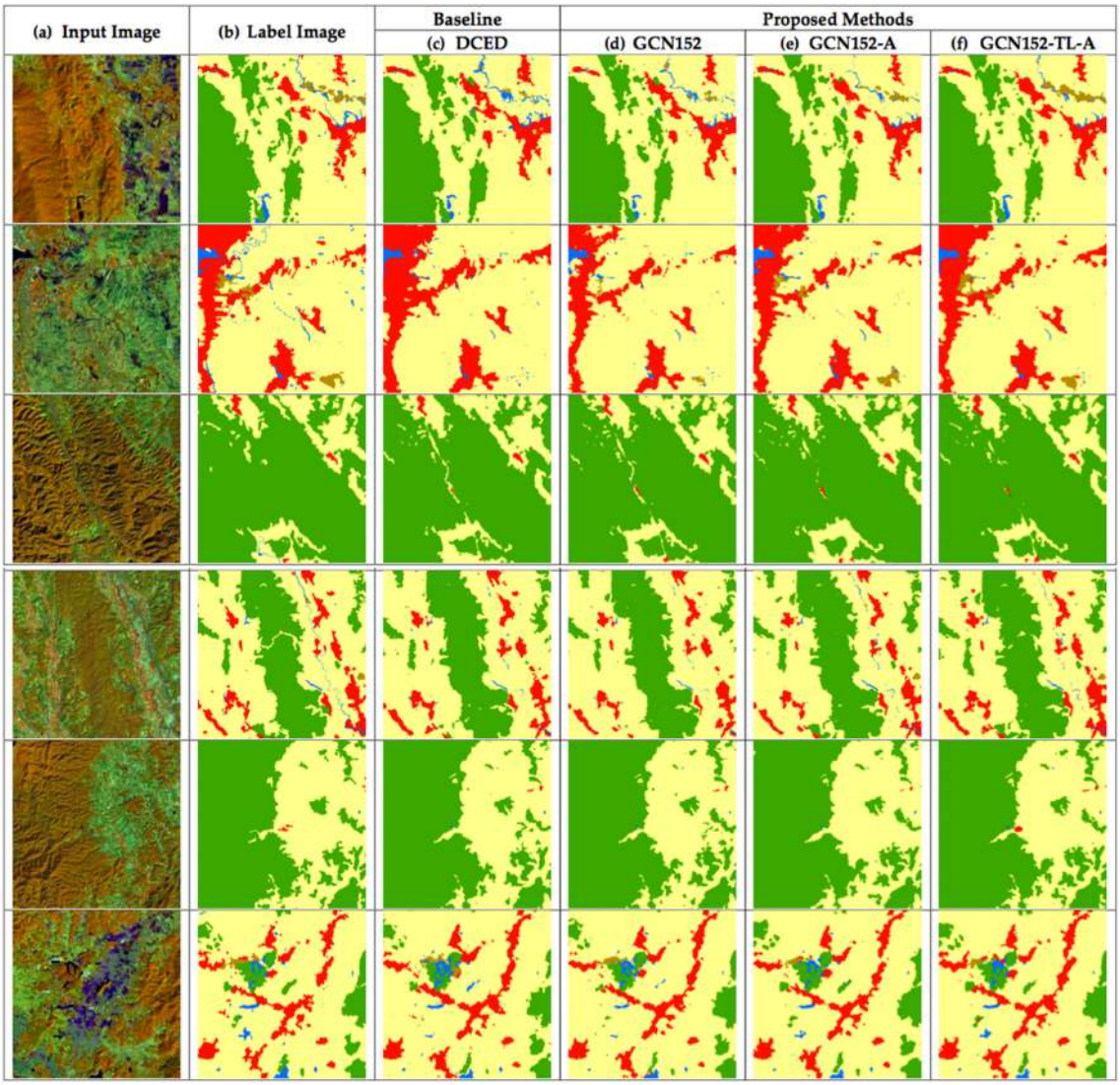



**Figure 10.** Six testing sample input and output satellite images on Landsat-8 in Nan in Thailand, where rows refer to different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of medium resolution dataset includes five categories: Agriculture (yellow), Forest (green), Miscellaneous (Misc, brown), Urban (red) and Water (blue).

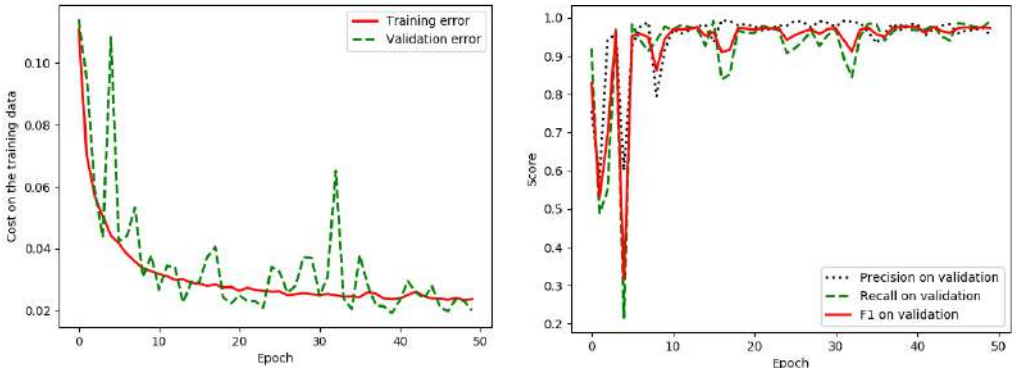
#### 5.1.3. The Effect of Using Domain-Specific Transfer Learning on Landsat-8 Corpus

Our last strategy aims to use approach of domain-specific transfer learning (details in Section 3.3) by reusing the pre-trained weight from the GCN152-A model on the ISPRS Vaihingen corpus. From Tables 2 and 3, the *F*1 of the GCN152-TL-A method is the winner; it clearly outperforms not only the baseline but also all previous generations. Its *F*1 is higher than that of the DCED (baseline) at 17.80%. Its *mean IoU* is higher than that of the DCED at 17.94%. Additionally, the result illustrates that the concept of domain-specific transfer learning can enhance both precision (0.8293) and recall (0.8476).

Figures 9 and 10 show 12 sample results from the proposed method. By applying all strategies, the images in the last column (Figures 9f and 10f) are similar to the ground truths (Figures 9b and 10b). Furthermore, *F*1-results and *mean IoU* scores are improved for each strategy we added to the network as shown in Figures 9c–f and 10c–f.

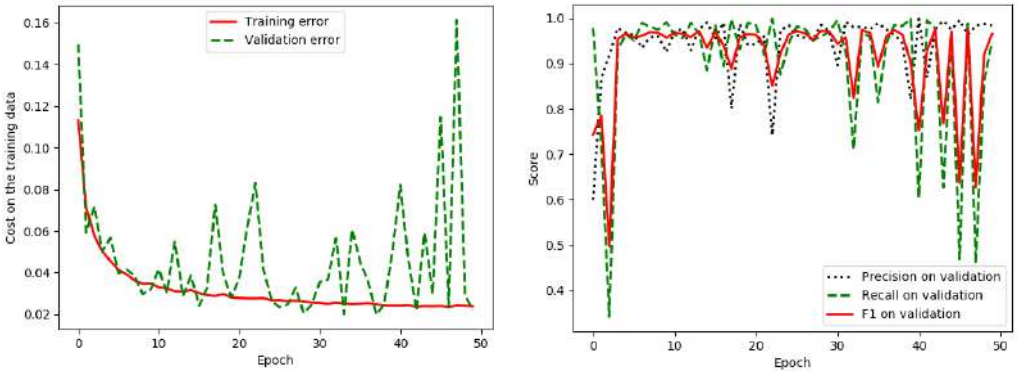
To achieve the highest accuracy, the network must be configured and many epochs must be trained until all parameters in the network are converged. Figure 11a illustrates that the proposed network has been properly set and trained until it is converged and runs more smoothly than the

baseline in Figure 12a. Furthermore, Figures 11b and 12b show that a higher number of epochs tend to show a better $F1$ score. Thus, the number of chosen epochs based on the validation data is 49 (the best model for this dataset).




(**a**) (**b**)

**Figure 11.** Iteration plot on Landsat-8 corpus of the proposed technique, GCN152-TL-A; $x$ refers to epochs and $y$ refers to different measures (**a**) Plot of model loss (cross entropy) on training and validation datasets; (**b**) performance plot on the validation dataset.




(**a**) (**b**)

**Figure 12.** Iteration plot on the Landsat-8 corpus of the baseline technique, the DCED [31–33]; $x$ refers to epochs and $y$ refers to different measures. (**a**) The plot of model loss (cross entropy) on training and validation datasets; (**b**) the performance plot on the validation dataset.

Twelve sample testing results (shown as Figures 9 and 10) are based on the proposed method with respect to Nan (one of the northern provinces (changwat) of Thailand and where agriculture is the main industry). The results of the last column look closest to the ground truth in the second column.

As can be seen in Figures 9 and 10, the performance of our best model outperforms other advanced models by a considerable margin on each category, especially for the agriculture, miscellaneous (Misc), and water classes. Furthermore, the loss curves shown in Figure 11a exhibit that our model performs better on all given categories.

### *5.2. Results of the ISPRS Vaihingen Challenge Corpus with Discussion*

An experiment was conducted on the ISPRS Vaihingen Challenge corpus, and the result is shown in Tables 4 and 5 by comparing between baseline and variations of the proposed techniques. This shows that our network with all strategies (GCN152-TL-A) outperforms other methods. More details will be discussed to show that each of the proposed techniques can improve accuracy. Only in this experiment is there one baseline, which is the DCED network.

**Table 4.** Results of the testing data of the ISPRS 2D semantic labeling challenge corpus between the baseline and five variations of our proposed techniques in terms of *precision*, *recall*, *F*1, and *mean IoU*.

| | Pretrained | Backbone | Model | *Precision* | *Recall* | *F*1 | *mean IoU* |
|---|---|---|---|---|---|---|---|
| **Baseline** | - | - | DCED [31–33] | 0.7519 | 0.7925 | 0.7693 | 0.8651 |
| **Proposed Method** | - | Res50 | GCN [15] | 0.7636 | 0.7917 | 0.776 | 0.8776 |
| | - | **Res101** | GCN | 0.7713 | **0.8059** | 0.7862 | 0.8972 |
| | - | **Res152** | GCN | 0.7736 | 0.8021 | 0.7864 | 0.8977 |
| | - | **Res152** | **GCN-A** | 0.7847 | 0.7961 | 0.7902 | 0.9057 |
| | **TL** | **Res152** | **GCN-A** | **0.7888** | 0.8001 | **0.7942** | **0.9123** |

**Table 5.** Results of the testing data of ISPRS Vaihingen Challenge corpus between each class with our proposed techniques in terms of *AverageAccuracy*.

| | Model | IS | Buildings | LV | Tree | Car |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [31–33] | 0.9590 | 0.9778 | 0.9108 | 0.9805 | 0.6832 |
| **Proposed Method** | GCN50 [15] | 0.9595 | 0.9628 | 0.9403 | 0.9896 | 0.7292 |
| | GCN101 | 0.9652 | 0.9827 | **0.9615** | 0.9797 | 0.7387 |
| | GCN152 | 0.9543 | **0.9962** | 0.9445 | 0.9754 | 0.7710 |
| | **GCN152-A** | 0.9614 | 0.9865 | 0.9554 | 0.9871 | 0.8181 |
| | **GCN152-TL-A** | **0.9664** | 0.9700 | 0.9499 | **0.9901** | **0.8567** |

#### 5.2.1. Effect of the Enhanced GCN on the ISPRS Vaihingen Corpus

Our first strategy aims to increase the *F*1 and *mean IoU* score of the network by varying backbones using ResNet 50, ResNet 101, and ResNet 152 rather than the traditional one, the DCED method. From Tables 4 and 5, the *F*1 of GCN152 (0.7864) outperforms that of GCN50 (0.776), GCN101 (0.768), and the baseline method, DCED (0.7693); this yields a higher *F*1 at 0.02%, 0.68%, and 1.01%, respectively. The *mean IoU* of GCN152 (0.8977) outperforms that of GCN50 (0.8776), GCN101 (0.8972), and the baseline method, DCED (0.8651); this yields a higher *mean IoU* at 0.02%, 0.68%, and 1.01% respectively. This can imply that an enhanced GCN is also more accurate than the DCED approach on a very high resolution dataset. ResNet with a large number of layers is still more robust than a small number of layers, the same as that performed on the Landsat-8 corpus (Section 5.1.1).

When comparing the results between the original GCN method and the enhanced GCN methods on the Landsat-8 corpus (Table 4), it is clear that the GCN with a larger backbone layer can improve network performance in terms of *F*1 and *mean IoU*

#### 5.2.2. Effect of Using Channel Attention on ISPRS Vaihingen Corpus

Our second mechanism focused on utilizing the channel attention block to change the weights of the features on each stage to enhance the consistency. From Tables 4 and 5, the *F*1 of GCN152-A (0.7902) is greater than that of GCN152 (0.7864); this yields a higher *F*1 score at 0.38%. The *mean IoU* of GCN152-A (0.9057) is better than that of GCN152 (0.8977); this yields a higher *mean IoU* score at 0.80%. The results (Figures 13e and 14e) show that this can also cause the network to obtain discriminative features stage-wise to make intra-class prediction consistent with respect to very high resolution images.

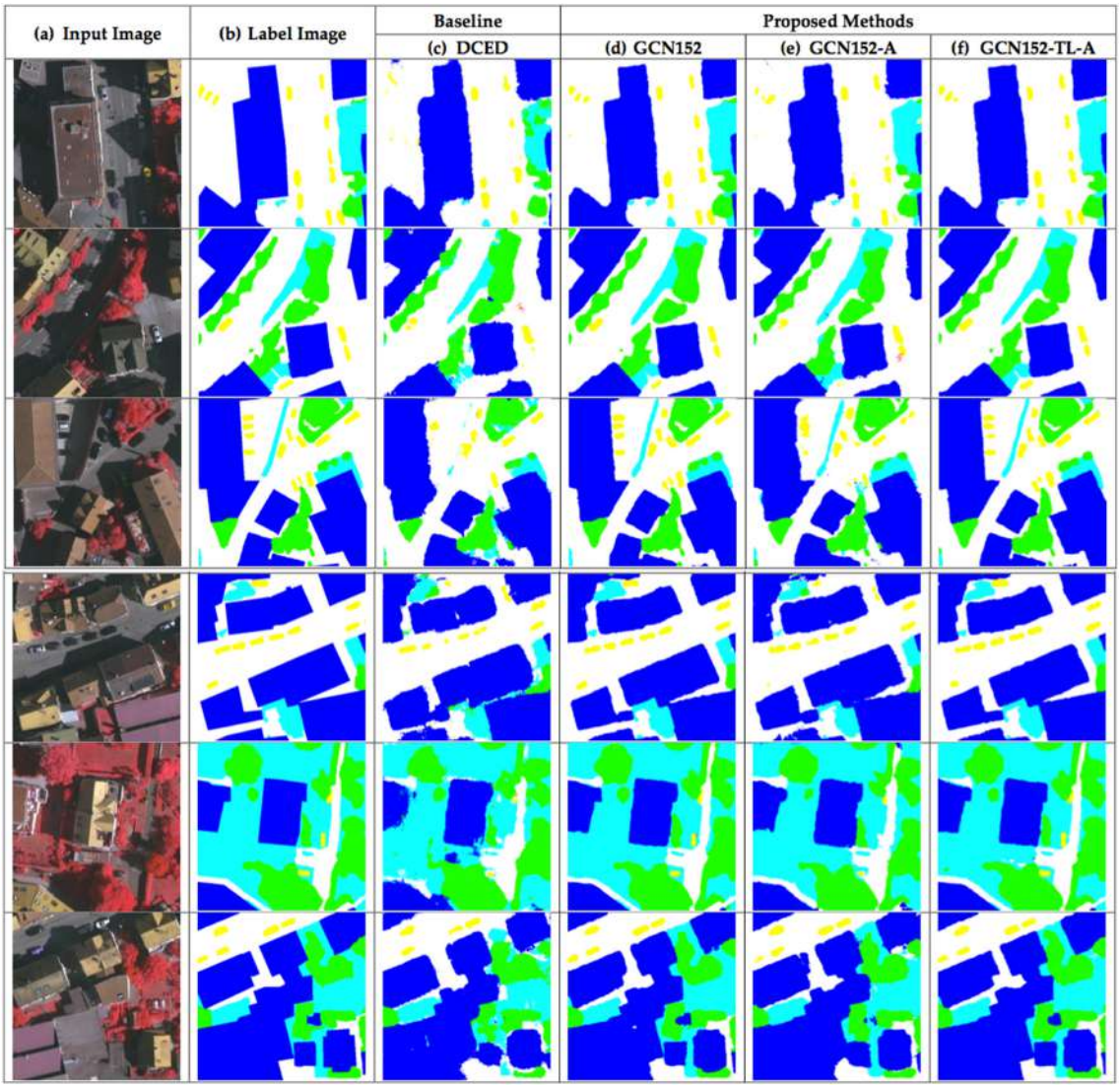



**Figure 13.** Six testing sample input and output aerial images on ISPRS Vaihingen Challenge corpus, where rows refer different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow) and clutter/background (red).

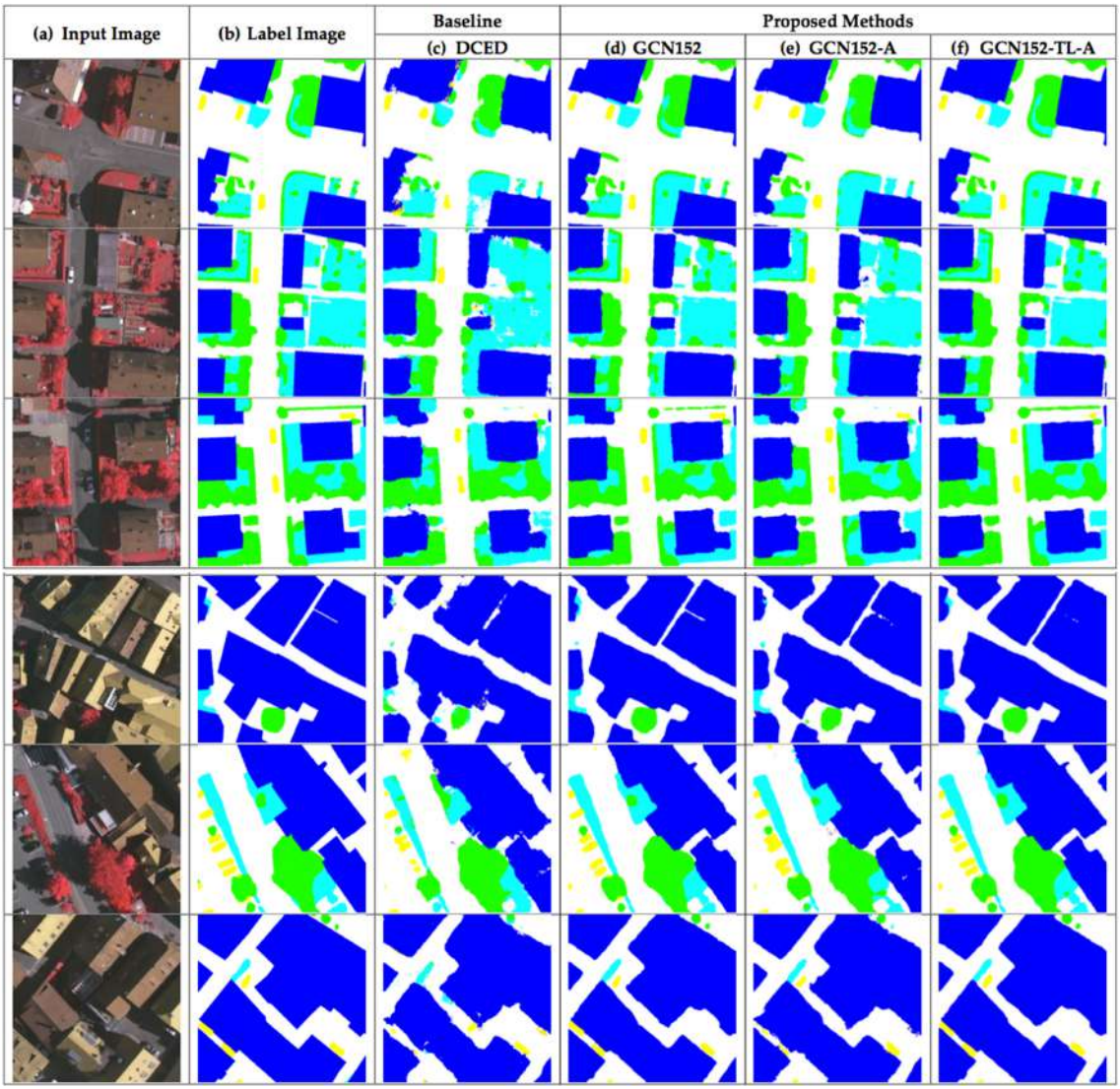



**Figure 14.** Six testing sample input and output aerial images on ISPRS Vaihingen Challenge corpus, where rows refer different images. (**a**) Original input image; (**b**) Target map (ground truth); (**c**) Output of Encoder–Decoder (Baseline); (**d**) Output of GCN152; (**e**) Output of GCN152-A; and (**f**) Output of GCN152-TL-A. The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow), and clutter/background (red).

#### 5.2.3. The Effect of Using Domain-Specific Transfer Learning on the ISPRS Vaihingen Corpus

Our last strategy aims to perform domain-specific transfer learning (details in Section 3.3) by reusing the pre-trained weight from the GCN152-A model on the Landsat-8 corpus. From Tables 4 and 5, the $F1$ of the GCN152-TL-A method is the winner; it clearly outperforms not only the baseline but also all previous generations. Its $F1$ is higher than the DCED (baseline) at 2.49% and 1.82% consecutively. Its *mean IoU* is higher than the DCED and the GCN at 4.76% and 3.51%, respectively. Additionally, the result illustrates that the concept of domain-specific transfer learning can enhance both precision (0.7888) and recall (0.8001).

Figures 13 and 14 shows 12 sample results from the proposed method. By applying all strategies, the images in the last column (Figures 13f and 14f) are similar to ground truths (Figures 13b and 14b). Furthermore, $F1$ results and *mean IoU* scores are improved for each strategy we added to the network as shown in Figures 13c–f and 14c–f.

To further evaluate the effectiveness of the proposed GCN152-TL-A comparisons with the baseline method on the one challenging benchmark and the one private benchmark are presented in Tables 2 and 3 for the Landsat-8 dataset with respect to Nan (Thailand) and Tables 4 and 5 for the Vaihengen dataset. All extensive experiments on the Landsat-8 and ISPRS datasets demonstrate that the proposed method clearly achieves promising gains compared with the baseline approach.

Figures 13 and 14 show twelve sample testing results from the proposed method on ISPRS Vaihingen corpus. The results of the last column are also similar to the ground truth in the second column same as performed on Landsat-8 corpus. Considering to each class (are shown in Tables 3 and 5), almost every classes (three out of five) from our proposed methods are the winner in term *Average Accuracy*.

As can be seen in Figures 13 and 14, the performance of our best model outperforms other advanced models by a considerable margin on each category, especially for the impervious surface (IS), tree, and car categories. To show the effectiveness of the proposed methods, we performed comparisons against a number of state-of-the-art semantic segmentation methods, as listed in Table 4, Table 5 with respect to the ISPRS corpus, and Tables 2 and 3 with respect to the Landsat-8 corpus. The DCED [31–33] and GCN [15] are the versions with ResNet-50 as their backbone. In particular, we re-implemented the DCED with Tensorflow-Slim [36], since the released code was built on Caffe [38]. We can see that our proposed methods significantly outperform other methods on both the *F*1 *score* and *mean IoU*.

In terms of the computational cost, our framework requires slightly additional training time compared to the baseline approach, DCED, by about 6.25% (6–7 h), and GCN, by about 4.5% (4–5 h). In our experiment, DCED's training procedure took approximately 16 h per dataset, and finished after 50 epochs with 1152 s per epoch. Our framework is a modification of the GCN-based deep learning architecture. The channel attention model increases the time by 20 min compared with the GCN152 method. There is no additional time required when reusing pre-trained weights.

## 6. Conclusions and Future Work

In this study, we propose a novel CNN framework to perform semantic labeling on remotely sensed images. Our proposed method achieves excellent performance by presenting three aspects. First, a global convolutional network (GCN) is employed and enhanced by adding larger numbers of layers to better capture complex features. Second, channel attention is proposed to assign a proper weight for each extracted feature on different stages of the network. Finally, domain-specific transfer learning is introduced to allay the scarcity issue by training the initial weights using other remotely sensed corpora whose resolutions can be different. The experiments were conducted on two datasets: Landsat-8 (medium resolution) and the ISPRS Vaihingen Challenge (very hign resolution) datasets. The results show that our model that combines all proposed strategies outperforms baseline models in terms of *F*1 and mean IoU. The final results show that our enhanced GCN outperforms the baseline (DCED)—17.48% for F1 on the Landsat-8 corpus and 2.48% on the ISPRS corpus.

In the future, more choices of semantic labeling, modern optimization techniques, and/or other novel activation functions will be investigated and compared to obtain the best GCN-based framework for semantic segmentation in remotely sensed images. Moreover, incorporating other data sources (e.g., a digital surface model) might be needed to increase the accuracy of deep learning for both the CNN and the modern deep learning layer with very low confidence simultaneously. These aforementioned issues will be investigated in future research.

**Author Contributions:** T.P. performed all the experiments and wrote the paper; P.V. and T.P. performed the results analysis and edited the manuscript. K.J., S.L., T.P. and P.S. reviewed the results. T.P. revised the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** T. Panboonyuen thanks the scholarship from The 100th Anniversary Chulalongkorn University Fund granted and The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund). We greatly acknowledge Geo-informatics and Space Technology Development Agency (GISTDA), Thailand, for providing satellite imagery used in this study and T. Panboonyuen thanks to the staff from the GISTDA (Thanwarat Anan, Suwalak Nakya, Bussakon Satta) for the supply of LANDSAT-8 imagery and supporting ground data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| BR | Boundary Refinement |
| CNN | Convolutional Neural Network |
| DCED | Deep Convolutional Encoder–Decoder |
| GCN | Global Convolutional Network |
| MR | Medium Resolution |
| RGB | Red–Green–Blue |
| LS | Landsat |
| TL | Transfer Learning |
| VHR | Very High Resolution |

## References

1. Liu, Y.; Fan, B.; Wang, L.; Bai, J.; Xiang, S.; Pan, C. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 78–95. [CrossRef]
2. Wang, H.; Wang, Y.; Zhang, Q.; Xiang, S.; Pan, C. Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sens.* **2017**, *9*, 446. [CrossRef]
3. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
4. Panboonyuen, T.; Vateekul, P.; Jitkajornwanich, K.; Lawawirojwong, S. An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery. In *Recent Advances in Information and Communication Technology Series*; Springer: Cham, Switzerland, 2017; Volume 566.
5. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv* **2016**, arXiv:1606.00915.
6. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
8. Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Road Segmentation of Remotely-Sensed Images Using Deep Convolutional Neural Networks with Landscape Metrics and Conditional Random Fields. *Remote Sens.* **2017**, *9*, 680. [CrossRef]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference onComputer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
10. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
11. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
12. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651.
13. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.

15. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1743–1751.
16. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a Discriminative Feature Network for Semantic Segmentation. *arXiv* **2018**, arXiv:1804.09337.
17. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *arXiv* **2017**, arXiv:1709.01507.
18. Xie, M.; Jean, N.; Burke, M.; Lobell, D.; Ermon, S. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv* **2015**, arXiv:1510.00098.
19. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 3320–3328.
20. Liu, J.; Wang, Y.; Qiao, Y. Sparse Deep Transfer Learning for Convolutional Neural Network. In Proceedings of the AAAI, San Francisco, CA, USA, 4–9 February 2017; pp. 2245–2251.
21. International Society for Photogrammetry and Remote Sensing. 2D Semantic Labeling Challenge. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 9 September 2018).
22. Valada, A.; Vertens, J.; Dhall, A.; Burgard, W. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4644–4651.
23. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder–decoder with atrous separable convolution for semantic image segmentation. *arXiv* **2018**, arXiv:1802.02611.
24. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *arXiv* **2018**, arXiv:1808.00897.
25. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
26. Bilinski, P.; Prisacariu, V. Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6596–6605.
27. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3684–3692.
28. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. *arXiv* **2016**, arXiv:1611.07709.
29. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images. *arXiv* **2017**, arXiv:1704.08545.
30. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
31. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
32. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *arXiv* **2015**, arXiv:1511.00561.
33. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder–decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
34. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
35. Barsi, J.A.; Lee, K.; Kvaran, G.; Markham, B.L.; Pedelty, J.A. The spectral response of the Landsat-8 operational land imager. *Remote Sens.* **2014**, *6*, 10232–10251. [CrossRef]

36. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
38. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.

*remote sensing*

MDPI

*Article*

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

**Teerapong Panboonyuen [1,*], Kulsawasd Jitkajornwanich [2], Siam Lawawirojwong [3], Panu Srestasathiern [3] and Peerapon Vateekul [1,*]**

1 Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd, Pathumwan, Bangkok 10330, Thailand

2 Data Science and Computational Intelligence (DSCI) Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd, Ladkrabang, Bangkok 10520, Thailand; kulsawasd.ji@kmitl.ac.th

3 Geo-Informatics and Space Technology Development Agency (Public Organization), 120, The Government Complex, Chaeng Wattana Rd, Lak Si, Bangkok 10210, Thailand; siam@gistda.or.th (S.L.); panu@gistda.or.th (P.S.)

* Correspondence: teerapong.panboonyuen@gmail.com (T.P.); peerapon.v@chula.ac.th (P.V.)

Received: 5 December 2018; Accepted: 1 January 2019; Published: 4 January 2019


**Abstract:** In the remote sensing domain, it is crucial to complete semantic segmentation on the raster images, e.g., river, building, forest, etc., on raster images. A deep convolutional encoder–decoder (DCED) network is the state-of-the-art semantic segmentation method for remotely sensed images. However, the accuracy is still limited, since the network is not designed for remotely sensed images and the training data in this domain is deficient. In this paper, we aim to propose a novel CNN for semantic segmentation particularly for remote sensing corpora with three main contributions. First, we propose applying a recent CNN called a global convolutional network (GCN), since it can capture different resolutions by extracting multi-scale features from different stages of the network. Additionally, we further enhance the network by improving its backbone using larger numbers of layers, which is suitable for medium resolution remotely sensed images. Second, "channel attention" is presented in our network in order to select the most discriminative filters (features). Third, "domain-specific transfer learning" is introduced to alleviate the scarcity issue by utilizing other remotely sensed corpora with different resolutions as pre-trained data. The experiment was then conducted on two given datasets: (i) medium resolution data collected from Landsat-8 satellite and (ii) very high resolution data called the ISPRS Vaihingen Challenge Dataset. The results show that our networks outperformed DCED in terms of $F1$ for 17.48% and 2.49% on medium and very high resolution corpora, respectively.

**Keywords:** deep convolutional neural networks; multi-class segmentation; global convolutional network; channel attention; transfer learning; ISPRS Vaihingen; Landsat-8

## 1. Introduction

Semantic segmentation of earthly objects such as agriculture fields, forests, roads, and urban and water areas from remotely sensed images has been manipulated in many applications in various domains, e.g., urban planning, map updates, route optimization, and navigation [1–5], allowing us to better understand the domain's images and create important real-world applications.

A deep convolutional neural network (CNN) is a well-known method for automatic feature learning. It can mechanically learn features at different levels and abstractions from raw images by multiple hierarchical stacking convolution and pooling layers [4–14]. To accomplish such a challenging task, features at different levels are required. Specifically, abstract high-level features are more suitable for the recognition of confusing manmade objects, while the labeling of finely structured objects could benefit from detailed low-level features [1]. Therefore, different numbers of layers will affect the performance of deep learning models.

In the past few years, the modern CNNs have been extensively proposed including Global Convolutional Network (GCN) [15] in which the large kernel and effective receptive field play an important role in performing classification and localization tasks simultaneously. The GCN is proposed to address the classification and localization issues for semantic segmentation and to suggest a residual-based boundary refinement for further refining object boundaries. However, this type of architecture ignores the global context such as weights of the features in each stage. Furthermore, most methods of this type are just summed up the features of adjacent stages without considering their diverse representations. 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.

In this paper, we present a novel global convolutional network for segmenting multi-objects from aerial and satellite images. To this end, it is focused on three aspects: (i) varying backbones using ResNet50, ResNet101, and ResNet152, (ii) applying a "channel attention block" [16,17] to assign weights for feature maps in each stage of the backbone architecture, and (iii) employing "domain-specific transfer learning" [18–20] to relieve scarcity. Experiments were conducted using satellite imagery (from the Landsat-8 satellite), which was provided by a government organization in Thailand, and using well-known aerial imagery from the ISPRS Vaihingen Challenge corpus [21], which is publicly available. The results showed that our method outperforms the baseline including deep convolutional encoder–decoder (DCED) in terms of $F1$ and by mean of class-wise intersection over union (*mean IoU*).

The remainder of this paper is arranged as follows. The related work is discussed in Section 2. Section 3 describes our proposed methodology. Experimental datasets and evaluations are described in Section 4. Experimental results and discussions are presented in Section 5. Finally, we conclude our work and discuss future work in Section 6.

## 2. Related Work

Deep learning has been successfully applied for remotely sensed data analysis, notably land cover mapping on urban areas [1–3], and has increasingly become a promising tool for accelerating the image recognition process with high accuracy [4–14,22–30]. It is a fast-growing field, and new architectures appear every few days. This section is divided into three subsections: we discuss deep learning concepts for semantic segmentation, a set of multi-objects segmentation techniques using modern deep learning architectures, and modern techniques of deep learning.

### *2.1. Deep Learning Concepts for Semantic Segmentation*

Semantic segmentation algorithms are often formulated to solve structured pixel-wise labeling problems based on a deep CNN. Noh et al. [13] proposed a novel semantic segmentation technique utilizing a deconvolutional neural network (DCNN) and the top layer from the DCNN adopted from

VGG16 [4,8]. The DCNN structure is composed of upsampling layers and deconvolution layers, describing pixel-wise class labels and predicting segmentation masks, respectively. Their proposed deep learning methods yield high performance in PASCAL VOC 2012 corpus, with the 72.5% accuracy in the best-case scenario (the highest accuracy—as of the time of the writing of this paper—compared to other methods that were trained without requiring additional or external data). Long et al. [12] proposed adapted contemporary classification networks incorporating Alex, VGG, and GoogLe networks into a fully CNN. In this method, some of the pooling layers were skipped: Layer 3 (FCN-8s), Layer 4 (FCN-16s), and Layer 5 (FCN-32s). The skip architecture reduces the potential over-fitting problem and has shown improvements in performance, ranging from 20% to 62.2% in the experiments tested on PASCAL VOC 2012 data. Ronneberger et al. [14] proposed U-Net, a DCNN for biomedical image segmentation. The architecture consists of a contracting path and a symmetric expanding path that captures context and consequently enables precise localization. The proposed network claimed to be capable of learning despite the limited number of training images and performed better than the prior best method (a sliding-window DCNN) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Vijay Badrinarayanan [31–33] proposed a deep convolutional encoder–decoder network (DCED), called "SegNet," that consists of two main networks, encoder and decoder, and some outer layers. The two outer layers of the decoder network are responsible for feature extraction, the results of which are transmitted to the layer adjacent to the last layer of the decoder network. This layer is responsible for pixel-wise classification (determining which pixel belongs to which class). There is no fully connected layer in between feature extraction layers. In the upsampling layer of the decoder, pool indices from the encoder are distributed to the decoder, where the kernel will be trained in each epoch (the training round) at the convolution layer. In the last layer (classification), softmax was used as a classifier for pixel-wise classification. The DCED is one of the deep learning models that exceeds the state of the art on many remote sensing corpus.

In this work, the DCED method was selected as our baseline since it is the most popular architecture used in various networks for semantic segmentation.

### *2.2. Modern Deep Learning Architectures For Semantic Segmentation*

Recently, many approaches based on the DCED have achieved high performance on different benchmarks [16,31–33]. However, most of them still suffer from accuracy performance issues. Therefore, many works of modern deep learning architectures have been proposed, such as instance-aware semantic segmentation [34], which is slightly different from semantic segmentation. Instead of labeling all pixels, it focuses on the target objects and labels only pixels of those objects. FCIS [28] is based on techniques based on fully convolutional networks (FCNs). The mask R-CNN [9] was built around the FCN and is incorporated with a proposed joint formulation. Peng [15] presented the concept of large kernel matters to improve semantic segmentation with a global convolutional network (GCN) as shown in Figure 1. They proposed a GCN to address both the classification and localization issues for semantic segmentation. Large separable kernels were used to expand the receptive field, and a boundary refinement block was added to further improve localization performance near the boundaries. From the Cityscapes challenge, the GCN outperforms methods of all previous publications (all modern deep learning baselines) and has become the new state of the art. Therefore, the GCN was selected as our proposed method and as the main model of our work.

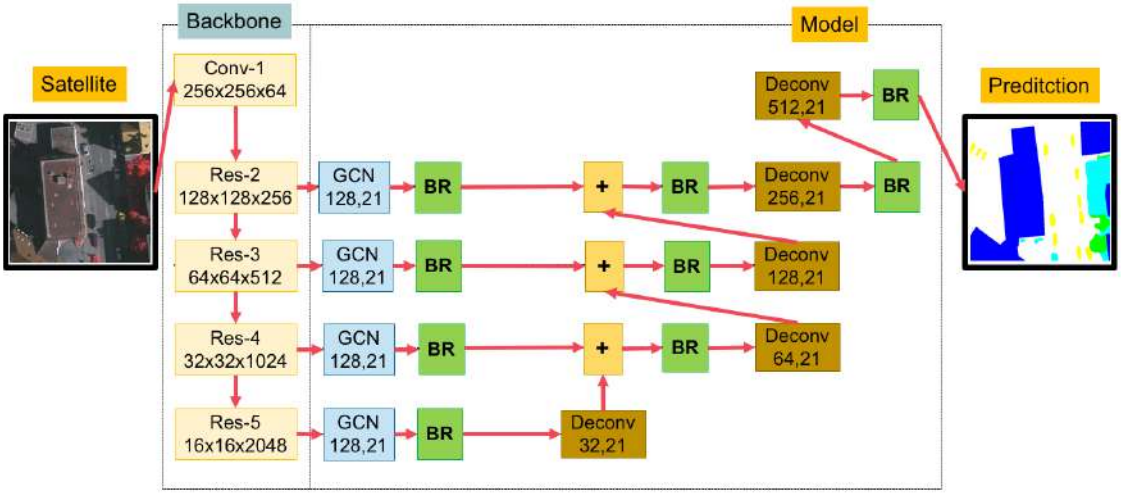



**Figure 1.** An overview of the original global convolutional network (GCN) and boundary refinement (BR) [15].

### *2.3. Modern Techniques of Deep Learning*

Modern techniques of deep learning are important for the accuracy of a CNN. The most popular modern ideas used for semantic segmentation tasks, such as global context, the attention module, and semantic boundary detection, have been used for boosting accuracy.

Global context [16] is a modern method that has proven the effectiveness of global average pooling in the semantic segmentation task. For example, PSPNet [30] and Deeplab v3 [5] respectively extend it to spatial pyramid pooling [30] and atrous spatial pyramid pooling [5], resulting in great performance at different benchmarks. However, to take advantage of the pyramid pooling module sufficiently, these two methods adopt the base feature network to downsample with atrous convolution eight times [5], which is time-consuming and memory-intensive.

Attention Module [16]: Attention is helpful to focus on what we want. Recently, the attention module has increasingly become a powerful tool for deep neural networks [16,17]. The method in [16,17] pays attention to different scale information. In this work, we utilize a channel attention block to select features, similar to learning a discriminative feature network [16].

Refinement Residual Block [16]: The feature maps of each stage in the feature network all go through the refinement residual block. For our work, we use the boundary refinement block (BR) to be a concept of "refinement residual block" from [15]. The first component of the block is a $1 \times 1$ convolution layer. We use it to unify the number of channels to 21. Meanwhile, it can combine the information across all channels. Then the following is a basic residual block [7], which can refine the feature map. Furthermore, this block can strengthen the recognition ability of each stage, inspired from the architecture of ResNet.

## 3. The Proposed Method

In this section, the details of our proposed network are explained (shown in Figure 2). The network is based on the GCN with three aspects of improvements: (i) the modification of backbone architecture (shown in P1 in Figure 2), (ii) applying the channel attention block (shown in P2 in Figure 2), and (iii) using the concept of domain-specific transfer learning (shown in P3 in Figure 2).

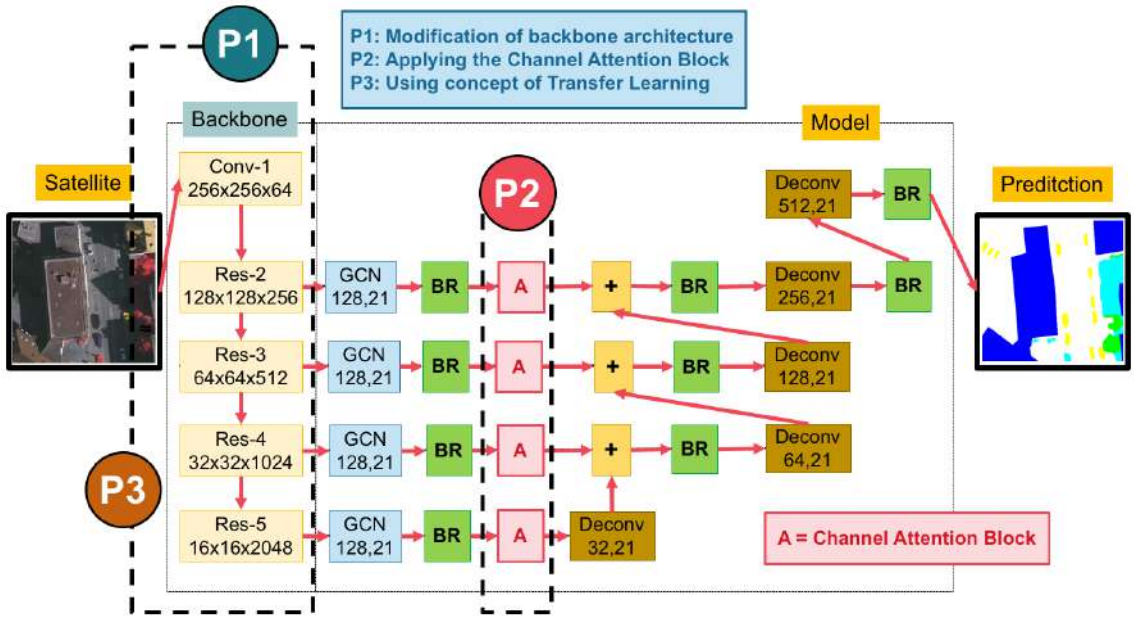



**Figure 2.** An overview of our proposed network.

### *3.1. Data Preprocessing*

In this paper, there are two benchmark corpuses, including (i) the ISPRS Vaihingen Challenge corpus and (ii) the Landsat-8 dataset. They are comprised of very high and medium resolution images, respectively. More details of the datasets will be explained in Sections 4.1 and 4.2. Before a discussion of the model, it is worth explaining our data preprocessing procedure, since it is required when working with neural network and deep learning models. Thus, the mean subtraction is executed.

In addition, data augmentation is often required on more complex object recognition tasks. Therefore, a random horizontal flip is generated to increase the training data. For the ISPRS corpus, all images are standardized and cropped into $512 \times 512$ pixels with a resolution of 9 $cm^2$/pixel. For the Landsat-8 corpus, each image is also flipped horizontally and scaled to $512 \times 512$ with a resolution of 30 $m^2$/pixel from the original images ($16{,}800 \times 15{,}800$ pixels).

### *3.2. A Global Convolutional Network (GCN) with Variations of Backbones*

GCN [15] as shown in Figure 1 is a modern architecture that surpasses the drawbacks of a traditional semantic segmentation network, such as deep convolutional encoder–decoder (DCED) networks. A traditional network usually cascades convolutional layers in order to generate sophisticated features; they can be considered as local features that are specialized for a specific task. However, it is not necessary to employ only specialized features; the general features are also important. Thus, a GCN overcomes this issue by introducing a multi-level architecture, where each level aims to capture a different resolution of features, so both local and global features are considered in the model.

As shown in Figure 1, there are two main blocks in the GCN: a localization block and a classification block. From the localization view in the left block, the structure is a stack of classical fully convolutional layers called "levels." Each level aims to construct features with different resolutions. From the classification view, there are two modules: the GCN and the boundary refinement (BR). For the GCN module, the kernel size of the convolutional structure should be as large as possible, which is motivated by the densely connected structure of the classification models. If the kernel size increases to the spatial size of the feature map (named the global convolution), the network will share the same benefits with the pure classification models. The BR module is added to further improve localization performance near the boundaries.

Although the GCN architecture has shown promising prediction performance, it is still possible to further improve by varying backbones using ResNet [7] with different numbers of layers as ResNet50, ResNet101, and ResNet152, as shown in Figure 3. Additionally, the GCN is suggested to work on a large kernel size. In this paper, we set the large kernel size as 9 (this previous work [15]).

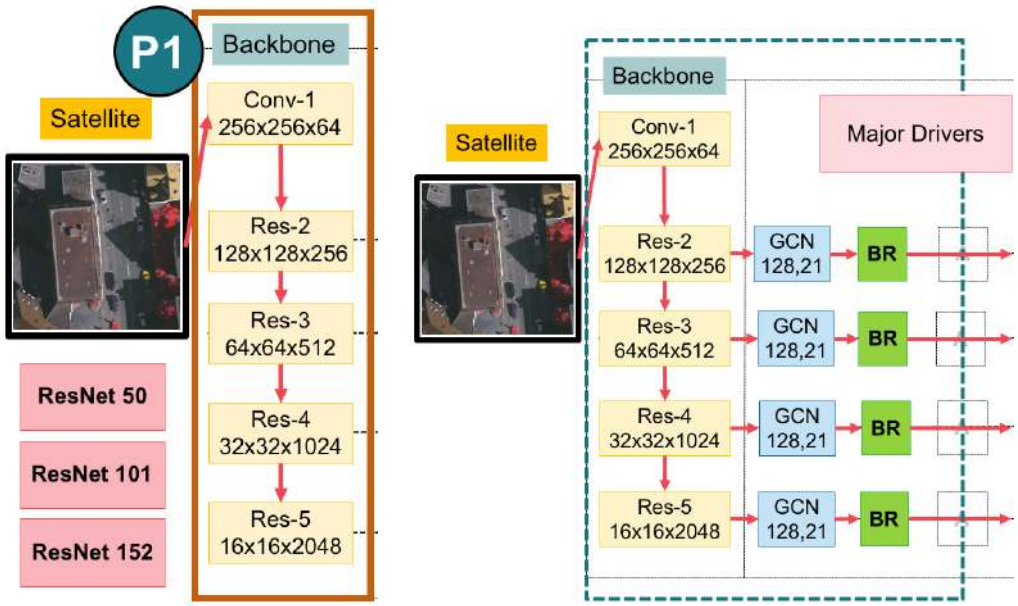



**Figure 3.** An overview of the whole backbone pipeline in (**left**) the main backbone with varying by ResNet50, ResNet101, and ResNet152; (**right**) the major drivers of our main classification network (composed of a global convolutional network (GCN) and a boundary refinement (BR) block [15]).

### *3.3. The Channel Attention Block*

Attention mechanisms [16,17] in neural networks are very loosely based on the visual attention mechanism found in humans and equips a neural network with the ability to focus on a subset of its inputs (or features): it selects specific inputs. Human visual attention is well-studied, and while there are different models, all of them essentially come down to being able to focus on a certain region of an image with a very high resolution, perceiving the surrounding image in a medium resolution, and then adjusting the focal point over time.

To apply this atttentional layer to our network, the channel attention block is shown in Block A in Figure 2 and its detailed architecture is shown in Figure 4. It is designed to change the weights of the remote sensing features on each stage (level), so that the weights are assigned more values on important features adaptively.

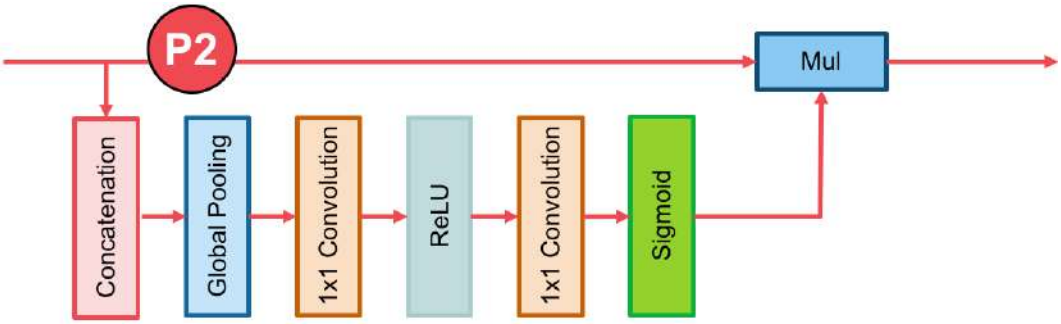



**Figure 4.** Components of the channel attention block. The red lines represent the downsample operators, respectively. The red line cannot change the size of feature maps. It is only a path for information passing.

In the proposed architecture, a convolution operator gives the probability of each class at each pixel. In Equation (1), the final score is summed over all channels of the feature maps.

$$y_k = F(x; w) = \sum_{i=1,j=1}^{D} w_{i,j} x_{i,j} \tag{1}$$

where $x$ is the output feature of network. $w$ represents the convolution's kernel, and $k \in 1, 2, 3, 4, 5, 6, 7, \ldots, K$. The number of channels is represented by $K$, and $D$ is the set of pixel positions.

$$\delta_i(y_k) = \frac{exp(y_k)}{\sum_{j=1}^{k} exp(y_j)} \tag{2}$$

where $\delta$ is the prediction probability. $y$ is the output of the network. As shown in Equations (1) and (2), the final predicted label is the category with the highest probability. Therefore, we suppose that the prediction result is $y_0$ of a certain patch, while its true label is $y_1$. Therefore, we can introduce a parameter $\alpha$ to change the highest probability value from $y_0$ to $y_1$, as Equation (3) shows.

$$\bar{y} = \alpha y = \begin{bmatrix} \alpha_1 \\ . \\ . \\ . \\ \alpha_k \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_k \end{bmatrix} = \begin{bmatrix} \alpha_1 w_1 \\ . \\ . \\ . \\ \alpha_k w_k \end{bmatrix} \times \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_k \end{bmatrix} \tag{3}$$

where $\bar{y}$ is the new prediction of the network, and $\alpha = Sigmoid(x; w)$.

Based on the above formulation of the Channel Attention Block, we can explore its practical significance. In Equation (1), it implicitly indicates that the weights of different channels are equal. However, the features in different stages have different degrees of discrimination, which results in different consistency of prediction. Consequently, in Equation (3), the $\alpha$ value applies the feature maps $x$, which represents the feature selection with the channel attention block.

### *3.4. Domain-Specific Transfer Learning*

The overall idea of transfer learning is to use knowledge learned from tasks for which many labeled data are usable in settings where only little-labeled data are available. Creating labeled data is expensive, so optimally leveraging an existing dataset is key. Certain low-level features, such as edges, shapes, corners, and intensity, can be shared across tasks, and new high-level features specific to the target problem can be learned [18]. Additionally, knowledge from an existing task acts as an additional input when learning a new target task.

Although the deep learning approach often performs promising prediction performance, it requires a large amount of training data. Since it is difficult to obtain annotated satellite images, the performance in prior works has been limited.

Fortunately, there is a recent concept called domain-specific transfer learning [18–20] that allows one to reuse the weights obtained from other domains' inputs. It is currently very popular in the field of deep learning because it enables one to train deep neural networks with comparatively insufficient data. This is very useful since most real-world problems typically do not have millions of labeled data points to train such complex models.

In terms of inadequacy, we propose an effective transfer deep neural network to perform knowledge transfer between a very high resolution (VHR) corpus and a medium resolution (MR) corpus. It is shown in Figure 5.

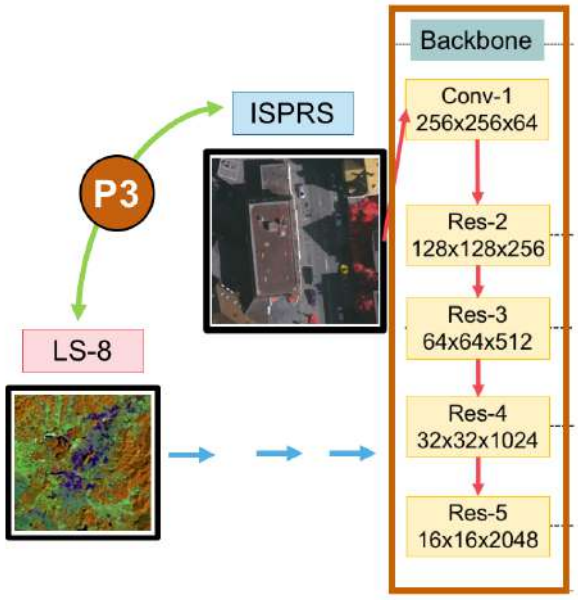



**Figure 5.** The domain-specific transfer learning strategy reuses pre-trained weights of models between two datasets—very high (ISPRS) and medium (Landsat-8; LS-8) resolution images.

## 4. Experimental Datasets and Evaluation

In our experiments, two types of datasets were used: (i) medium resolution imagery (satellite images; Landsat-8 dataset) made by the government organization in Thailand, named GISTDA (Geo-Informatics and Space Technology Development Agency (Public Organization)), and (ii) very high resolution imagery (aerial images; ISPRS Vaihingen dataset). All experiments were evaluated based on major metrics, such as *average accuracy*, *F*1 score, and *mean IoU* score.

### *4.1. Landsat-8 Dataset*

Landsat-8 is an American earth observation satellite and it collects and archive medium resolution (30-m spatial resolution) multispectral image data affording seasonal coverage of the global landmasses for a period of no less than 5 years. Landsat-8 [35] images consist of nine spectral bands with a spatial resolution of 30 m for Bands 1–7 and 9. The ultra blue Band 1 is useful for coastal and aerosol studies. Band 9 is useful for cirrus cloud detection. The resolution for Band 8 (panchromatic) is 15 m. Thermal Bands 10 and 11 are useful in providing more accurate surface temperatures and are collected at 100 m. The approximate scene size is 170 km north–south by 183 km east–west (106 mi by 114 mi). Since Landsat-8 data includes additional bands, the combinations used to create RGB composites differ from Landsat 7 and Landsat 5. For instance, Bands 4, 3, and 2 are used to create a color infrared (CIR) image using Landsat 7 or Landsat 5. To create a CIR composite using Landsat 8 data, Bands 5, 4, and 3 are used.

In this type of data, the satellite images are from Nan, a province in Thailand. The dataset is obtained from Landsat-8 satellite consisting of 1012 satellite images as shown by some samples in Figure 6.

This corpus is comprised of a large, diverse set of medium resolution (16,800 × 15,800) pixels, where 1012 of these images have high quality pixel-level labels of five classes: agriculture, forest, miscellaneous, urban, and water. The 1012 images were split into 800 training and 112 validation images with publicly available annotation, as well as 100 test images with annotations withheld, and comparison to other methods were performed via a dedicated evaluation server. For quantitative evaluation, mean of class-wise intersection over union (*mean IoU*) and *F*1 score are used.

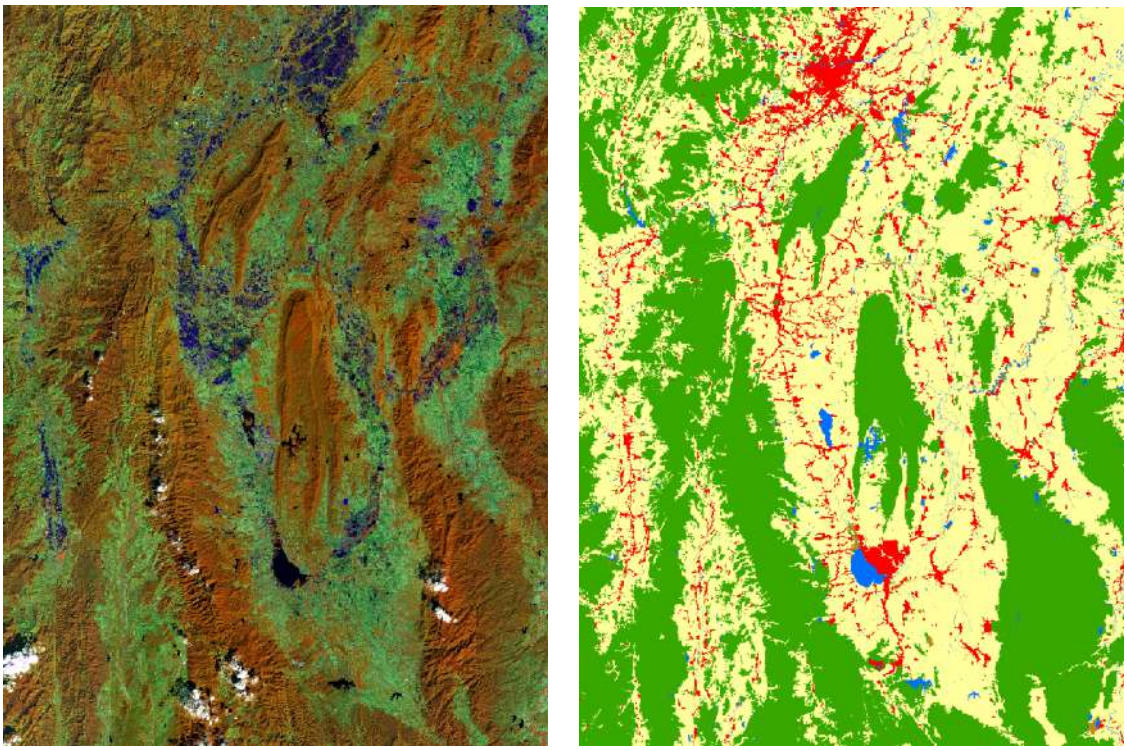


**Figure 6.** Sample satellite images from Nan, a province in Thailand (**left**), and corresponding ground truth (**right**). The label of medium resolution dataset includes five categories: agriculture (yellow), forest (green), miscellaneous (brown), urban (red), and water (blue).

### *4.2. ISPRS Vaihingen Dataset*

One of the major challenges in remote sensing is the automated extraction of urban objects from data acquired by airborne sensors. The Semantic Labeling Contest provides two state-of-the-art airborne image corpora. The Vaihingen corpus shows a relatively small village with many detached buildings and small multi-story buildings, and the Potsdam corpus shows a typical historic city with large building blocks, narrow streets, and dense settlement structure. In our experiments, the Vaihingen corpus was selected and used.

The ISPRS 2D Semantic labeling challenge in Vaihingen [21] (Figures 7 and 8) was used as our benchmark dataset. It consists of three spectral bands (i.e., red, green, and near-infrared bands), the corresponding DSM (digital surface model) and the NDSM (normalized digital surface model) data. Overall, there are 33 images of about 2500 $\times$ 2000 pixels at a ground sampling distance (GSD) of about 9 cm in the image data. Among them, the ground truth of only 16 images are available, and those of the remaining 17 images are withheld by the challenge organizer for the online test. For offline validation, we randomly split the 16 images with ground truth available into a training set of 10 images and a validation set of 6 images. For this work, DSM and NDSM data in all experiments on this dataset were not used. Following other methods, four tiles (Image Numbers 5, 7, 23, and 30) were removed from the training set as the validation set. Experimental results are reported on the validation set if not specified.


**Figure 7.** Overview of the ISPRS 2D Vaihingen Labeling corpus. There are 33 tiles. Numbers in the figure refer to the individual tile flag.

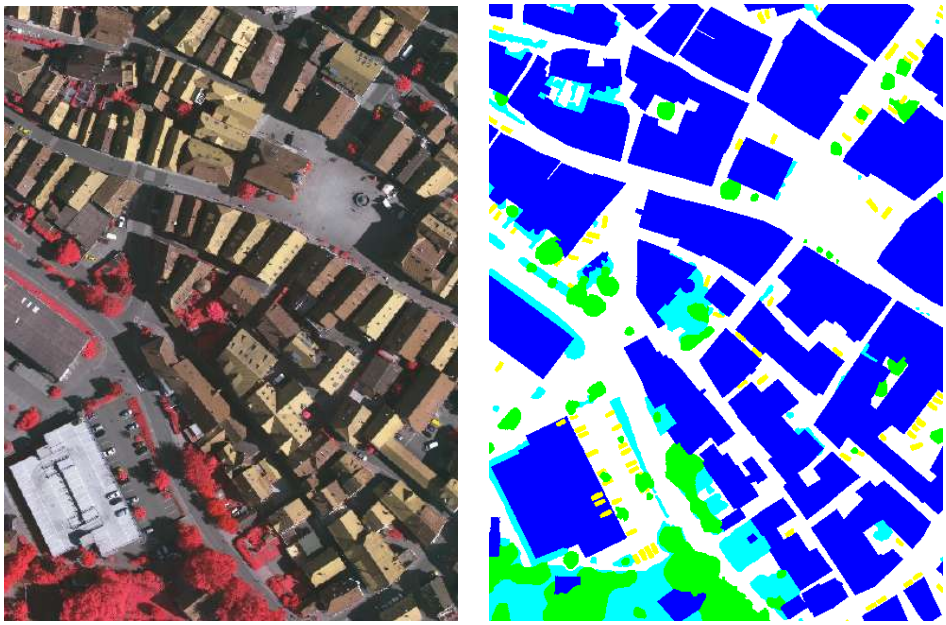


**Figure 8.** The sample input tile from Figure 7 (**left**) and corresponding ground truth (**right**). The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow), and clutter/background (red).

### *4.3. Evaluation*

The multi-class classification task can be considered as multi-segmentation, where class pixels are positives and the remaining non-spotlight pixels are negatives. Let $TP$ denote the number of true positives, $TN$ denote the number of true negatives, $FP$ denote the number of false positives, and $FN$ denote the number of false negatives.

*Precision*, *recall*, $F1$, and *mean IoU* are shown in Equations (4)–(8). Precision is the percentage of correctly classified main pixels among all predicted pixels by the classifier. Recall is the percentage of correctly classified main pixels among all actual main pixels. $F1$ is a combination of *precision* and *recall*.

To evaluate the performance of different deep models, we will discuss the above two major metrics ($F1$), the mean of class-wise intersection over union (*mean IoU*)) on each category, and the mean value of metrics to assess the average performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{7}$$

$$Mean\ IoU = \frac{TP}{TP + FP + FN}. \tag{8}$$

## 5. Experimental Results and Discussion

The implementation is based on a deep learning framework, called "Tensorflow-Slim" [36], which is extended from Tensorflow. All experiments were conducted on servers with an Intel® Xeon® Processor E5-2660 v3 (25M Cache, 2.60 GHz), 32 GB of memory (RAM), an Nvidia GeForce GTX 1070 (8 GB), an Nvidia GeForce GTX 1080 (8 GB), and an Nvidia GeForce GTX 1080 Ti (11 GB). Instead of using the whole image (1500 × 1500 pixels) to train the network, we randomly cropped all images to be 512 × 512 as inputs of each epoch.

For training, the Adam optimizer [11] was chosen with an initial learning rate of 0.004 and the weight decay of 0.00001. Batch normalization [10] is used before each convolutional layer in our implementation to ease the training and make it be able to concatenate feature maps from different layers. To avoid overfitting, common data augmentations are used as details in Section 3.1. For measurements, we use the mean pixel intersection-over-union (*mean IoU*) and the *F*1 score as the metric.

Inspired by [16,27,37], we use the "poly" learning rate policy where the learning rate is multiplied by Equation (9) with a power of 0.9 and an initial learning rate as $4 \times 10^{-3}$. The learning rate is scheduled by multiplying the initial as seen in Equation (9).

$$\text{learning rate} = (1 - \frac{epoch}{MaxEpoch})^{0.9}. \tag{9}$$

All models are trained for 50 epochs with a mini-batch size of 4, and each batch contains the cropped images that are randomly selected from training patches. These patches are resized to 521 × 521 pixels. The statistics of BN is updated on the whole mini-batch.

This section illustrates the details of our experiments. The proposed deep learning network is based on the GCN with three improvements: (i) varying the backbones using ResNet, (ii) channel attention and global average pooling, and (iii) domain-specific transfer learning. From all proposed strategies, there are six acronyms of strategies as shown in Table 1.

**Table 1.** Abbreviations on our proposed deep learning methods.

| 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 |

For the experimental setup, there were three experiments on two remotely sensed datasets: the Landsat-8 dataset and the ISPRS Vaihingen Challenge dataset (details in Sections 4.1 and 4.2). The experiments aimed to illustrate that each proposed strategy can improve the performance. First, the GCN152 method was compared to the GCN50 method and the GCN101 method for the varying backbones using ResNet with different numbers of layers on the GCN network strategy. Second, the GCN152-A method was compared to the GCN152 method for the channel attention strategy. Third, the full proposed technique GCN152-TL-A method was compared to existing methods for the concept of domain-specific transfer learning.

### *5.1. Results of the Landsat-8 Corpus with Discussion*

An experiment was conducted on the Landsat-8 corpus, and the result is shown in Tables 2 and 3 by comparing between baseline and variations of the proposed techniques. It is shown that our network with all strategies, GCN152-TL-A, outperforms other methods. More details will be discussed to show that each of the proposed techniques can improve accuracy. Only in this experiment is there a state-of-the-art baseline, including a deep convolutional encoder–decoder (DCED) [31–33].

**Table 2.** Results of the testing data of the Landsat-8 corpus between baseline and five variations of our proposed techniques in terms of *precision*, *recall*, *F*1, and *mean IoU*.

| | Pretrained | Backbone | Model | *Precision* | *Recall* | *F*1 | *mean IoU* |
|---|---|---|---|---|---|---|---|
| **Baseline** | - | - | DCED [31–33] | 0.6137 | 0.7209 | 0.6495 | 0.5384 |
| **Proposed Method** | - | Res50 | GCN [15] | 0.6678 | 0.7333 | 0.6847 | 0.5734 |
| | - | **Res101** | GCN | 0.6899 | 0.8031 | 0.7290 | 0.6154 |
| | - | **Res152** | GCN | 0.7115 | 0.8131 | 0.7563 | 0.6364 |
| | - | **Res152** | **GCN-A** | 0.7997 | 0.7937 | 0.7897 | 0.6726 |
| | **TL** | **Res152** | **GCN-A** | **0.8293** | **0.8476** | **0.8275** | **0.7178** |

**Table 3.** Results of the testing data of Landsat-8 corpus between each class with our proposed techniques in terms of *averageaccuracy*.

| | Model | Agriculture | Forest | Misc | Urban | Water |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [31–33] | 0.9616 | 0.7472 | 0.0976 | 0.7878 | 0.4742 |
| **Proposed Method** | GCN50 [15] | 0.9407 | 0.8258 | 0.1470 | **0.8828** | 0.5426 |
| | GCN101 | 0.9677 | 0.8806 | 0.2561 | 0.7971 | 0.5480 |
| | GCN152 | 0.9780 | 0.8444 | 0.4256 | 0.7158 | 0.5937 |
| | **GCN152-A** | 0.9502 | **0.9118** | 0.6689 | 0.8675 | 0.6001 |
| | **GCN152-TL-A** | **0.9781** | 0.8472 | **0.8732** | 0.7988 | **0.6493** |

#### 5.1.1. The Effect of an Enhanced GCN on the Landsat-8 Corpus

Our first strategy aims to increase an *F*1 and *mean IoU* score of the network by varying backbones using ResNet 50, ResNet 101, and ResNet 152 rather than the traditional one, the DCED method. From Tables 2 and 3, the *F*1 of GCN152 (0.7563) outperforms that of GCN50 (0.6847), GCN101 (0.7290), and the baseline method, DCED (0.6495); this yields a higher *F*1 at 2.74%, 3.52%, and 4.43%, respectively. The *mean IoU* of GCN152 (0.6364) outperforms that of GCN50 (0.5734), GCN101 (0.6154), and the baseline method, DCED (0.5384); this yields a higher *mean IoU* at 2.10%, 3.50%, and 4.20%, consecutively. The main reason is due to higher precision, but a slightly lower recall. This can imply that enhanced GCN is more significantly efficient than the DCED method (baseline) for this medium resolution corpus and ResNet with a large number of layers is more robust than the small number of layers.

When comparing the results between the original GCN method and the enhanced GCN methods on the Landsat-8 corpus (Table 2), it is clearly shown that a GCN with a larger layer of backbone can improve network performance in terms of *F*1 and *mean IoU*.

#### 5.1.2. The Effect of Using Channel Attention on the Landsat-8 Corpus

Our second mechanism focused on applying the channel attention block (details in Section 3.4) to change the weights of the features on each stage to enhance consistency. In Tables 2 and 3, the $F1$ of GCN152-A (0.7897) is greater than that of GCN152 (0.7563); this yields a higher $F1$ score at 3.34%. The *mean IoU* of GCN152-A (0.6726) is superior to that of GCN152 (0.6364); this yields a higher *mean IoU* score at 3.62%. The result (Figures 9e and 10e) shows that can make the network to obtain discriminative features stage-wise to make the prediction intra-class consistent. This is based on the consideration that we re-weighted all feature maps of each layer.

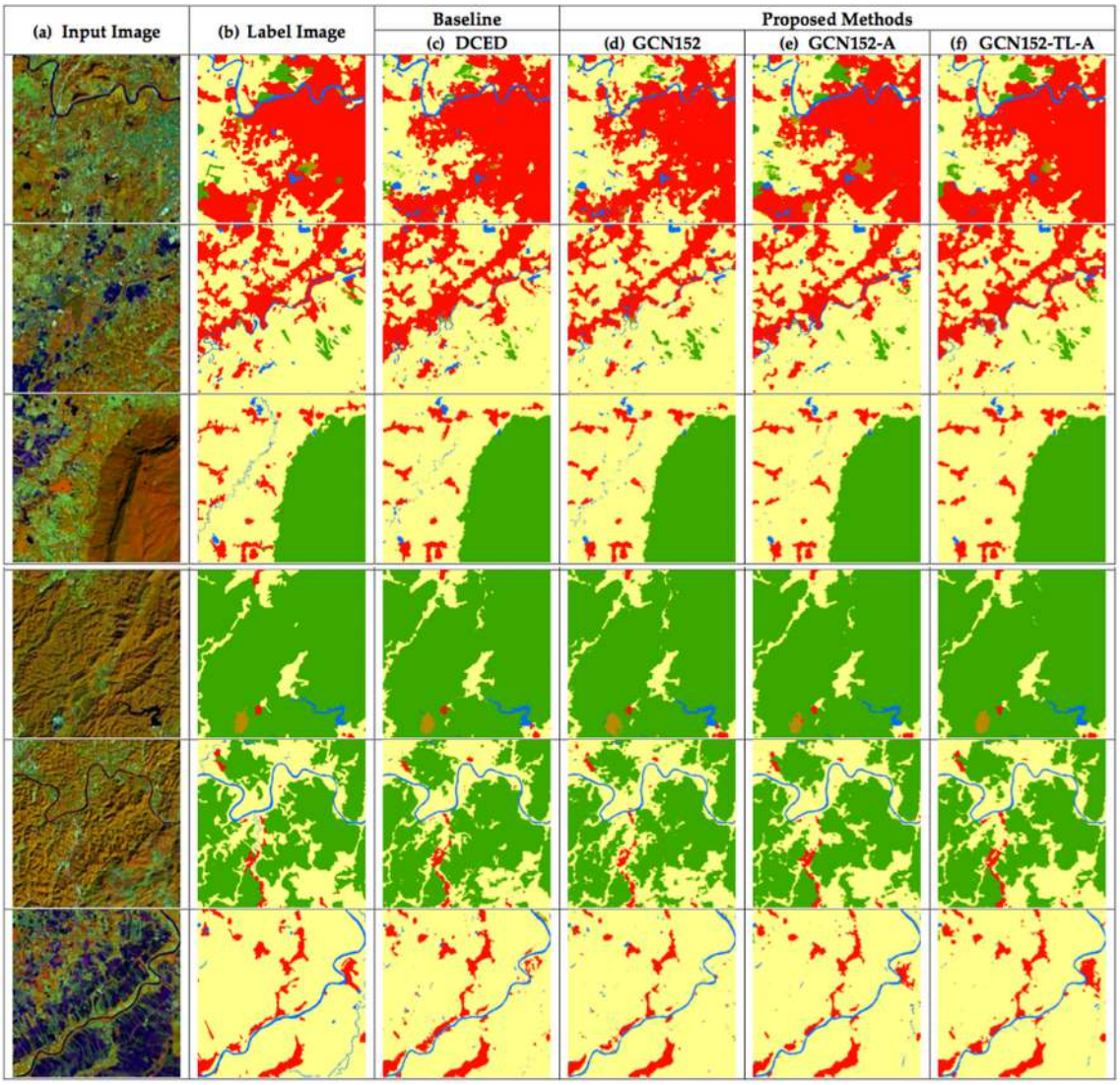



**Figure 9.** Six testing sample inputs and output satellite images on Landsat-8 in the Nan province in Thailand, where rows refer to different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of medium resolution dataset includes five categories: Agriculture (yellow), Forest (green), Miscellaneous (Misc, brown), Urban (red) and Water (blue).

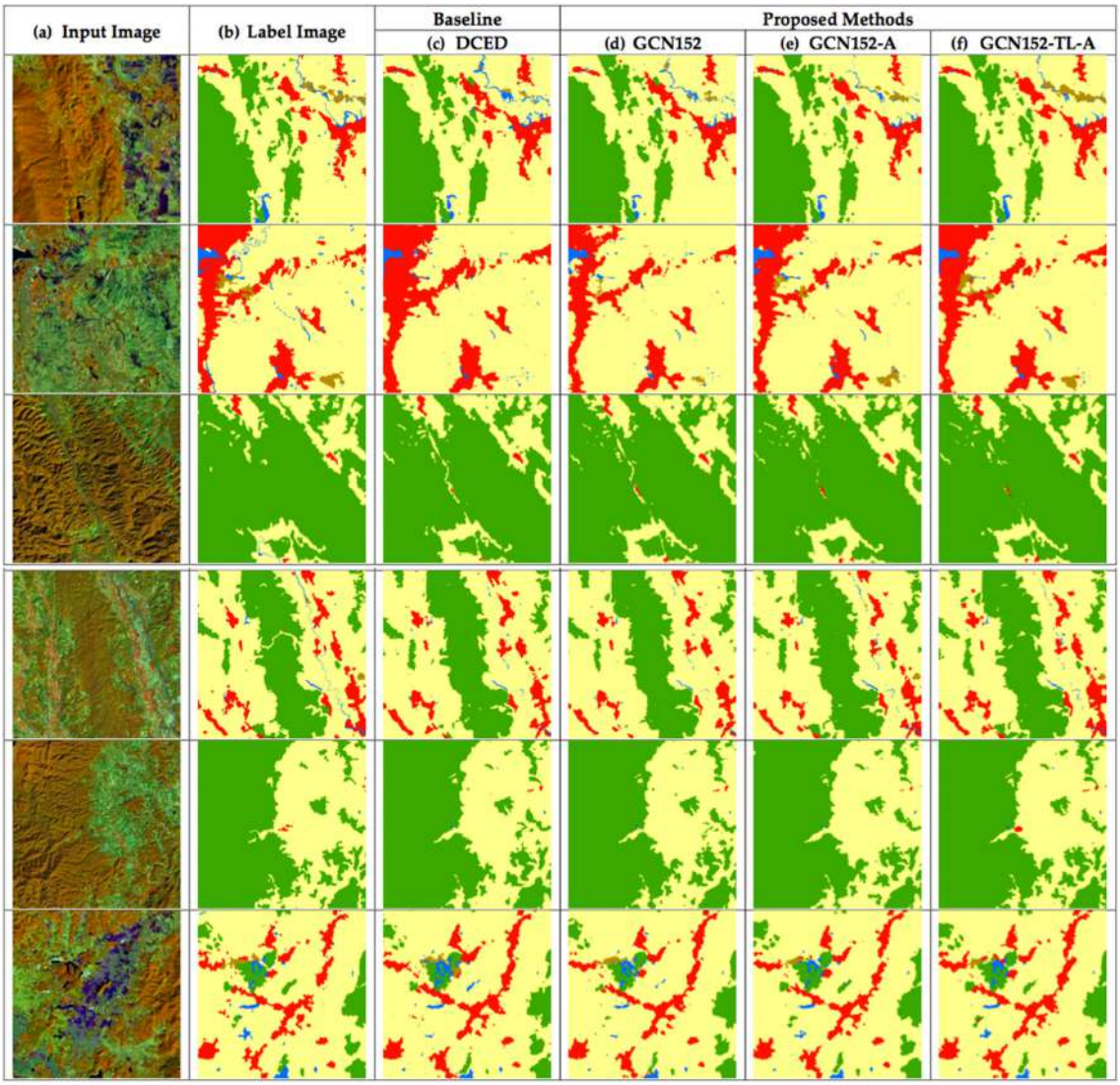



**Figure 10.** Six testing sample input and output satellite images on Landsat-8 in Nan in Thailand, where rows refer to different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of medium resolution dataset includes five categories: Agriculture (yellow), Forest (green), Miscellaneous (Misc, brown), Urban (red) and Water (blue).

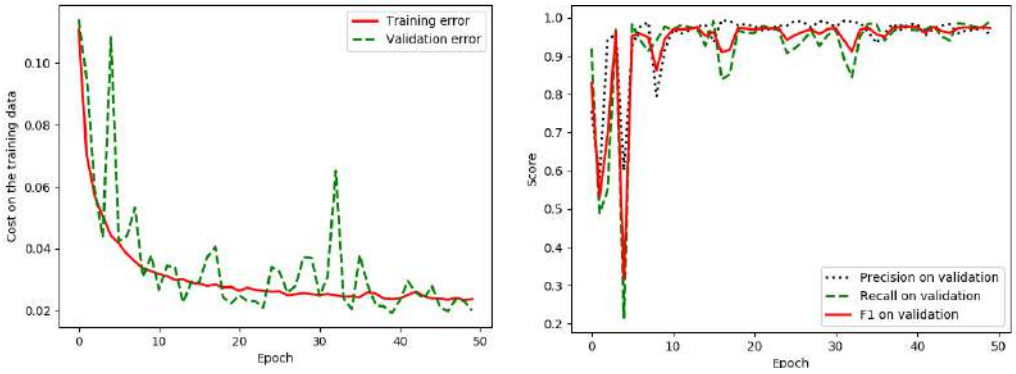
#### 5.1.3. The Effect of Using Domain-Specific Transfer Learning on Landsat-8 Corpus

Our last strategy aims to use approach of domain-specific transfer learning (details in Section 3.3) by reusing the pre-trained weight from the GCN152-A model on the ISPRS Vaihingen corpus. From Tables 2 and 3, the *F*1 of the GCN152-TL-A method is the winner; it clearly outperforms not only the baseline but also all previous generations. Its *F*1 is higher than that of the DCED (baseline) at 17.80%. Its *mean IoU* is higher than that of the DCED at 17.94%. Additionally, the result illustrates that the concept of domain-specific transfer learning can enhance both precision (0.8293) and recall (0.8476).

Figures 9 and 10 show 12 sample results from the proposed method. By applying all strategies, the images in the last column (Figures 9f and 10f) are similar to the ground truths (Figures 9b and 10b). Furthermore, *F*1-results and *mean IoU* scores are improved for each strategy we added to the network as shown in Figures 9c–f and 10c–f.

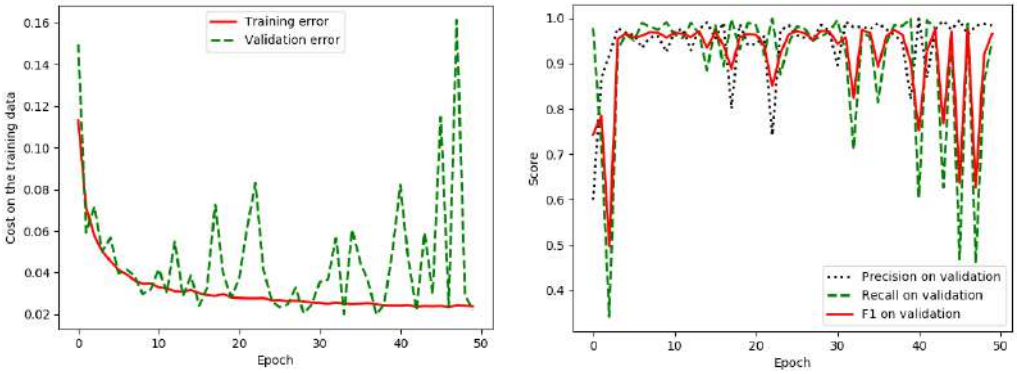
To achieve the highest accuracy, the network must be configured and many epochs must be trained until all parameters in the network are converged. Figure 11a illustrates that the proposed network has been properly set and trained until it is converged and runs more smoothly than the

baseline in Figure 12a. Furthermore, Figures 11b and 12b show that a higher number of epochs tend to show a better $F1$ score. Thus, the number of chosen epochs based on the validation data is 49 (the best model for this dataset).




(**a**) (**b**)

**Figure 11.** Iteration plot on Landsat-8 corpus of the proposed technique, GCN152-TL-A; $x$ refers to epochs and $y$ refers to different measures (**a**) Plot of model loss (cross entropy) on training and validation datasets; (**b**) performance plot on the validation dataset.




(**a**) (**b**)

**Figure 12.** Iteration plot on the Landsat-8 corpus of the baseline technique, the DCED [31–33]; $x$ refers to epochs and $y$ refers to different measures. (**a**) The plot of model loss (cross entropy) on training and validation datasets; (**b**) the performance plot on the validation dataset.

Twelve sample testing results (shown as Figures 9 and 10) are based on the proposed method with respect to Nan (one of the northern provinces (changwat) of Thailand and where agriculture is the main industry). The results of the last column look closest to the ground truth in the second column.

As can be seen in Figures 9 and 10, the performance of our best model outperforms other advanced models by a considerable margin on each category, especially for the agriculture, miscellaneous (Misc), and water classes. Furthermore, the loss curves shown in Figure 11a exhibit that our model performs better on all given categories.

### *5.2. Results of the ISPRS Vaihingen Challenge Corpus with Discussion*

An experiment was conducted on the ISPRS Vaihingen Challenge corpus, and the result is shown in Tables 4 and 5 by comparing between baseline and variations of the proposed techniques. This shows that our network with all strategies (GCN152-TL-A) outperforms other methods. More details will be discussed to show that each of the proposed techniques can improve accuracy. Only in this experiment is there one baseline, which is the DCED network.

**Table 4.** Results of the testing data of the ISPRS 2D semantic labeling challenge corpus between the baseline and five variations of our proposed techniques in terms of *precision*, *recall*, *F*1, and *mean IoU*.

| | Pretrained | Backbone | Model | *Precision* | *Recall* | *F*1 | *mean IoU* |
|---|---|---|---|---|---|---|---|
| **Baseline** | - | - | DCED [31–33] | 0.7519 | 0.7925 | 0.7693 | 0.8651 |
| **Proposed Method** | - | Res50 | GCN [15] | 0.7636 | 0.7917 | 0.776 | 0.8776 |
| | - | **Res101** | GCN | 0.7713 | **0.8059** | 0.7862 | 0.8972 |
| | - | **Res152** | GCN | 0.7736 | 0.8021 | 0.7864 | 0.8977 |
| | - | **Res152** | **GCN-A** | 0.7847 | 0.7961 | 0.7902 | 0.9057 |
| | **TL** | **Res152** | **GCN-A** | **0.7888** | 0.8001 | **0.7942** | **0.9123** |

**Table 5.** Results of the testing data of ISPRS Vaihingen Challenge corpus between each class with our proposed techniques in terms of *AverageAccuracy*.

| | Model | IS | Buildings | LV | Tree | Car |
|---|---|---|---|---|---|---|
| **Baseline** | DCED [31–33] | 0.9590 | 0.9778 | 0.9108 | 0.9805 | 0.6832 |
| **Proposed Method** | GCN50 [15] | 0.9595 | 0.9628 | 0.9403 | 0.9896 | 0.7292 |
| | GCN101 | 0.9652 | 0.9827 | **0.9615** | 0.9797 | 0.7387 |
| | GCN152 | 0.9543 | **0.9962** | 0.9445 | 0.9754 | 0.7710 |
| | **GCN152-A** | 0.9614 | 0.9865 | 0.9554 | 0.9871 | 0.8181 |
| | **GCN152-TL-A** | **0.9664** | 0.9700 | 0.9499 | **0.9901** | **0.8567** |

#### 5.2.1. Effect of the Enhanced GCN on the ISPRS Vaihingen Corpus

Our first strategy aims to increase the *F*1 and *mean IoU* score of the network by varying backbones using ResNet 50, ResNet 101, and ResNet 152 rather than the traditional one, the DCED method. From Tables 4 and 5, the *F*1 of GCN152 (0.7864) outperforms that of GCN50 (0.776), GCN101 (0.768), and the baseline method, DCED (0.7693); this yields a higher *F*1 at 0.02%, 0.68%, and 1.01%, respectively. The *mean IoU* of GCN152 (0.8977) outperforms that of GCN50 (0.8776), GCN101 (0.8972), and the baseline method, DCED (0.8651); this yields a higher *mean IoU* at 0.02%, 0.68%, and 1.01% respectively. This can imply that an enhanced GCN is also more accurate than the DCED approach on a very high resolution dataset. ResNet with a large number of layers is still more robust than a small number of layers, the same as that performed on the Landsat-8 corpus (Section 5.1.1).

When comparing the results between the original GCN method and the enhanced GCN methods on the Landsat-8 corpus (Table 4), it is clear that the GCN with a larger backbone layer can improve network performance in terms of *F*1 and *mean IoU*

#### 5.2.2. Effect of Using Channel Attention on ISPRS Vaihingen Corpus

Our second mechanism focused on utilizing the channel attention block to change the weights of the features on each stage to enhance the consistency. From Tables 4 and 5, the *F*1 of GCN152-A (0.7902) is greater than that of GCN152 (0.7864); this yields a higher *F*1 score at 0.38%. The *mean IoU* of GCN152-A (0.9057) is better than that of GCN152 (0.8977); this yields a higher *mean IoU* score at 0.80%. The results (Figures 13e and 14e) show that this can also cause the network to obtain discriminative features stage-wise to make intra-class prediction consistent with respect to very high resolution images.

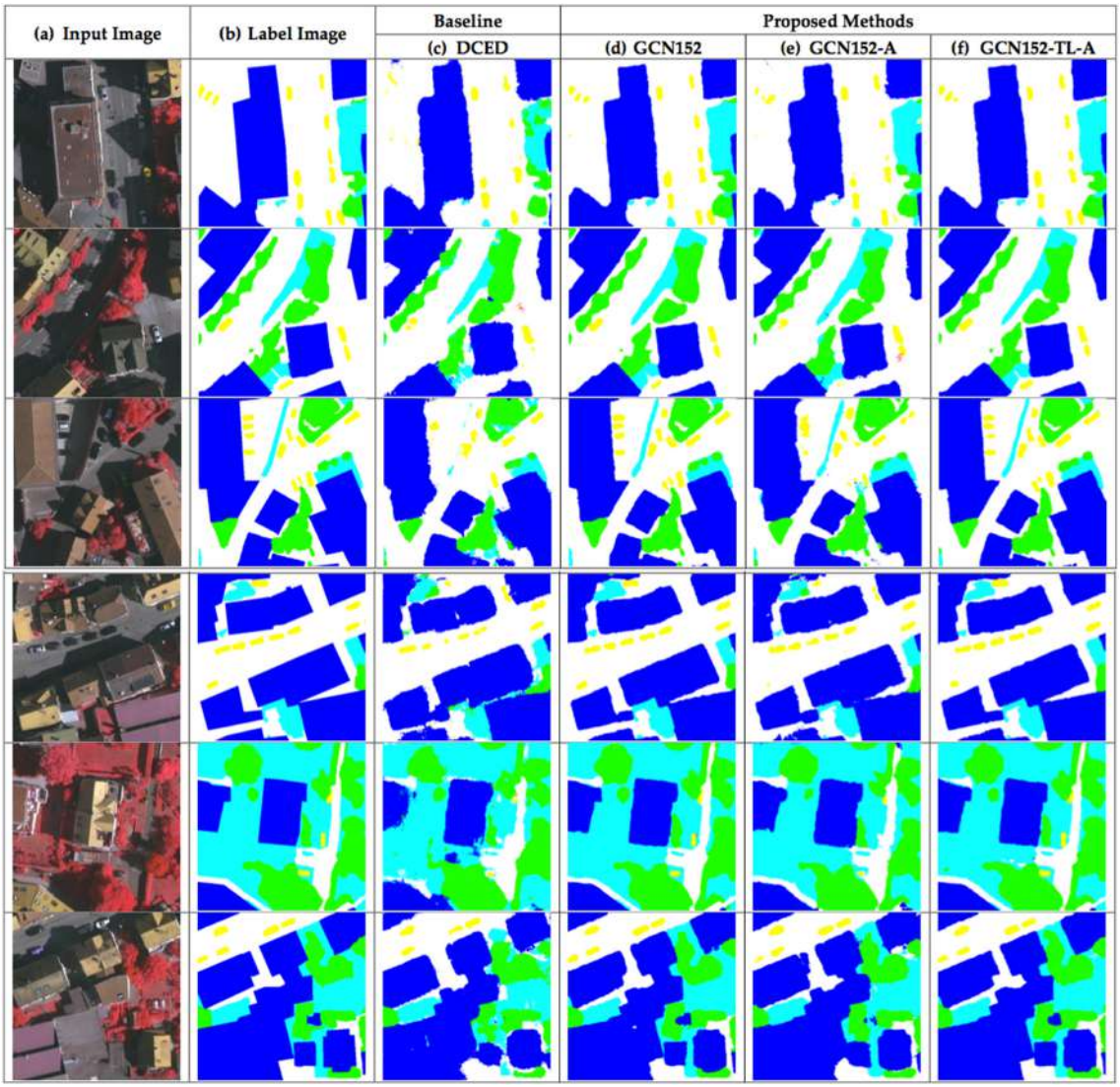



**Figure 13.** Six testing sample input and output aerial images on ISPRS Vaihingen Challenge corpus, where rows refer different images. (**a**) Original input image. (**b**) Target map (ground truth). (**c**) Output of Encoder–Decoder (Baseline). (**d**) Output of GCN152. (**e**) Output of GCN152-A. and (**f**) Output of GCN152-TL-A. The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow) and clutter/background (red).

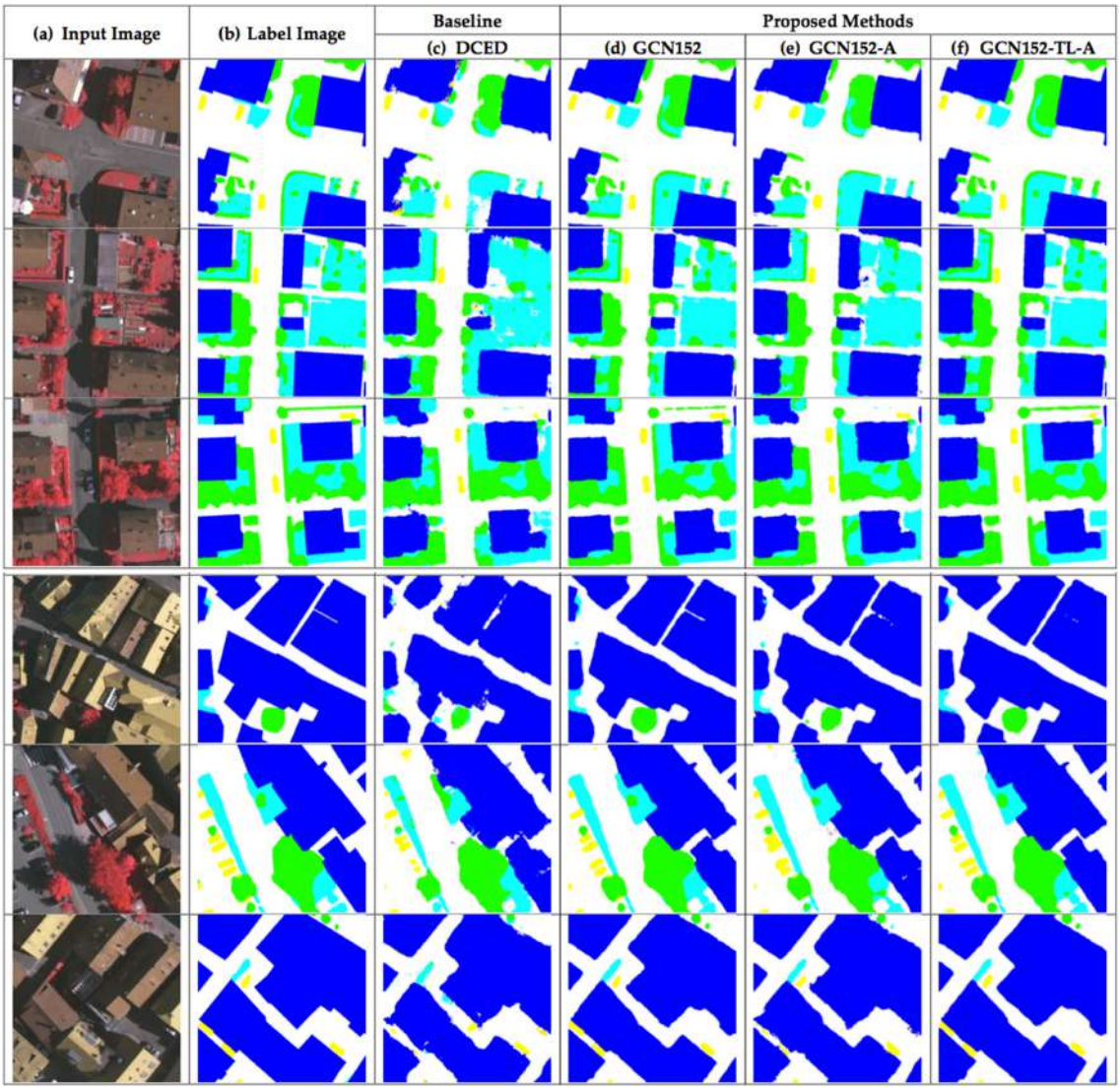



**Figure 14.** Six testing sample input and output aerial images on ISPRS Vaihingen Challenge corpus, where rows refer different images. (**a**) Original input image; (**b**) Target map (ground truth); (**c**) Output of Encoder–Decoder (Baseline); (**d**) Output of GCN152; (**e**) Output of GCN152-A; and (**f**) Output of GCN152-TL-A. The label of the Vaihingen Challenge includes six categories: impervious surface (imp surf, white), building (blue), low vegetation (low veg, cyan), tree (green), car (yellow), and clutter/background (red).

#### 5.2.3. The Effect of Using Domain-Specific Transfer Learning on the ISPRS Vaihingen Corpus

Our last strategy aims to perform domain-specific transfer learning (details in Section 3.3) by reusing the pre-trained weight from the GCN152-A model on the Landsat-8 corpus. From Tables 4 and 5, the $F1$ of the GCN152-TL-A method is the winner; it clearly outperforms not only the baseline but also all previous generations. Its $F1$ is higher than the DCED (baseline) at 2.49% and 1.82% consecutively. Its *mean IoU* is higher than the DCED and the GCN at 4.76% and 3.51%, respectively. Additionally, the result illustrates that the concept of domain-specific transfer learning can enhance both precision (0.7888) and recall (0.8001).

Figures 13 and 14 shows 12 sample results from the proposed method. By applying all strategies, the images in the last column (Figures 13f and 14f) are similar to ground truths (Figures 13b and 14b). Furthermore, $F1$ results and *mean IoU* scores are improved for each strategy we added to the network as shown in Figures 13c–f and 14c–f.

To further evaluate the effectiveness of the proposed GCN152-TL-A comparisons with the baseline method on the one challenging benchmark and the one private benchmark are presented in Tables 2 and 3 for the Landsat-8 dataset with respect to Nan (Thailand) and Tables 4 and 5 for the Vaihengen dataset. All extensive experiments on the Landsat-8 and ISPRS datasets demonstrate that the proposed method clearly achieves promising gains compared with the baseline approach.

Figures 13 and 14 show twelve sample testing results from the proposed method on ISPRS Vaihingen corpus. The results of the last column are also similar to the ground truth in the second column same as performed on Landsat-8 corpus. Considering to each class (are shown in Tables 3 and 5), almost every classes (three out of five) from our proposed methods are the winner in term *Average Accuracy*.

As can be seen in Figures 13 and 14, the performance of our best model outperforms other advanced models by a considerable margin on each category, especially for the impervious surface (IS), tree, and car categories. To show the effectiveness of the proposed methods, we performed comparisons against a number of state-of-the-art semantic segmentation methods, as listed in Table 4, Table 5 with respect to the ISPRS corpus, and Tables 2 and 3 with respect to the Landsat-8 corpus. The DCED [31–33] and GCN [15] are the versions with ResNet-50 as their backbone. In particular, we re-implemented the DCED with Tensorflow-Slim [36], since the released code was built on Caffe [38]. We can see that our proposed methods significantly outperform other methods on both the *F*1 *score* and *mean IoU*.

In terms of the computational cost, our framework requires slightly additional training time compared to the baseline approach, DCED, by about 6.25% (6–7 h), and GCN, by about 4.5% (4–5 h). In our experiment, DCED's training procedure took approximately 16 h per dataset, and finished after 50 epochs with 1152 s per epoch. Our framework is a modification of the GCN-based deep learning architecture. The channel attention model increases the time by 20 min compared with the GCN152 method. There is no additional time required when reusing pre-trained weights.

## 6. Conclusions and Future Work

In this study, we propose a novel CNN framework to perform semantic labeling on remotely sensed images. Our proposed method achieves excellent performance by presenting three aspects. First, a global convolutional network (GCN) is employed and enhanced by adding larger numbers of layers to better capture complex features. Second, channel attention is proposed to assign a proper weight for each extracted feature on different stages of the network. Finally, domain-specific transfer learning is introduced to allay the scarcity issue by training the initial weights using other remotely sensed corpora whose resolutions can be different. The experiments were conducted on two datasets: Landsat-8 (medium resolution) and the ISPRS Vaihingen Challenge (very hign resolution) datasets. The results show that our model that combines all proposed strategies outperforms baseline models in terms of *F*1 and mean IoU. The final results show that our enhanced GCN outperforms the baseline (DCED)—17.48% for F1 on the Landsat-8 corpus and 2.48% on the ISPRS corpus.

In the future, more choices of semantic labeling, modern optimization techniques, and/or other novel activation functions will be investigated and compared to obtain the best GCN-based framework for semantic segmentation in remotely sensed images. Moreover, incorporating other data sources (e.g., a digital surface model) might be needed to increase the accuracy of deep learning for both the CNN and the modern deep learning layer with very low confidence simultaneously. These aforementioned issues will be investigated in future research.

**Author Contributions:** T.P. performed all the experiments and wrote the paper; P.V. and T.P. performed the results analysis and edited the manuscript. K.J., S.L., T.P. and P.S. reviewed the results. T.P. revised the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** T. Panboonyuen thanks the scholarship from The 100th Anniversary Chulalongkorn University Fund granted and The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund). We greatly acknowledge Geo-informatics and Space Technology Development Agency (GISTDA), Thailand, for providing satellite imagery used in this study and T. Panboonyuen thanks to the staff from the GISTDA (Thanwarat Anan, Suwalak Nakya, Bussakon Satta) for the supply of LANDSAT-8 imagery and supporting ground data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| BR | Boundary Refinement |
| CNN | Convolutional Neural Network |
| DCED | Deep Convolutional Encoder–Decoder |
| GCN | Global Convolutional Network |
| MR | Medium Resolution |
| RGB | Red–Green–Blue |
| LS | Landsat |
| TL | Transfer Learning |
| VHR | Very High Resolution |

## References

1. Liu, Y.; Fan, B.; Wang, L.; Bai, J.; Xiang, S.; Pan, C. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 78–95. [CrossRef]
2. Wang, H.; Wang, Y.; Zhang, Q.; Xiang, S.; Pan, C. Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sens.* **2017**, *9*, 446. [CrossRef]
3. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
4. Panboonyuen, T.; Vateekul, P.; Jitkajornwanich, K.; Lawawirojwong, S. An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery. In *Recent Advances in Information and Communication Technology Series*; Springer: Cham, Switzerland, 2017; Volume 566.
5. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv* **2016**, arXiv:1606.00915.
6. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
8. Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Road Segmentation of Remotely-Sensed Images Using Deep Convolutional Neural Networks with Landscape Metrics and Conditional Random Fields. *Remote Sens.* **2017**, *9*, 680. [CrossRef]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference onComputer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
10. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
11. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
12. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651.
13. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.

15. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1743–1751.
16. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a Discriminative Feature Network for Semantic Segmentation. *arXiv* **2018**, arXiv:1804.09337.
17. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *arXiv* **2017**, arXiv:1709.01507.
18. Xie, M.; Jean, N.; Burke, M.; Lobell, D.; Ermon, S. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv* **2015**, arXiv:1510.00098.
19. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 3320–3328.
20. Liu, J.; Wang, Y.; Qiao, Y. Sparse Deep Transfer Learning for Convolutional Neural Network. In Proceedings of the AAAI, San Francisco, CA, USA, 4–9 February 2017; pp. 2245–2251.
21. International Society for Photogrammetry and Remote Sensing. 2D Semantic Labeling Challenge. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 9 September 2018).
22. Valada, A.; Vertens, J.; Dhall, A.; Burgard, W. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4644–4651.
23. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder–decoder with atrous separable convolution for semantic image segmentation. *arXiv* **2018**, arXiv:1802.02611.
24. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *arXiv* **2018**, arXiv:1808.00897.
25. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
26. Bilinski, P.; Prisacariu, V. Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6596–6605.
27. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3684–3692.
28. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. *arXiv* **2016**, arXiv:1611.07709.
29. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images. *arXiv* **2017**, arXiv:1704.08545.
30. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
31. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
32. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *arXiv* **2015**, arXiv:1511.00561.
33. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder–decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
34. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
35. Barsi, J.A.; Lee, K.; Kvaran, G.; Markham, B.L.; Pedelty, J.A. The spectral response of the Landsat-8 operational land imager. *Remote Sens.* **2014**, *6*, 10232–10251. [CrossRef]

36. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
38. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.

*remote sensing*

MDPI

*Article*

# Road Segmentation of Remotely-Sensed Images Using Deep Convolutional Neural Networks with Landscape Metrics and Conditional Random Fields

**Teerapong Panboonyuen [1], Kulsawasd Jitkajornwanich [2], Siam Lawawirojwong [3], Panu Srestasathiern [3] and Peerapon Vateekul [1,*]**

1 Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Phayathai Rd., Pathumwan, Bangkok 10330, Thailand; teerapong.pan@student.chula.ac.th

2 Data Science and Computational Intelligence (DSCI) Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd., Ladkrabang, Bangkok 10520, Thailand; kulsawasd.ji@kmitl.ac.th

3 Geo-Informatics and Space Technology Development Agency (Public Organization), 120, The Government Complex, Chaeng Wattana Rd., Lak Si, Bangkok 10210, Thailand; siam@gistda.or.th (S.L.); panu@gistda.or.th (P.S.)

* Correspondence: peerapon.v@chula.ac.th; Tel.: +6-62-218-6989

Academic Editors: Qi Wang, Nicolas H. Younan, Carlos López-Martínez and Prasad S. Thenkabail
Received: 1 June 2017; Accepted: 26 June 2017; Published: 1 July 2017

**Abstract:** Object segmentation of remotely-sensed aerial (or very-high resolution, VHS) images and satellite (or high-resolution, HR) images, has been applied to many application domains, especially in road extraction in which the segmented objects are served as a mandatory layer in geospatial databases. Several attempts at applying the deep convolutional neural network (DCNN) to extract roads from remote sensing images have been made; however, the accuracy is still limited. In this paper, we present an enhanced DCNN framework specifically tailored for road extraction of remote sensing images by applying landscape metrics (LMs) and conditional random fields (CRFs). To improve the DCNN, a modern activation function called the exponential linear unit (ELU), is employed in our network, resulting in a higher number of, and yet more accurate, extracted roads. To further reduce falsely classified road objects, a solution based on an adoption of LMs is proposed. Finally, to sharpen the extracted roads, a CRF method is added to our framework. The experiments were conducted on Massachusetts road aerial imagery as well as the Thailand Earth Observation System (THEOS) satellite imagery data sets. The results showed that our proposed framework outperformed Segnet, a state-of-the-art object segmentation technique, on any kinds of remote sensing imagery, in most of the cases in terms of *precision*, *recall*, and *F*1.

**Keywords:** deep convolutional neural networks; road segmentation; conditional random fields; satellite images; aerial images; THEOS

## 1. Introduction

Extraction of terrestrial objects such as buildings and roads, from remotely-sensed images has been employed in many applications in various areas, e.g., urban planning, map updates, route optimization, and navigation. For road extraction, most primary research is based on unsupervised learning, such as graph cut and global optimization techniques [1]. These unsupervised methods, however; have one common limitation, color-sensitivity, since they rely on only the color features.

That is, the segmentation algorithms will not perform well if the road colors presented in the suburban remotely-sensed images contain more than one color (e.g., yellowish brown roads in the countryside regions and cement-grayed roads in the suburban regions). This, in fact, has become a motivation of this work, that is, to overcome the color sensitivity issues.

Deep learning, a large convolutional neural network with performance that can be scaled depending on the size of training data and model complexity as well as processing power, has shown significant improvements in object segmentation from images as seen in many recent works [2–13]. Unlike unsupervised learning, more than one feature—other than color—can be extracted: line, shape, and texture, among others. The traditional deep learning methods such as the deep convolutional neural network (DCNN) [3,14], deep deconvolutional neural network (DeCNN) [5], recurrent neural network, namely reSeg [15], and fully convolutional networks [4]; however all suffer from accuracy performance issues.

A deep convolutional encoder-decoder (DCED) architecture, one of the most efficient newly developed neural networks, has been proposed for object segmentation. The DCED network is designed to be a core segmentation engine for pixel-wise semantic segmentation, and has shown good performance in the experiments tested using PASCAL VOC 2012 data—a well-known benchmark data set for image segmentation research [6,8,16]. In this architecture, the rectified linear unit (ReLU) is employed as an activation function.

In the road extraction task, there are many issues that can cause limited detection performance. First, based on [6,8], although the most recent DCED approach for object segmentation (or SegNet) showed promising detection performance on overall classes, the result for road objects is still limited as it fails to detect many road objects. This could be caused by the rectified linear unit (ReLU) which is sensitive to the gradient vanishing problem. Second, even when we apply Gaussian smoothing at the last step to connect detected roads together, this still yields excessive detected road objects (false road objects).

In this paper, we present an improved deep convolutional encoder-decoder network (DCED) for segmenting road objects from aerial and satellite images. Several aspects of the proposed method are enhanced, including incorporation of exponential linear units (ELUs), as opposed to ReLUs that typically outperform ELU in most object classification cases; adoption of landscape metrics (LMs) to further improve the overall quality of results by removing falsely detected road objects; and lastly, combination with the traditional fully-connected conditional random field (CRF) algorithms used in semantic segmentation problems. Although the ELU-SegNet-LM network may suffer a performance issue due to the loss of spatial accuracy, it can be alleviated by the conditional random fields algorithm, which takes into account the low-level information captured by the local interactions of pixels and edges [17–19]. The experiments were conducted using well-known aerial imagery, a Massachusetts roads data set (Mass. Roads), which is publicly available, and satellite imagery (from the Thailand Earth Observation System (THEOS) satellite) which is provided by GISTDA. The results showed that our method outperforms all of the baselines including SegNet in terms of *precision*, *recall*, and $F1$ scores. The paper is organized as follows. Related work is discussed in Section 2. Section 3 describes our proposed methodology. Experimental data sets and evaluations are described in Section 4. Experimental results and discussions are presented in Section 5. Finally, we conclude our work and discuss future work in Section 6.

## 2. Related Work

Deep learning is one of the fast-growing fields in machine learning which has been successfully applied to remotely-sensed data analysis, notably land cover mapping on urban areas [20]. It has increasingly become a promising tool for accelerating image recognition process with high accuracy results [4], [6], [21]; new architectures are proposed constantly on a weekly basis. This related work is divided into three subsections: we first discuss deep learning concepts for semantic segmentation,

followed by a set of road object segmentation techniques using deep learning, and finally activation functions and post processing technique of deep learning are discussed.

Note that this paper only focuses on approaches built around deep learning techniques. Therefore, prior attempts at semantic segmentation [22,23] are not included and compared here since they are not based on a deep learning approach.

### *2.1. Deep Learning for Semantic Segmentation*

Semantic segmentation algorithms are often formulated to solve structured pixel-wise labeling problems based on the deep convolutional neural network (DCNN), and are state-of-the-art supervised learning algorithms for modeling and extracting latent feature hierarchies. Noh et al. [5] proposed a novel semantic segmentation technique utilizing a deconvolutional neural network (DeCNN) and the top layer from DCNN adopted from VGG16 [24]. The DeCNN structure is composed of upsampling layers and deconvolution layers, describing pixel-wise class labels and predicting segmentation masks, respectively. Their proposed deep learning methods yield high performance in the PASCAL VOC 2012 data set [16], with a 72.5% accuracy in the best case scenario (this was the highest accuracy—at the time of writing this paper—compared to other methods that were trained without requiring additional or external data). Long et al. [4] proposed an adapted contemporary classification network incorporating Alex, VGG and Google networks into a full DCNN. In this method, some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s). The skip architecture reduces the potential over-fitting problem and has shown improvements in performance ranging from 20 to 62.2% in the experiments tested using PASCAL VOC 2012 data. Ronneberger et al. [12] proposed U-Net, a DCNN for biomedical image segmentation. The architecture consists of a contracting path and a symmetric expanding path that capture context and consequently, enable precise localization. The proposed network claimed to be capable of learning despite the limited number of training images, and performed better than the prior best method (a sliding-window DCNN) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. In this work, VGG16 is selected as our baseline architecture since it is the most popular architecture used in various networks for object recognition. Furthermore, we will investigate the effect of the skipped layer technique, especially FCN-8s, since it is the top-ranking architecture as shown in Long et al. [4].

There is a new research area called "instance-aware semantic segmentation" which is slightly different from "semantic segmentation." Instead of labeling all pixels, it focuses on the target objects and labels only pixels of those objects. FCIS [25] is a technique developed based on fully convolutional networks (FCN). Mask R-CNN [26] is also created on top of FCN but incorporates with a proposed joint formulation. Even though their results are promising, they are not directly related to our scope on "semantic segmentation." In the future, we can extend these works and compare them to our proposed technique.

### *2.2. Deep Learning for Road Segmentation*

There are many approaches to road network extraction in very-high-resolution (VHR) aerial and satellite imagery literature. Wand et al. [14] proposed a DCNN and finite state machine (FSM)-based framework to extract road networks from aerial and satellite images. DCNN recognizes patterns from a sophisticated and arbitrary environment while FSM translates the recognized patterns to states such that their tracking behaviors can be captured. The results showed that their approach is more accurate compared to the traditional methods. The extension of the method for automatic road point initialization was left for future work. DCNN for multiple object extraction from aerial imagery was proposed in [3] by Saito et al. Both features (extractors and classifiers) of DCNN were automated in that a new technique to train a single DCNN for extracting multiple kinds of objects simultaneously was developed. Two objects were extracted: buildings and roads, thus a label image consists of three channels: buildings, roads, and background. Finally, the results showed

that the proposed technique not only improved the prediction performance but also outperformed the cutting-edge method tested on a publicly available aerial imagery data set. Muruganandham et al. [2] designed an automated framework to extract semantic maps of roads and highways, so the urban growth of cities from remote sensing images could be tracked. They used the VGG16 model—a simplistic architecture with homogeneous $3 \times 3$ convolution kernels and $2 \times 2$ max pooling throughout the pipeline—as a baseline for a fixed feature extractor. The experimental results showed that their proposed technique for the prediction performance was improved with $F1$ scores of 0.76 on the Mass. Roads data set.

### *2.3. Recent Techniques in Deep Learning*

Activation function is an important factor for the accuracy of DCNN. While the most popular activation function for neural networks is the rectified linear unit (ReLU), Clevert et al. [21] have just proposed the exponential linear unit (ELU), which can speed up the learning process in DCNN and therefore lead to higher classification accuracies as well as overcoming the previously unsolvable problem, i.e., the vanishing gradient problem. Compared to other methods with different activation functions, ELU has greatly improved many of the learning characteristics. In the experiments, ELUs enable fast learning as well as more effective generalization performance than the ReLUs and the leaky rectified linear units (LReLUs) in networks with five layers or more. In ImageNet, ELU networks substantially increased the learning time compared to ReLU networks with the identical architecture; less than 10% classification error was presented for a single crop, model network.

Recently, there have been some efforts to enhance the performance of DCNN by combining it with other classifier as a post-processing step. Conditional random fields (CRFs) has been reported successful in increasing the accuracy of DCNN, especially in the image segmentation domain. CRFs have been employed to smooth maps [7,17–19]. Typically these models contain energy terms that couple neighboring nodes, favoring same-label assignments to spatially proximal pixels. Qualitatively, the primary function of these short-range CRFs has been used to clean up the spurious predictions of weak classifiers built on top of local hand-engineered features.

## 3. Proposed Methodology

In this section, we propose an enhanced, improved DCED network (or SegNet) to efficiently segment road objects from aerial and satellite images. Three aspects of the proposed method are enhanced: (**1**) modification of DCED architecture; (**2**) incorporation of landscape metrics (LMs); and (**3**) adoption of conditional random fields (CRFs). An overview of our proposed method is shown in Figure 1.


**Figure 1.** A process in our proposed framework.

### *3.1. Data Preprocessing*

Data preparation is required when working with neural network and deep learning models. In addition, data augmentation is often required in more complex object recognition tasks. Thus, we increased the size of our data sets to improve the method efficiency by rotating them incrementally with eight different angles. All images on Massachusetts road data sets are standardized and cropped into $1500 \times 1500$ pixels with a resolution of 1 $m^2$/pixel. The data sets consist of 1108 training images, 49 test images, and 14 validation images. The original training images were further extended to 8864 training images.

On the THEOS data sets, we also increased the size of data sets in a similar fashion. Each image has 1500 × 1500 pixels with a resolution of 2 $m^2$/pixel.

### *3.2. Object Segmentation (ELU-SegNet)*

SegNet, one of the deep convolutional encoder-decoder architectures, consists of two main networks encoder and decoder, and some outer layers. The two outer layers of the decoder network are responsible for feature extraction task, the results of which are transmitted to the next layer adjacent to the last layer of the decoder network. This layer is responsible for pixel-wise classification (determining which pixel belongs to which class). There is no fully connected layer in between feature extraction layers. In the upsampling layer of decoder, pool indices from encoder are distributed to the decoder where the kernel will be trained in each epoch (training round) at the convolution layer. In the last layer (classification), softmax is used as a classifier for pixel-wise classification. The encoder network consists of convolution layer and pooling layer. A technique, called batch normalization (proposed by Ioffe and Szegedy [27]), is used to speed up the learning process of the DCNN by reducing internal covariate shift. In the encoder network, the number of layers is reduced to 13 (VGG16) by removing the last three layers (fully connected layers) [6,8,28,29] for the following two reasons: to maintain the high-resolution feature maps in the encoder network, and to minimize the countless number of parameters from 134 million features to 14.7 million features compared to the traditional deep learning networks such as DCNN [4] and DeCNN [5], where the fully connected layer remains intact. In the activation function of feature extraction, ReLU, max-pooling, and 7 × 7 kernels are used in both encoder and decoder networks. For training images, three-channel images (RGB) are used. The exponential linear unit (ELU) was introduced in [21], which can speed up learning in deep neural networks, offer higher classification accuracies, and give better generalization performance than ReLUs and LReLUs on networks. In SegNet architecture, to perform optimization for training networks,the stochastic gradient descent (SGD) [30] with a fixed learning rate of 0.1 and momentum of 0.9 is used. In each training round (epoch), a mini-batch (a set of 12 images) is chosen such that each image is used once. The model with the best performance on the validation data set in each epoch will be selected. Our architecture (see Figure 2) is enhanced from SegNet, consisting of two main networks responsible for feature extraction. In each network, there are 13 layers, with the last layer being the classification based on softmax supporting pixel-wise classification.

In our work, an activation function called ELU is used as opposed to ReLU based on its performances. For the network training optimization, stochastic gradient descent (SGD) is used and configured with a fixed learning rate of 0.001 and momentum of 0.9 to delay the convergence time and so, can avoid local optimization trap.

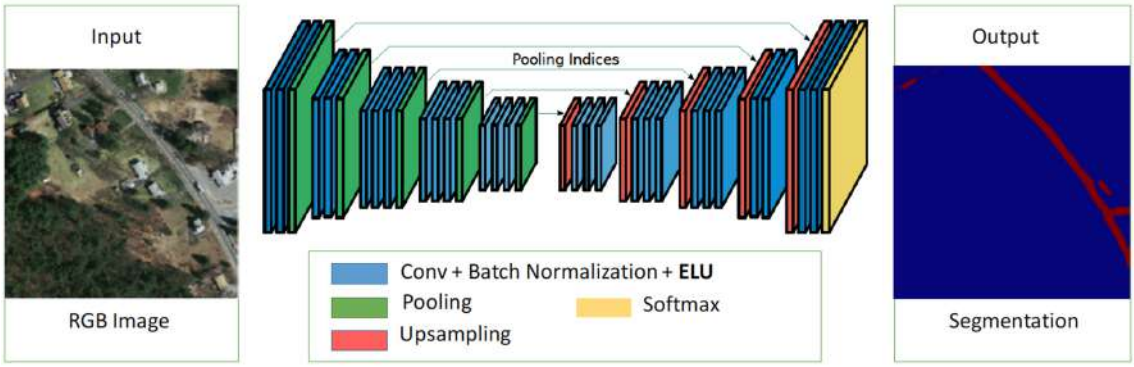



**Figure 2.** A proposed network architecture for object segmentation (exponential linear unit (ELU)-SegNet).

### *3.3. Gaussian Smoothing*

Gaussian smoothing [31] is a 2-D convolution operator that is used to 'blur' images and remove unnecessary details and noises by utilizing the Gaussian function. The Gaussian function is used to determine the transformation needed for each pixel, resulting in a more complete extended road objects. We applied the Gaussian function first in the post-processing step in order to expand and prepare objects that are close to each other to be combined into components in the next step (as we shall see in Section 3.4).

The 1-D and 2-D Gaussian functions are described in Equations (1) and (2), respectively.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \tag{1}$$

$$G(x) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2-y^2}{2\sigma^2}} \tag{2}$$

where $x$ represents the distance from the origin in the $X$-axis, $y$ represents the distance from the origin in the $Y$-axis, and $\sigma$ represent the standard deviation of the Gaussian distribution.

### *3.4. Connected Component Labeling (CCL)*

In connected components labeling (CCL) [31], all pixels are scanned and adjacent pixels with similar connectivity values are combined. Eight neighbors of each pixel were considered when analyzing connected components.

The expanded and overlapped objects from the Gaussian smoothing were actually grouped together in this step. The labeled objects will be further calculated using geometric attributes (e.g., area and perimeter) based on landscape metrics (LMs) as described in the next section.

### *3.5. False Road Object Removal (LMs)*

After smoothing and labeling the objects, we compute the shape complexity of the objects through the shape index (as seen in Equation (3)), one of the landscape metrics for measuring arrangement and composition property of spatial objects. The resulting objects along with their shape scores are shown in Figure 3. As seen in Figure 3, the geometrical characteristics of roads were captured and differentiated from other spatial objects in the given image. Other geometry metrics can also be used such as rectangular degree, aspect ratio, etc. More information on other landscape metrics can be found in [32,33].

$$shape\ index = \frac{e(i)}{4x\sqrt{A(i)}} \tag{3}$$

where $e(i)$ and $A(i)$ denote the perimeter and area for object $i$, respectively.

### *3.6. Road Object Sharpening (CRFs)*

Conditional random fields (CRFs) have traditionally been implemented to sharpen noisy segmentation maps [18]. These models are generally composed of energy terms comprising nodes in the neighborhood, causing false assignments of pixels that are in close proximity. To resolve these spatial limitations of short-range CRFs, the fully connected CRFs are integrated into our system [19]. Equation (4) expresses the energy function of the dense CRFs.

In the last step, we extended the ELU-SegNet-LMs model to ELU-SegNet-LMs-CRFs to enhance the network performance by adding explicit dependencies among the neural network outputs. Particularly, we added smoothness terms between neighboring pixels to our model, which can eliminate the need to learn smoothness from remotely-sensed images. Using the resulting models

as part of the post-processing significantly increases the overall performance of the network over unstructured deep neural networks.

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{4}$$

where $x$ denotes the label assignment for pixels. A unary potential used is $\theta_i(x_i)) = -logP(x_i)$, while $P(x_i)$ denotes the label assignment probability at pixel $i$ as computed by a DCNN.

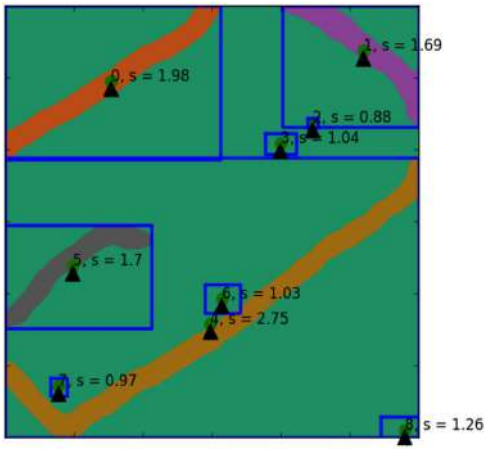



**Figure 3.** Illustration of shape index scores on each extracted road object. Any objects with shape index score lower than 1.25 are considered as noises and subsequently removed.

The inference can be efficiently established in the pair-wise potentials when using the fully connected graph. We treated the unary potential as local classifiers which are defined by the output of the ELU-SegNet-LMs model, which is a probability map for each class in each of the pixels. The pairwise potentials depict the interaction of pixels in the neighborhood and are influenced by the color similarity. In the DeepLab CRF model [19], the dense CRFs (instead of neighboring information) are used as a means to identify relationships between pixels. Furthermore, they define the following pairwise potentials as shown in Equation (5).

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)[w_1 \exp(-\frac{\| p_i - p_j \|^2}{2\sigma_\alpha^2} - \frac{\| I_i - I_j \|^2}{2\sigma_\beta^2}) + w_2 \exp(-\frac{\| p_i - p_j \|^2}{2\sigma_\gamma^2})] \tag{5}$$

where $\mu(x_i, x_j) = 1 \; if \; x_i \neq x_j$ and zero otherwise, which, as in the Potts model, means that only nodes with distinct labels are penalized. The remaining expression uses two Gaussian kernels in different feature spaces; the first, 'bilateral' kernel depends on both pixel positions (denoted as $p$) and red-green-blue (RGB) color (denoted as $I$), and the second kernel only depends on pixel positions. The hyperparameters $\sigma_\alpha$, $\sigma_\beta$ *and* $\sigma_\gamma$ control the scale of Gaussian kernels. The first kernel forces pixels to similar color and position to have similar labels, while the second kernel only considers spatial proximity when enforcing smoothness.

In summary, the first term of pairwise potentials depends on both pixel positions and color intensities whereas the second term depends solely on the pixel positions [18,19]. Although the dense CRFs can have billions of edges (which is technically infeasible to solve), it was recently found that the inference/maximum posterior can be approximated by the mean-field algorithm.

## 4. Experimental Data Sets and Evaluation

In our experiments, two types of data sets are used: aerial images and satellite images. Table 1 shows one aerial data set (Massachusetts) and five satellite data sets (Nakhonpathom,

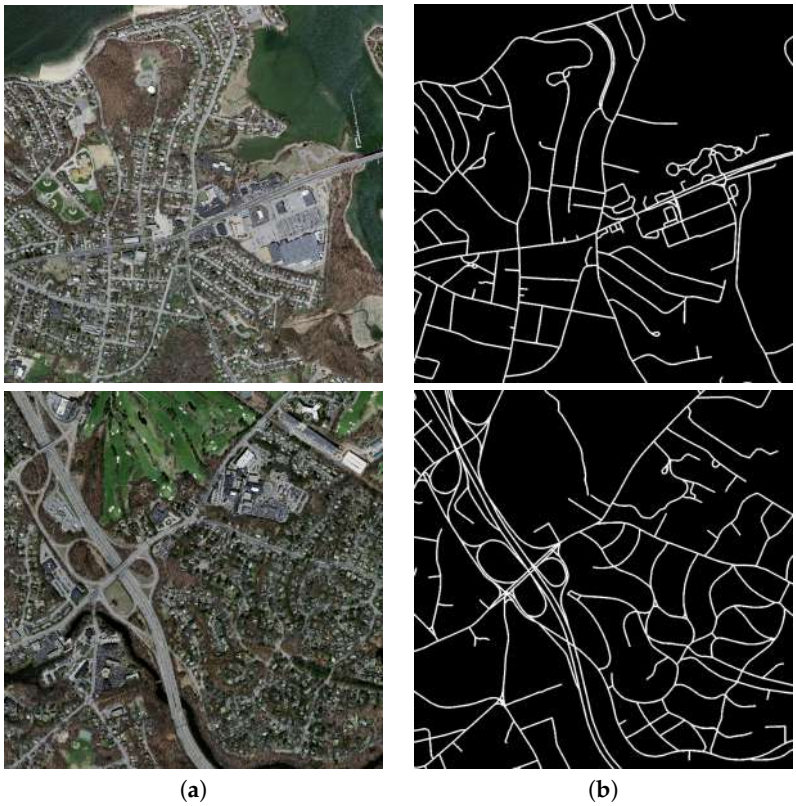
Chonburi, Songkhla, Surin, and Ubonratchathani). All experiments are evaluated based on *precision*, *recall*, and *F*1.

**Table 1.** Numbers of training, validation, and testing sets.

| | Training Set | Validation Set | Testing Set |
|---|---|---|---|
| **Massachusetts** | 1108 | 14 | 49 |
| **Nakhonpathom** | 200 | 14 | 49 |
| **Chonburi** | 100 | 14 | 49 |
| **Songkhla** | 100 | 14 | 49 |
| **Surin** | 70 | 14 | 49 |
| **Ubonratchathani** | 70 | 14 | 49 |

### *4.1. Massachusetts Road Data Set (Aerial Imagery)*

This data set (made publicly available by [7]) consists of 1171 aerial images of the state of Massachusetts. Each image is 1500 × 1500 pixels in size, covering an area of 2.25 square kilometers. We randomly split the data into a training set of 1108 images, a validation set of 14 images and a testing set of 49 images. The samples of this data set are shown in Figure 4. The data set covers a wide variety of urban, suburban, and rural regions with a total area of over 2600 square kilometers. With our test set alone, it covers more than 110 square kilometers which is by far the largest and most challenging aerial image labeling data set.



(**a**) (**b**)

**Figure 4.** Two sample aerial images from the Massachusetts road corpus, where a row refers to each image (**a**) Aerial image and (**b**) Binary map, which is a ground truth image denoting the location of roads.

### *4.2. THEOS Data Sets (Satellite Imagery)*

In this type of data, the satellite images were separated into five data sets—one for each province. The datasets were obtained from the Thailand Earth Observation System (THEOS), also known as Thaichote, an Earth observation satellite of Thailand developed by EADS Astrium SAS, France. This data set consists of 855 satellite images covering five provinces: 263 images of Nakhonpathom, 163 images of Chonburi, 163 images of Songkhla, 133 images of Surin, and 133 images of Ubonratchathani. Some samples of these images are shown in Figure 5.

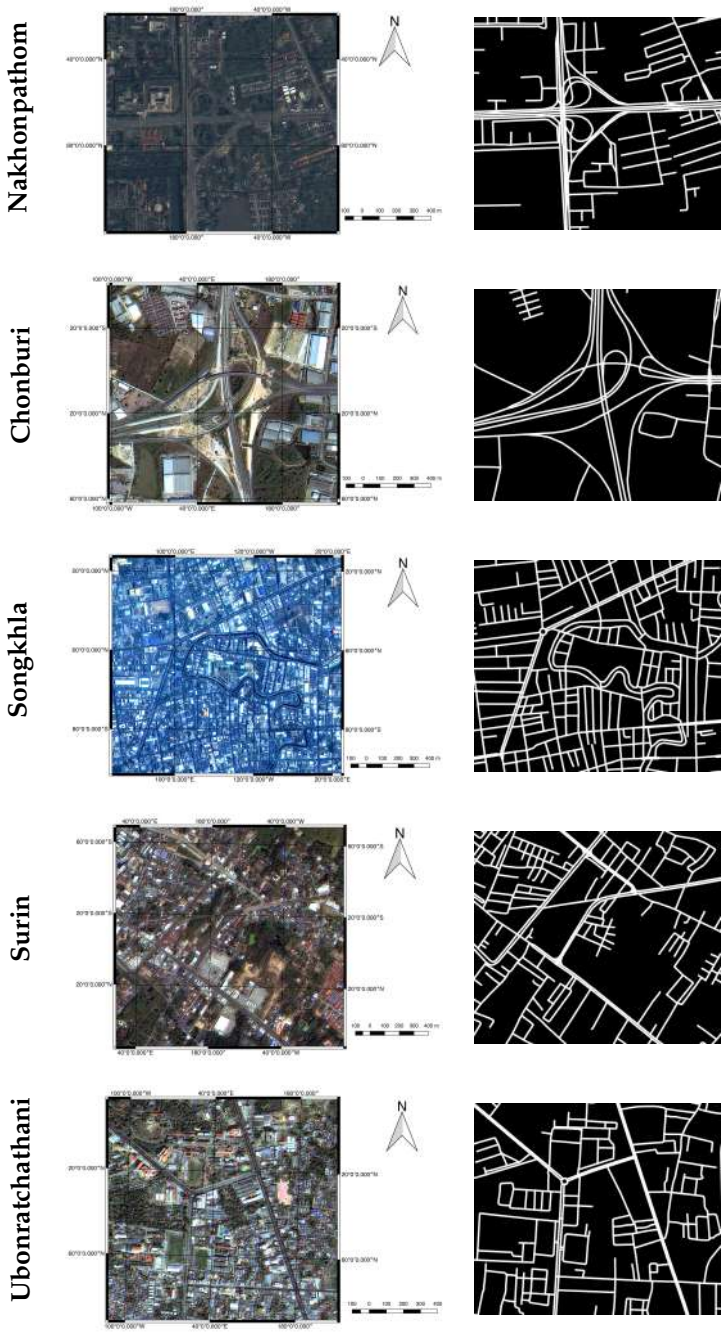


**Figure 5.** Sample satellite images from five provinces of our data sets; each row refers to a single sample image from one province (Nakhonpathom, Chonburi, Songkhla, Surin, and Ubonratchathani) in a satellite image format (**a**) and in a binary map (**b**), which is served as a ground truth image denoting the location of roads.

### *4.3. Evaluation*

The road extraction task can be considered as binary classification, where road pixels are positives and the remaining non-road pixels are negatives. Let TP denote the number of true positives (the number of correctly classified road pixels), TN denote the number of true negatives (the number of correctly classified non-road pixels), FP denote the number of false positives (the number of mistakenly classified road pixels), and FN denote the number of false negatives (the number of mistakenly classified non-road pixels).

The performance measures used are *precision*, *recall*, and $F1$ as shown in Equations (6)–(8). Precision is the percentage of correctly classified road pixels among all predicted pixels by the classifier. Recall is the percentage of correctly classified road pixels among all actual road pixels. $F1$ is a combination of precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{8}$$

## 5. Experimental Results and Discussions

This section illustrates details of our experiments. The proposed deep learning network is based on SegNet with three improvements: (**1**) it employs the ELU activation function; (**2**) it uses LMs to filter incorrect detected roads; and (**3**) it applies CRFs to sharpen broad roads. Thus, there are three variations of the proposed methods as shown in Table 2.

**Table 2.** Variations of our proposed deep learning methods. LM: landscape metric; CRF: conditional random field.

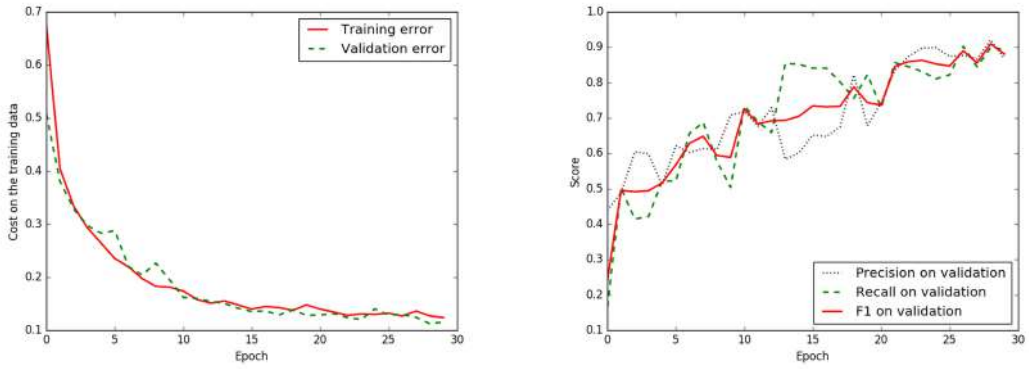
| Abbreviation | Description |
|---|---|
| **ELU**-SegNet | SegNet + **ELU activation** |
| ELU-SegNet-**LMs** | SegNet + ELU activation + **Landscape Metrics** |
| ELU-SegNet-LMs-**CRFs** | SegNet + ELU activation + Landscape Metrics + **CRFs** |

For the experimental setup, there are three experiments on two remotely-sensed data sets: the Massachusetts road data set and THEOS data sets (details in Section 4). The experiments aim to illustrate that each proposed strategy can really improve the performance. First, ELU-SegNet is compared to SegNet for the ELU strategy. Second, ELU-Segnet-LMs is compared to ELU-SegNet for the LM strategy. Third, the full proposed technique (ELU-Segnet-LMs-CRFs) is compared to existing methods for the CRF technique.

The implementation is based on a deep learning framework, called "Lasagne", which is extended from Theano. All experiments were conducted on a server with Intel Core i5-4590S Processor (6M Cache, up to 3.70 GHz), 32 GB of memory, Nvidia GeForce GTX 960 (4 GB), and Nvidia GeForce GTX 1080 (8 GB). Instead of using the whole image (1500 $\times$ 1500 pixels) to train the network, we randomly cropped all images to be 224 $\times$ 224 as inputs of each epoch.

### *5.1. Results on Aerial Imagery (Massachusetts Data Set)*

In this sub-section, the experiment was conducted on the Massachusetts aerial corpus. To achieve the highest accuracy, the network must be configured and trained many epochs until all parameters in the network are converged. Figure 6a illustrates that the proposed network has been properly set and trained until it really is converged. Furthermore, Figure 6b shows that the higher number of epochs tends to show a better *F*1-score. Thus, the number of chosen epochs based on the validation data is 29 (the best model for this data set).




(**a**) (**b**)

**Figure 6.** Iteration plot on Massachusetts aerial corpus of the proposed technique, ELU-SegNet-LMs-CRFs; $x$ refers to epochs and $y$ refers to different measures. (**a**) Plot of model loss (cross entropy) on training and validation data sets, and (**b**) Performance plot on the validation data set.

The result is shown in Table 3 by comparing between baselines and variations of the proposed techniques. It shows that our network with all strategies (ELU-SegNet-LMs-CRFs) outperforms other methods. More details will be discussed to show that each of the proposed techniques can really improve an accuracy. Only in this experiment, there are four baselines, including Basic-model, FCN-no-skip, FCN-8s, and SegNet. Note that SegNet has been implemented and tested on the experimental data set, while the results of other three baselines are carried from the original paper [2].

**Table 3.** Results on the testing data of Massachusetts aerial corpus between four baselines and three variations of our proposed techniques in terms of *precision*, *recall*, and *F*1. FCN: fully convolutional network.

| | Model | Precsion | Recall | F1 |
|---|---|---|---|---|
| **Baselines** | Basic-model [2] | 0.657 | 0.657 | 0.657 |
| | FCN-no-skip [2] | 0.742 | 0.742 | 0.742 |
| | FCN-8s [2] | 0.762 | 0.762 | 0.762 |
| | SegNet | 0.773 | 0.765 | 0.768 |
| **Proposed Method** | **ELU**-SegNet | 0.852 | 0.733 | 0.788 |
| | ELU-SegNet-**LMs** | 0.854 | 0.861 | 0.857 |
| | ELU-SegNet-LMs-**CRFs** | **0.858** | **0.894** | **0.876** |

#### 5.1.1. Results of Enhanced SegNet (ELU-SegNet)

Our first strategy aims to increase an accuracy of the network by using ELU as an activation function (ELU-SegNet) rather than the traditional one, ReLU (SegNet). Details are shown in Section 3.2. From Table 3, *F*1 of ELU-SegNet (0.788) outperforms that of SegNet (0.768); this yields higher *F*1

at 2.6%. The main reason is due to higher *precision*, but slightly lower *recall*. This can imply that ELU is more robust than ReLU to detect road pixels.

#### 5.1.2. Results of Enhanced SegNet with Landscape Metrics (ELU-SegNet-LMs)

Our second mechanism focuses on applying LMs (details in Section 3.5) on top of ELU-SegNet to filter false road objects. From Table 3, the *F*1 of ELU-SegNet-LMs (0.857) is superior to that of ELU-SegNet (0.788) and SegNet (0.768); this yields higher *F*1 at 6.9% and 8.9%, consecutively. Although LM is specifically designed to increase *precision*, the result shows that it can increase both *precision* (0.854) and *recall* (0.861). It is interesting that *recall* is also improved since all noises in the training images have been removed by the LMs filtering technique resulting in a better quality of the training data set.

#### 5.1.3. Results of All Modules (ELU-SegNet-LMs-CRFs)

Our last strategy aims to sharpen road objects (details in Section 3.6) by integrating CRFs into our deep learning network. From Table 3, *F*1 of ELU-SegNet-LMs-CRFs (0.876) is the winner; it clearly outperforms not only the baselines, but also all previous generations. Its *F*1 is higher than SegNet (0.768) at 10.8%. Also, the result illustrates that CRFs can enhance both *precision* (0.858) and *recall* (0.894).

Figure 7 shows two sample results from the proposed method. By applying all strategies, the images in the last column (Figure 7e) look very close to the ground truths (Figure 7b). Furthermore, *F*1-results are improved for each strategy we added to the network as shown in Figure 7c–e.

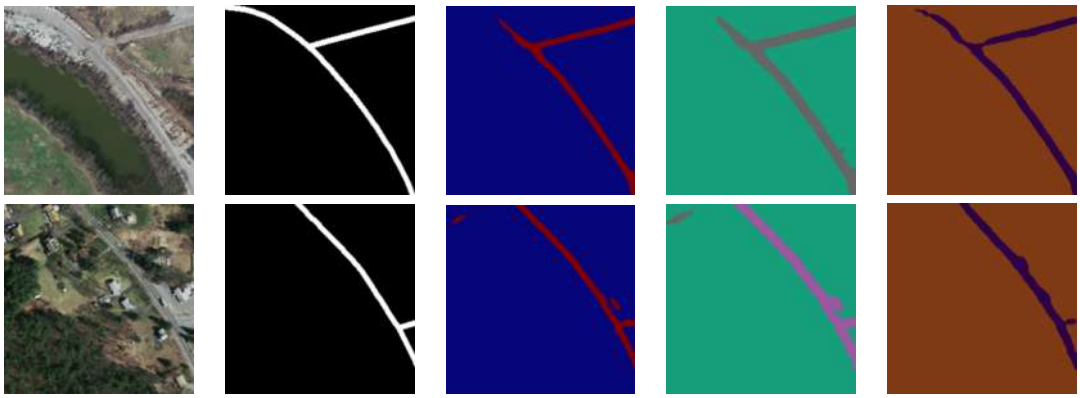



**Figure 7.** Two sample input and output aerial images on Massachusetts corpus, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) Output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

### *5.2. Results for Satellite Imagery (THEOS Data Sets)*

In this sub-section, the experiment was conducted on THEOS satellite images. There are five data sets referring to different provinces: Nakhonpathom, Chonburin, Songkla, Surin, and Ubonratchathani; therefore, there are five learning models. Figure 8 shows that each model is properly set up and trained until it is converged and obtained the best *F*1. The best epochs (models) for each province are 25, 15, 30, 21, and 20, respectively.

The results are shown in Tables 4–6 for measures in terms of *F*1, *precision*, and *recall*, respectively. It is interesting that the proposed network with all strategies (ELU-SegNet-LMs-CRFs) is the winner showing the best performance on any measures and provinces. Also, an improvement in the satellite images is higher than that in the aerial images. More details on each proposed strategy will be discussed.

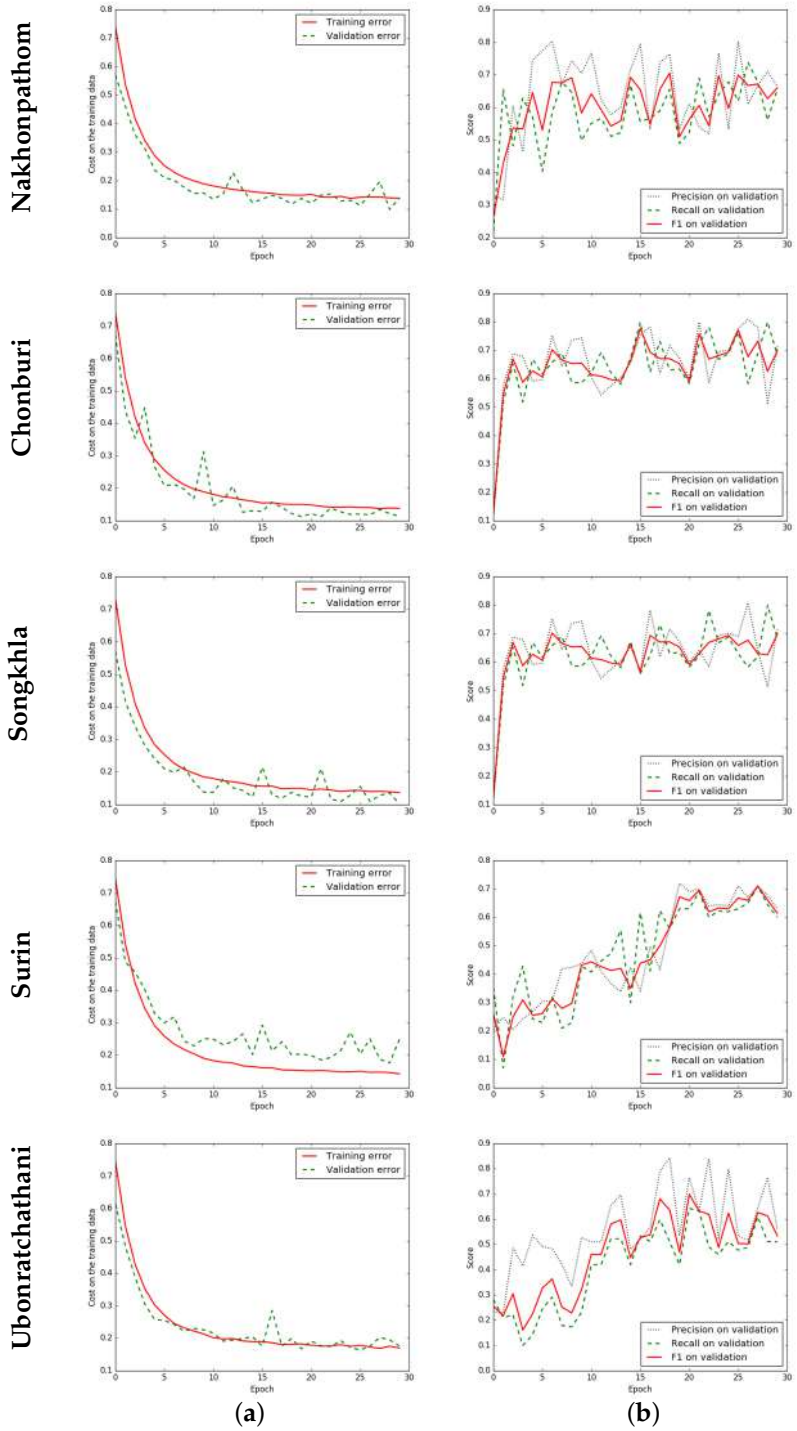



**Figure 8.** Iteration plot on THEOS satellite data sets of the proposed technique, ELU-SegNet-LMs-CRFs. $x$ refers to epochs and $y$ refers to different measures. Each row refers to different data set (province). (**a**) Plot of model loss (cross entropy) on training and validation data sets; and (**b**) Performance plot on the validation data set.

**Table 4.** *F*1 on the testing data of the Thailand Earth Observation System (THEOS) satellite data sets between baseline (SegNet) and three variations of our proposed techniques; columns refer to five different provinces (data sets).

| | Model | Nakhon. | Chonburi | Songkhla | Surin | Ubon. | Avg. |
|---|---|---|---|---|---|---|---|
| **Baseline** | SegNet | 0.422 | 0.572 | 0.424 | 0.501 | 0.406 | 0.465 |
| **Proposed Method** | **ELU**-SegNet | 0.463 | 0.690 | 0.497 | 0.591 | 0.534 | 0.555 |
| | ELU-SegNet-**LMs** | 0.488 | 0.732 | 0.526 | 0.625 | 0.562 | 0.587 |
| | ELU-SegNet-LMs-**CRFs** | **0.550** | **0.775** | **0.607** | **0.707** | **0.608** | **0.649** |

**Table 5.** *precision* on the testing data of THEOS satellite data sets between baseline (SegNet) and three variations of our proposed techniques; columns refer to five different provinces (data sets).

| | Model | Nakhon. | Chonburi | Songkhla | Surin | Ubon. | Avg. |
|---|---|---|---|---|---|---|---|
| **Baseline** | SegNet | 0.435 | 0.668 | 0.456 | 0.598 | 0.601 | 0.552 |
| **Proposed Method** | **ELU**-SegNet | 0.410 | 0.702 | 0.478 | **0.840** | 0.852 | 0.656 |
| | ELU-SegNet-**LMs** | 0.494 | 0.852 | 0.557 | 0.770 | 0.867 | 0.708 |
| | ELU-SegNet-LMs-**CRFs** | **0.535** | **0.909** | **0.650** | 0.786 | **0.871** | **0.751** |

**Table 6.** *recall* on the testing data of THEOS satellite data sets between baseline (SegNet) and three variations of our proposed techniques; columns refer to five different provinces (data sets).

| | Model | Nakhon. | Chonburi | Songkhla | Surin | Ubon. | Avg. |
|---|---|---|---|---|---|---|---|
| **Baseline** | SegNet | 0.410 | 0.499 | 0.395 | 0.431 | 0.306 | 0.408 |
| **Proposed Method** | **ELU**-SegNet | 0.532 | **0.678** | 0.517 | 0.456 | 0.389 | 0.515 |
| | ELU-SegNet-**LMs** | 0.483 | 0.642 | 0.498 | 0.526 | 0.416 | 0.513 |
| | ELU-SegNet-LMs-**CRFs** | **0.566** | **0.676** | **0.570** | **0.643** | **0.467** | **0.584** |

#### 5.2.1. Results of Enhanced SegNet (ELU-SegNet)

The ELU activation function can increase the performance of the network. In terms of *F*1, Table 4 shows that ELU-SegNet outperforms the traditional network (SegNet) for all provinces. It performs better than SegNet by 9.08% on average for all provinces, where Ubonratchathani and Chonburi show the highest *F*1-improvement, at over 10%. For *precision* and *recall*, Tables 5 and 6 illustrate that almost all data sets can be improved employing the ELU function with improvements of 10.48% and 10.68% on average for all provinces, respectively, .

#### 5.2.2. Results of Enhanced SegNet with Landscape Metrics (ELU-SegNet-LMs)

The LMs filtering strategy aims to remove all inaccurately extracted roads (false positives: FP) resulting in higher *precision* and *F*1, but this might imply a slight loss in *recall*. Comparing to the previous generation (ELU-SegNet), there are improvements by LMs on average for all provinces of 5.2% and 3.2% in terms of *precision* (Table 5) and *F*1 (Table 4), respectively, with a slight loss of −0.22% in terms of *recall* (Table 6). Compared to the baseline, LMs outperforms SegNet on all performance measures.

#### 5.2.3. Results of All Modules (ELU-SegNet-LMs-CRFs)

To further improve the performance, CRFs is integrated into the network from the previous section. This is considered to use all proposed modules: ELU, LMs, and CRFs. From Tables 4–6, the results show that ELU-SegNet-LMs-CRFs is the winner compared the previous generations and baseline (SegNet) on any of the measures (*precision*, *recall*, and *F*1). As of *F*1 average of all provinces, it outperforms ELU-SegNet-LMs, ELU-SegNet, and SegNet by 6.28%, 9.44% and 18.44%, respectively.

Figures 9–13 show sample results from the proposed method on five provinces. The results of the last column look closest to the ground truth in the second column.

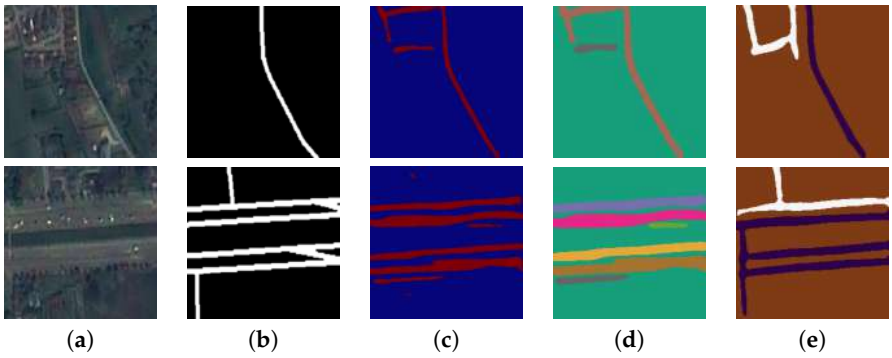



**Figure 9.** Two sample input and output THEOS satellite images on the Nakhonpathom data set, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) Output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

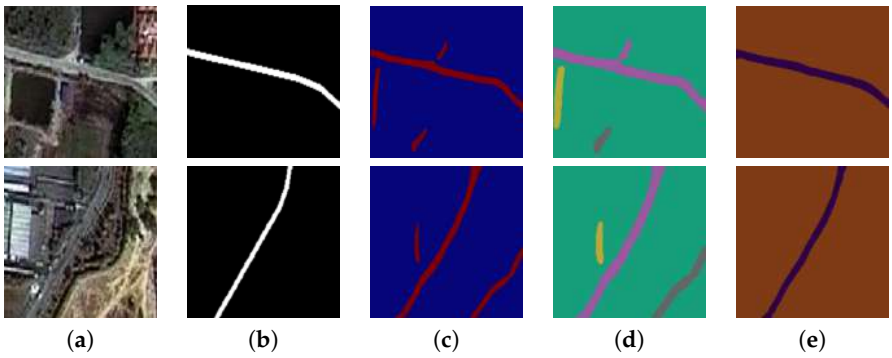



**Figure 10.** Two sample input and output THEOS satellite images on the Chonburi data set, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) Output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

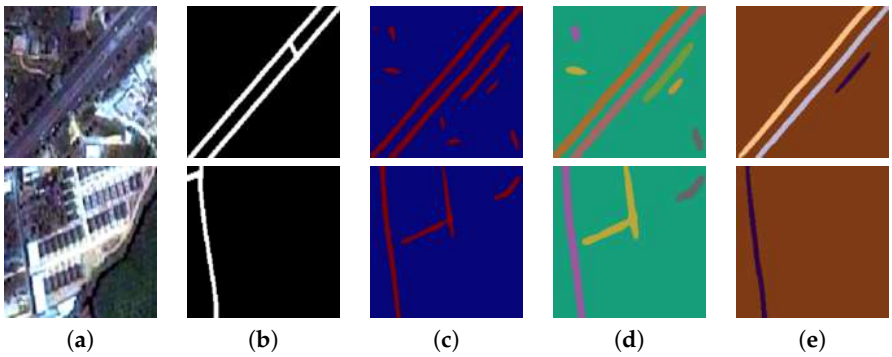



**Figure 11.** Two sample input and output THEOS satellite images on the Songkhla data set, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) Output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

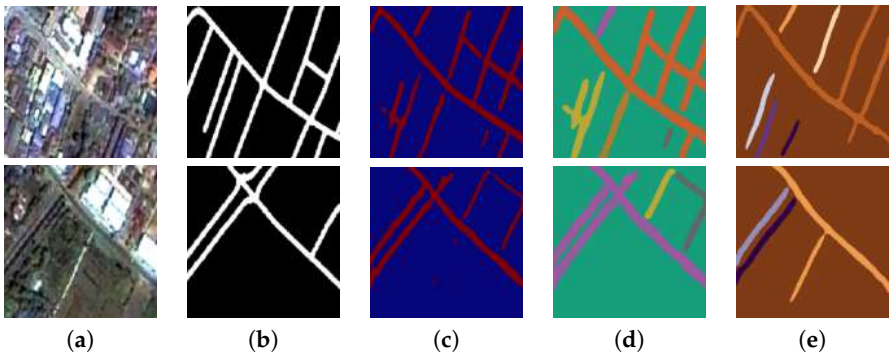



**Figure 12.** Two sample input and output THEOS satellite images on the Surin data set, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

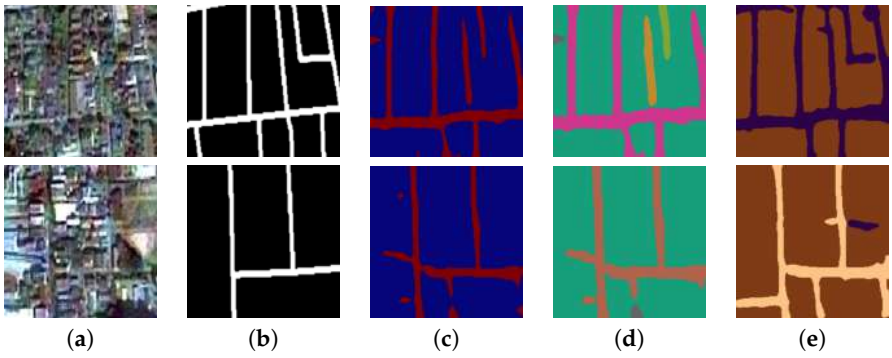



**Figure 13.** Two sample input and output THEOS satellite images on Ubonratchathani data set, where rows refer different images. (**a**) Original input image; (**b**) Target road map (ground truth); (**c**) Output of ELU-SegNet; (**d**) Output of ELU-SegNet-LMs; and (**e**) Output of ELU-SegNet-LMs-CRFs.

### *5.3. Discussions*

In terms of accuracy ($F1$-measure), the results have shown that our proposed framework with all strategies (ELU-SegNet-LMs-CRFs) outperforms the state-of-the-art algorithm, SegNet. On the aerial imagery, our $F1$ (0.876) is greater than SegNet's $F1$ (0.768) by 10.8%. On the satellite imagery, our $F1$ (0.6494) is greater than SegNet's $F1$ (0.465) by 18.44% on average for all five provinces. In terms of the computational cost, our framework requires slightly additional training time compared to the baseline approach, SegNet, by about 6.25% (2–3 h). In our experiment, SegNet's training procedure took approximately 48 h per data set, and finished after 200 epochs with 864 s per epoch. Our framework is built on top of SegNet. There is no additional time required by changing an activation function from ReLU to ELU. The LMs and CRF processes took around 1–2 h and 1 h, consecutively, so there are approximately 2–3 additional hours required on top of SegNet (48 h).

Although our work does not solely rely on the color feature like previous attempts in road extraction, it is recommended for application to high- and very-high resolution remotely-sensed images. It is difficult to identify roads from low- and medium-resolution images, even by humans.

## 6. Conclusions and Future Work

In this study, we present a novel deep learning network framework to extract road objects from both aerial and satellite images. The network is based on the deep convolutional encoder–decoder network (DCED), called "SegNet". To improve the network's precision, we incorporate the recent activation function, called the exponential linear unit (ELU), into our proposed method. The method is also further improved to detect more road patterns by utilizing landscape

metrics and conditional random fields. Excessive detected roads are then eliminated by applying landscape metrics thresholding. Finally, we extend the SegNet network to ELU-SegNet-LMs-CRFs. The experiments were conducted on a Massachusetts road data set as well as THEOS (Thailand) road data sets, and compared to the existing techniques. The results show that our proposed (ELU-SegNet-LMs-CRFs) outperforms the original method on both aerial and satellite imagery for $F1$ as well as for all other baselines.

In future work, more choices of image segmentation, optimization techniques and/or other activation functions will be investigated and compared to obtain the best DCED-based framework for semantic road segmentation.

**Acknowledgments:** We greatly acknowledge Geo-informatics and Space Technology Development Agency (GISTDA), Thailand, for providing satellite imagery used in this study. T. Panboonyuen thanks the scholarship from Chulalongkorn University to commemorate the 72nd Anniversary of H.M. King Bhumibala Aduladeja.

**Author Contributions:** The experiment design was carried out by all of the authors. Teerapong Panboonyuen and Peerapon Vateekul performed the experiments and results analysis. Kulsawasd Jitkajornwanich, Siam Lawawirojwong and Panu Srestasathiern supervised research and reviewed results. The article was co-written by the five authors. All authors read and approved the submitted manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CCL | connected component labeling |
| CNN | convolutional neural network |
| CRFs | conditional random fields |
| DCED | deep convolutional encoder-decoder |
| DCNN | deep convolutional neural network |
| DL | deep learning |
| ELU | exponential linear unit |
| FCIS | fully convolutional instance-aware semantic segmentation |
| FCN | fully convolutional network |
| FN | false negative |
| FP | false positive |
| GISTDA | geo-informatics and apace technology development agency |
| HR | high resolution |
| LMs | landscape metrics |
| PASCAL VOC | pascal visual object classes |
| R-CNN | region-based convolutional neural network |
| ReLU | rectified linear unit |
| RGB | red-green-blue |
| SGD | stochastic gradient descent |
| TN | true negative |
| TP | true positive |
| VGG | visual geometry group |
| VHR | very-high resolution |
| VOC | visual object classes |

## References

1. Poullis, C. Tensor-Cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images. *ISPRS J. Photogramm. Remote Sens.* **2014**, *95*, 93–108.
2. Muruganandham, S. Semantic Segmentation of Satellite Images using Deep Learning. Master Thesis, Lulea University of Technolog, Lulea, Sweden, 2016.
3. Saito, S.; Yamashita, T.; Aoki, Y. Multiple object extraction from aerial imagery with convolutional neural networks. *Electron. Imaging* **2016**, *2016*, 1–9.

4. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.
5. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1520–1528.
6. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
7. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2013.
8. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv* **2015**, arXiv:1511.00561.
9. Volpi, M.; Ferrari, V. Semantic segmentation of urban scenes by learning local class interactions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.
10. Liu, J.; Liu, B.; Lu, H. Detection guided deconvolutional network for hierarchical feature learning. *Pattern Recognit.* **2015**, *48*, 2645–2655.
11. Hong, S.; Noh, H.; Han, B. Decoupled deep neural network for semi-supervised semantic segmentation. *Adv. Neural Inf. Processing Syst.* **2015**, 1495–1503, arXiv:1506.04924.
12. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: New York, NY, USA, 2015; pp. 234–241.
13. Andrearczyk, V.; Whelan, P.F. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognit. Lett.* **2016**, *84*, 63–69.
14. Wang, J.; Song, J.; Chen, M.; Yang, Z. Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine. *Int. J. Remote Sens.* **2015**, *36*, 3144–3169.
15. Visin, F.; Ciccone, M.; Romero, A.; Kastner, K.; Cho, K.; Bengio, Y.; Matteucci, M.; Courville, A. Reseg: A recurrent neural network-based model for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 41–48.
16. Liu, Z.; Li, X.; Luo, P.; Loy, C.C.; Tang, X. Deep Learning Markov Random Field for Semantic Segmentation. *arXiv* **2016**, arXiv:1606.07230.
17. Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst.* **2011**, *2*, 4.
18. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
19. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv* **2016**, arXiv:1606.00915.
20. Audebert, N.; Saux, B.L.; Lefèvre, S. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sens.* **2017**, *9*, 368.
21. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
22. Wang, Q.; Fang, J.; Yuan, Y. Adaptive road detection via context-aware label transfer. *Neurocomputing* **2015**, *158*, 174–183.
23. Yuan, Y.; Jiang, Z.; Wang, Q. Video-based road detection via online structural learning. *Neurocomputing* **2015**, *168*, 336–347.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* **2014**, arXiv:1409.1556.
25. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully Convolutional Instance-aware Semantic Segmentation. *arXiv* **2016**, arXiv:1611.07709.
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. *arXiv* **2017**, arXiv:1703.06870.
27. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.

28. Panboonyuen, T.; Vateekul, P.; Jitkajornwanich, K.; Lawawirojwong, S. An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery. In *Recent Advances in Information and Communication Technology Series*, Proceedings of International Conference on Computing and Information Technology, Tunis, Tunisia, 27–28 April 2017; Volume 566.
29. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
30. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Gonzalez, R.; Woods, R. *Digital Image Processing*; Prentice Hall: Upper Saddle River, NJ, USA, 2008.
32. McGarigal, K. *Landscape Metrics for Categorical Map Patterns*. Available online: http://studylib.net/doc/7944344/landscape-metrics-for-categorical-map-patterns (accessed on 1 December 2008).
33. Huang, X.; Zhang, L. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *Int. J. Remote Sens.* **2009**, *30*, 1977–1987.