Undergraduate Project Report
2018/19

The Research of Semantic Segmentation with Light Neural Networks

Name:	Fuxiao Liu
School:	International School
Class:	2015215106
QM Student No.:	151010084
BUPT Student No.:	2015213011

Date: 24-4-2019

Table of Contents

Abstract

The new trend in semantic segmentation of computer vision is to utilize them in embedded systems which have limited computational ability and storage space. To achieve that, the model must be light enough. This project designs a lightweight network model to finish the semantic segmentation task in remote sensing imagery with high accuracy. In the report, the project first review the state-of-the-art deep neural models of semantic segmentation and lightweight models. Subsequently, The project presents a new cost-efficient semantic segmentation framework by using pointwise group, depthwise convolutions and channel shuffle with shortcut blocks to decrease memory and computation cost greatly. After that, conditional random field (CRF) method is used to improve the accuracy. Finally, the project measures the model on two Remote Sensing datasets: Vaihingen and Potsdam. Then compare the results with U-Net, MobileNet_Fully Convolutional Network(Mobile_FCN). The results indicate that the new architecture in this project achieves Fully Convolutional Network(FCN) level overall accuracy: 87.1% in Vaihingen dataset, while maintains less memory and computation budget than other lightweight_FCN based models.

Key words: semantic segmentation; remote sensing; lightweight model; U-Net; MobileNet; deep learning

摘要

计算机视觉语义分割的新趋势是在具有有限计算能力和存储空间的嵌入式系统中利用它们。要实现这一点，模型必须足够轻。该项目设计了一个轻量级的网络模型，以高精度完成遥感图像中的语义分割任务。在该报告中，该项目首先回顾了语义分割和轻量级模型的最新模型。随后，该项目提出了一种新的高效的语义分割框架，通过使用逐点卷积，深度卷积和带有快捷方式的通道混洗来大大降低内存和计算成本。之后，使用条件随机场（CRF）方法来提高准确性。最后，该项目在两个遥感数据集的模型：Vaihingen 和 Potsdam 中估计模型的效果。然后将结果与 U-Net，MobileNet_完全卷积网络（Mobile_FCN）进行比较。结果表明我们的新架构实现了完全卷积网络（FCN）级别的整体准确度：在 Vaihingen 数据集中准确率为 87.1%，比其他的语义分割模型有更少的内存和计算预算。

关键词：语义分割；遥感；轻量级模型;U-Net；MobileNet；深度学习

Chapter 1: Introduction

1.1 Project purpose

In recent years, a large number of sematic segmentation methods have been proposed to classify very-high-resolution (VHR) remotely sensed imagery in land-use/land-cover (LULC) applications (1–7). To make the whole model to be trained end-to-end, the deep fully convolutional neural networks (FCNs) based semantic segmentation framework was proposed afterward (8,9,9–12). The VHR remotely sensed images can be segmented and classified simultaneously by a fine-tuned end-to-end FCN model.

Many FCN-based frameworks achieve higher overall accuracy than traditional pixel-based image classification (PBIC) (13) and object-based image classification (OBIC, or GeoOBIA) (14–16) methods for remote sensing images semantic segmentation, and have been successfully applied to LULC (17), object detection (18), urban mapping (19), etc.

However, this kind of structure follows and failures which most DCNN have. For instance, because of the time consumed for computational cost and the size of the model, some embedded applications with limited computational ability and storage space can hardly use these models. Especially for Satellite systems to do semantic segmentation.

In most situation, pictures taken from satellites will be transferred to ground control systems. However, satellite systems are not able to distinguish between useful and useless(have no object) pictures. So all pictures will be transferred to ground. It will take people a lot of money during the transmission process. Therefore, The program in this paper is to deal with these problems by designing a lightweight models which can be utilized in semantic segmentation. As a result, satellite systems can make some prejudgment, then just transfer the useful pictures.

1.2 Project Description

1.2.1 The main work

The program develop a new cost-efficient architecture for remote sensing imagery, which is a variant of FCN, to address the expensive cost in remote sensing imagery. Moreover, this project designed a series of comparative experiments with U-Net (20), Mobile_FCN and trained them on ISPRS semantic segmentation dataset without using Digital Surface Model(DSM).

1.2.2 The major innovation

- The project introduces new lightweight semantic segmentation framework which has better performance than other lightweight models.
- The project compares many lightweight methods and analysis the pros and cons of them.

Chapter 2: Background

2.1 semantic segmentation

2.1.1 The development of CNN

Convolutional Neural Network (CNN) is a common deep learning architecture inspired by biological natural visual cognitive mechanisms. In the 1990s, LeCun et al. and others published papers that established the modern structure of CNN and later refined it. They designed a multi-layered artificial neural network called LeNet-5 to classify handwritten numbers (21). The structure of LeNet-5 is shown in the figure 1. CNN can derive the effective representation of the original image, which enables CNN to recognize the laws above the vision directly from the original pixels with very little pre-processing. However, due to the lack of large-scale training data at the time, the computing power of the computer could not keep up. LeNet-5's processing of complex problems was not satisfactory.

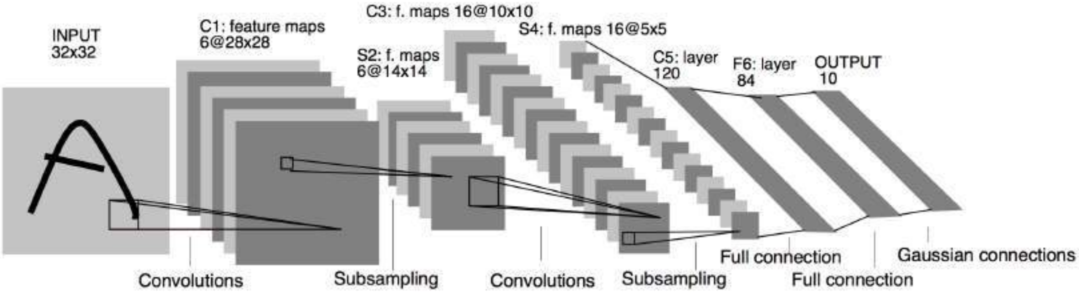


Fig. 1. Architecture of LeNet-5(21).

Since 2006, many methods have been designed to overcome the difficulty of training deep CNN. Among them, the most famous is Krizhevsky et al. proposed a classic CNN structure and made a major breakthrough in image recognition tasks. The overall framework for its approach is called AlexNet(22). Its structure is similar to that of LeNet-5, but it is deeper.

After AlexNet's success, the researchers proposed other improvements, the most famous of which are ZFNet (23), VGGNet (24), GoogleNet (25) and ResNet (26). One direction of CNN development is that the number of layers has become more. For example, ILSVRC 2015 champion ResNet (26) is more than 20 times that of AlexNet (22) and more than 8 times that of VGGNet (24). By increasing the depth, the network can more closely represent the deep information with the added nonlinear structure, which can better characterize the image. However, these methods also increase the overall complexity of the network, making the network difficult to optimize and easy to overfit.

In recent years, the CNN model has been used as the basis for various image tasks, such as image classification, target detection and semantic segmentation.

2.1.2 FCN

Before deep learning was applied to the field of computer vision, researchers generally used texture primitive forest (Texton Forest (27)) or random forest (Random Forest (28)) methods to construct classifiers for semantic segmentation.

In 2014, Fully Convolutional Networks (FCN), proposed by Long et al (29). at the University of California at Berkeley, extended the original CNN structure to enable intensive prediction without a fully connected layer. The proposed structure allows the segmentation map to generate images of any size, and also improves the processing speed compared to the image block classification method. Later, almost all recent research on semantic segmentation adopted this structure. Figure 1 shows the basic structure of the FCN.

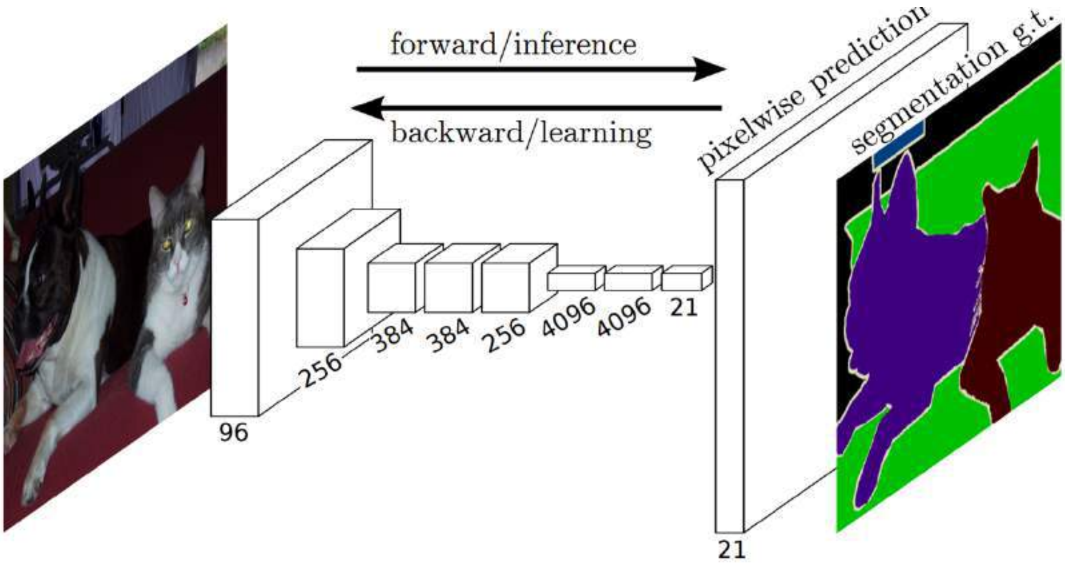


Fig. 2. Architecture of FCN (29).

However, there is a problem with this fully connected layer structure: the existence of a pooled layer. The pooling layer can increase the receptive field of the upper convolution kernel, however, it discards part of the location information while aggregating the background. The semantic segmentation method requires precise adjustment of the category map, so the position information discarded in the pooling layer needs to be retained. In this regard, the researchers proposed two different forms of structure to solve this problem.

2.1.3 U-Net

The first method is an encoder-decoder structure. Among them, the encoder uses the pooling layer to gradually reduce the spatial dimension of the input data, and the decoder gradually recovers the details of the target and the corresponding spatial dimension through a network layer such as a deconvolution layer. It is worth noting that there is usually a direct information connection between the encoder and the decoder to help the decoder better recover the target details. In this method, a typical structure is a U-Net (20) which is showed as figure 3.

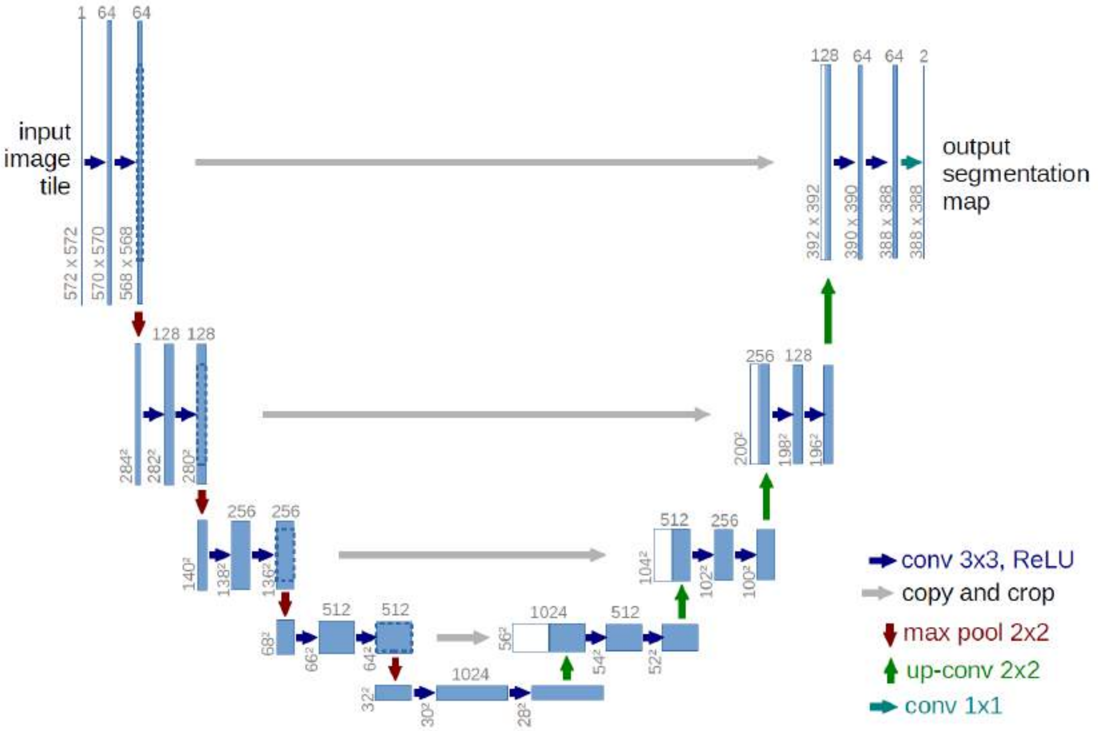


Fig. 3. Architecture of U-Net (20).

2.1.4 Dilated Convolutions

The second method uses a structure called hole convolution. This structure was first proposed by Yu, Fisher et al (30), which replaced the pooled layer structure in the middle of the FCN with the structure shown in Fig. 4, and removed the pooled layer structure. The advantage of dilated is that without the pooling operation, the receptive field is increased, and each convolution output contains a large range of information.

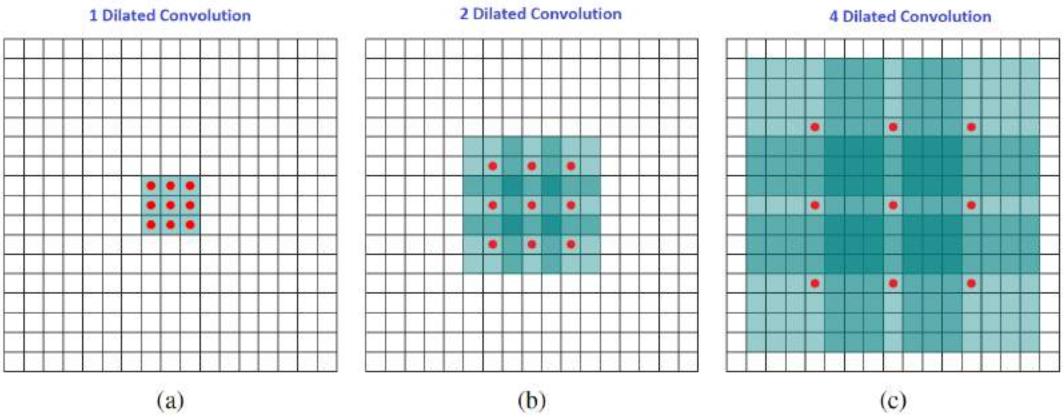


Figure 4: The detail of Dilated Convolutions: systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage, (a) F1 is produced from F0 by a 1-dilated convolution; each element in F1 has a receptive field of 3×3, (b) F2 is produced from F1 by a 2-dilated convolution; each element in F2 has a receptive field of 7×7, (c) F3 is produced from F2 by a 4-dilated convolution; each element in F3 has a receptive field of 15×15 (30).

2.2 Lightweight Model

The deep neural network model is widely used in machine vision tasks such as image classification and object detection, and has achieved great success. However, due to storage space and power consumption limitations, the storage and computation of neural network models on embedded devices remains a huge challenge.

In order to solve this problem, artificially designing a lightweight neural network model is a common method. Currently, there are four types of state-of-the-art lightweight neural network models: MobileNet (31), MobileNetV2 (32), ShuffleNet(33) and ShuffleNetV2(34). They are lightweighted by special convolution methods and structures.

2.2.1 MobileNet and MobileNet V2

MobileNetV1 is Google's first convolutional neural network for small, computationally intensive, mobile devices. The reason why MobileNetV1 is so lightweight is that it replaces the standard convolution with Depthwise Separable Convolution (31) and uses the width multiply (31) to reduce the amount of parameters.

MobileNetV1 is designed with reference to the traditional VGG Net chain architecture to increase the network depth by stacking convolution layers to improve recognition accuracy. But there is a problem when stacking too many convolutions, which is the gradient of Vanishing. However, the residual network makes it easier for information to flow between layers, which provides feature reuse in forward propagation while mitigating the disappearance of gradient signals during backpropagation. So the improved version of MobileNet V2 adds skip connection structure. The basic blocks of ResNet and Mobilenet V1 are improved as follows:

- Continue to use the deep separable convolution of Mobilenet V1 to reduce the amount of convolution calculations.
- Increase the skip connection to provide feature reuse for forward propagation.
- Inverted residual block structure is adopted. The structure uses Point wise
- The feature map is upgraded by convolution, and then the ReLU is connected to the feature map after the dimension is upgraded to reduce the damage of the feature by ReLU.

Figure 5,6 shows the different between MobileNet and MobileNet V2

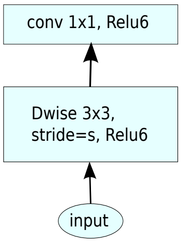


Fig. 5. Architecture of MobileNet (31).

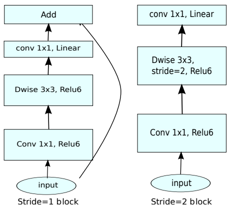


Fig. 6. Architecture of MobileNetV2(34).

2.2.2 ShuffleNet and ShuffleNet V2

ShuffleNet is a lightweight network structure proposed by Face++. The main idea is to improve ResNet with Group convolution(33) and Channel shuffle(33), which can be regarded as a compressed version of ResNet.

The common feature of Mobile V1&V2, shuffleNet V1 is that FLOPS is used as the evaluation standard of the model, but each condition needs to be met in the mobile terminal device: fewer parameters, faster speed and higher precision. Therefore, less parameters do not necessarily represent The model is fast and accurate. In this regard, Face++ proposed ShuffeNet V2 replaces indirect evaluation indicators (such as FLOPS) with direct indicators (operation speed) and evaluates them in mobile terminals such as ARM. Based on reducing the amount of calculations, four principles are proposed:

- Increase the amount of calculation of convolution using different input and output channel widths;
- Reduce the group convolution increases the MAC;
- Use multi-branch to reduce computational efficiency;
- Use element level operations to increase the amount of calculation.

Figure 7 shows the differences between shuffleNet and shuffleNet V2 (a) ShuffleNet base unit; (b) ShuffleNet unit for spatial downsampling (2×); (c) basic unit for ShuffleNet V2; (d) ShuffleNet V2 unit for space Sampling (2×).

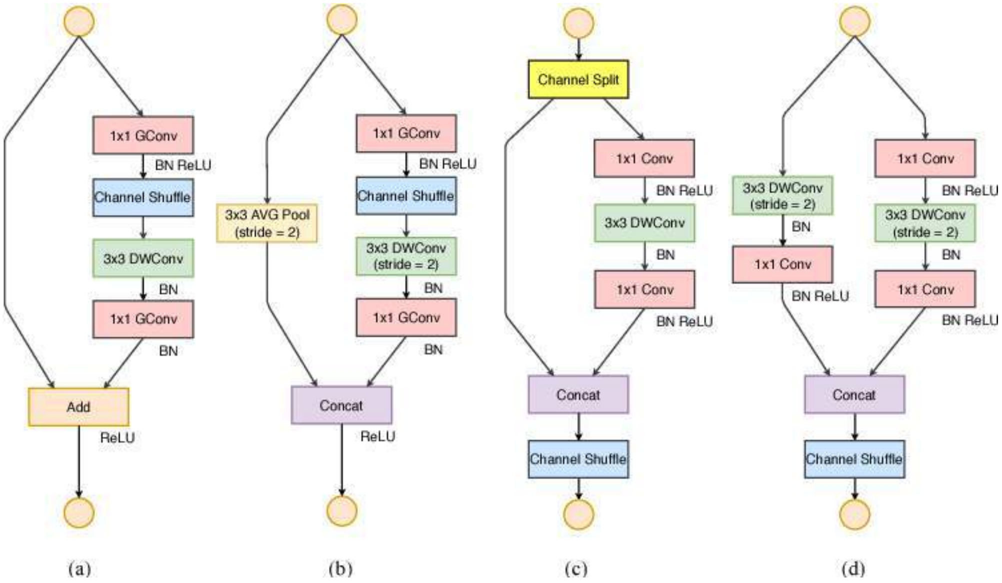


Fig. 7. The differences between shuffleNet and shuffleNet V2(34).

Chapter 3: Design and Implementation

Normally there will be a part about the design and implementation of the system, especially for an implementation project. However, every project has its unique phases so you should talk to your supervisor about it.

3.1 Lightweight Model Strategy

In most classic light weight models, they achieve our purpose by replacing the standard convolution into some special convolution methods. This program will utilize some convolution methods as part of the whole models. Then comparing them with experiments.

3.1.1 Depthwise separable convolution

MobileNet is one of the classic lightweight model and the main innovation in MobileNet is Depthwise separable convolution (31). It is a special convolutional method, it has two steps: depthwise convolution and pointwise convolution which is showed in figure 8. In the first step, deep convolution is equipped with a separate filter for each input channel. Then, the pointwise convolution combines the results of the previous step with a 1 × 1 convolution. Standard convolution can filter in one step and combine the inputs into a new set of outputs, while deepwise separable convolution divides it into two layers: one for filtering and one for combining.

This structure has the effect of significantly reducing the computation and model size. Consider a feature map with M × Win × Hin (Win and Hin are spatial width and height of the input, M is the input depth) as input and a N × Wout × Hout (Wout and Hout are spatial width and height of the output, N is the output depth) as output. The convolution kernel is of size K × K. For standard convolution, the computational cost :

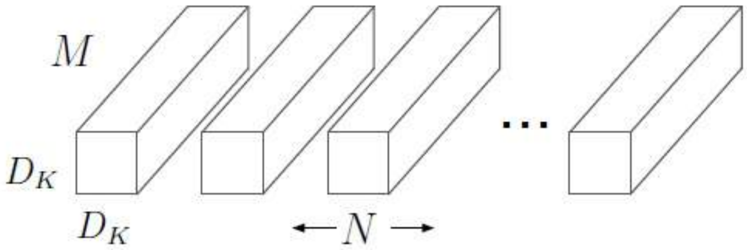
$$k^2 MNW_{in}H_{in}FLOPs \tag{1}$$

The multiplying Relationship between there parameters produces huge budgets. To address such problems, the depthwise separable convolutions which factorize the standard one into a depthwise convolution and a 1 ×1 convolution named pointwise convolution was proposed as the basic block of MobileNet. We can see it in Fig. 8(b). This technique apply a single filter for every input channel as depthwise convolution operation. In spite that this operation is extremely more efficient, it also breaks the interaction between different channels. So, the pointwise convolutions combine them to generate new 2-D features.

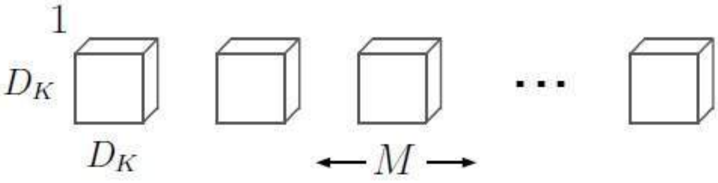
To compare, for depthwise Separable convolutions, the computational cost is:

$$(k^2MW_{in}H_{in} + MNW_{in}H_{in})FLOPs \tag{2}$$

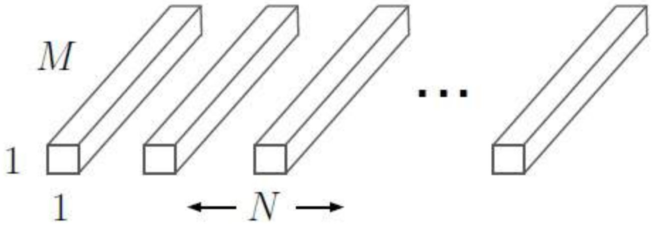
The cost and parameters of the depthwise separable convolution are 9 times less than the standard convolution If we assume K = 3 and the number of channels is 100.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c)Pointwise Convolution

Figure 8. (a) The standard convolutional filters, (b) depthwise convolution, (c) pointwise convolution(31)

3.1.2 Group Convolution

Although the above structure can effectively reduce the amount of calculation, it still takes a long time to perform pointwise convolution. To reduce this problem, the project use a group convolution similar to shuffleNet (33), which divides the different channels into different groups and then performs convolution operations in separate groups.

Consider g as the number of groups, them the computation cost will reduce g time in each bottleneck as it's showed in figure 9.

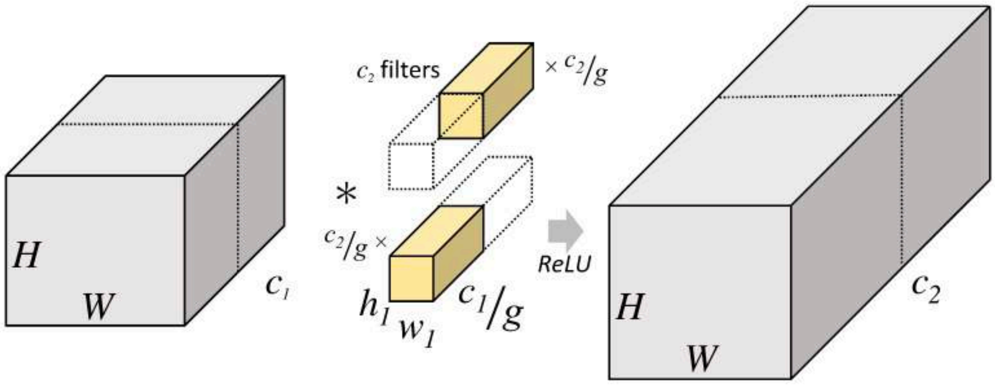


Figure 9. the process of group convolution

3.1.3 Channel shuffle

Group convolution will blocks the connection between different channels. Therefore, this project use the "channel shuffling" operation. It achieves the effect of increasing the connection between the channels by shuffling the components from different channels as it is showed in figure 10.

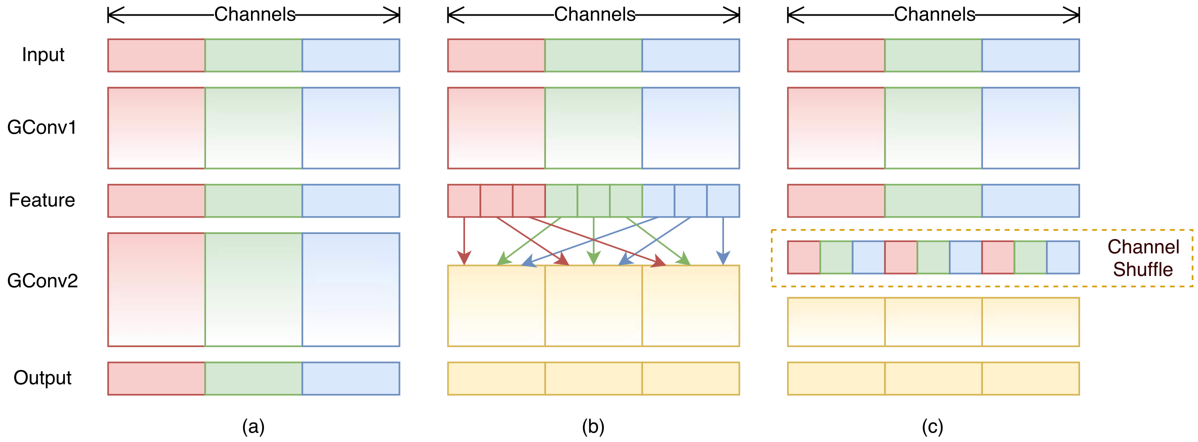


Figure 10. the process of channel shuffle(33)

3.1.4 Channel Split and Concatenation

Most models will utilize shortcut block to increase connection between two convolution process. However, most of them use "add" method to connect two shortcut part. This project replace "add" method to "concat" method. At the begin of shortcut block, input channel will be split into two part and then do convolution process separately. After that, two part will be concatenated together as it is showed in figure 11. Compared to "add" method, "concat" method will reduce two time computation cost.

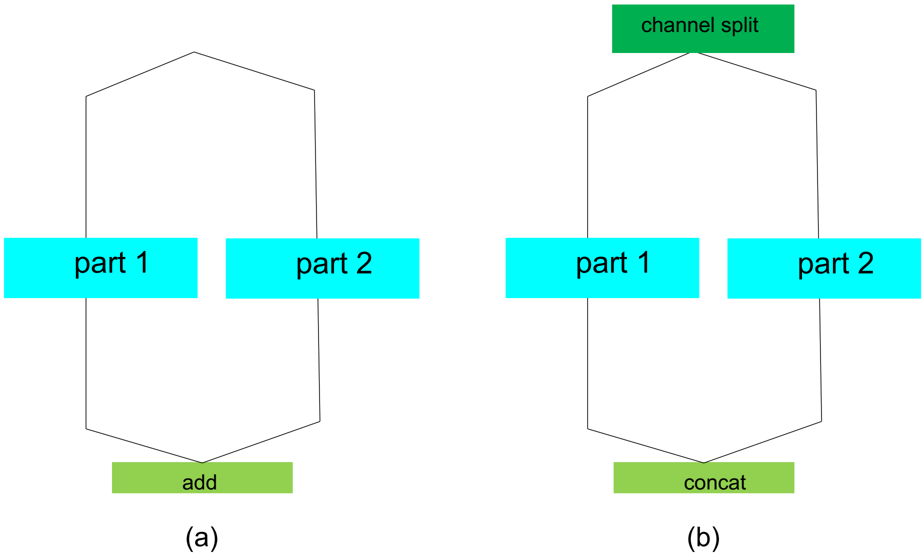


Figure 11. (a) traditional shortcut method. (b) shortcut method in this project.

3.2 Construct the Model

3.2.1 Basic Shortcut Convolutional Block

Figure 12 shows the basic block in this project, which is modified from ShuffleNet unit. This block has two branches in figure. 5(a). Firstly, it started with a channel split to split channels from the former process into two parts which have equal channel number. After that the project utilizes a 1 × 1 pointwise group convolution and 3 x 3 depthwise convolution in one part. Between them, channel shuffle is used to increase the information exchange between different channel. Then, to make the channel number equal to another part, another pointwise group convolution is used. Specially, batch normalization[youyige] is used to solve the covariate shift problem, in each convolution layer. Additionnal, ReLU (35) method is used in each convolution part.

$$BN(x) = \max\left(\frac{x - E(x)}{\sqrt{Var(x) + \alpha}} \times \beta + \gamma, 0\right) \tag{3}$$

At last, dropout(proposed by Hintion [liu 35]) method is used to deal with the overfitting problem.

$$F(x) = Dropout(ReLU(F_{main}(x) + x)) \tag{4}$$

Fig. 12(b) shows down samples process by replacing the stride number from 1 to 2. At the same time, average pooling is used in another branch.

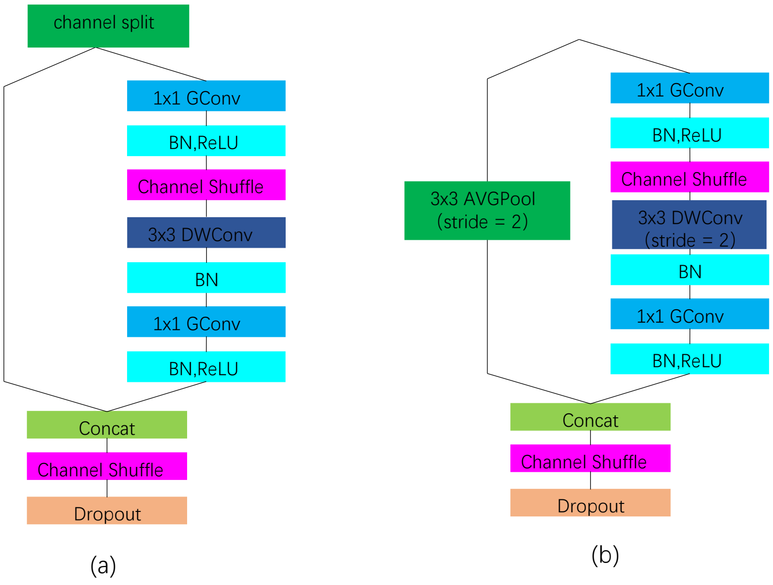


Figure 12. **(a)**block with stride =1. **(b)**block with stride =2.

3.2.2 Channel Number and Group Number

There are two parameters which will affect the computation cost and accuracy in this structure: channel number and group number in each block: the more the channel number, the high the accuracy, because it will extract more information of the picture. However, it will have more computation cost. And the situation for group number is just opposite. Because the more group number, the worse the information exchange between different channels.

The project have determined the best channel number and group number for special database.

3.3 Whole network structure

Figure 6 shows the overall symmetrical framework for the project. This structure is similar as FCN. But we apply four shortcut process (the red part in figure 13) to increase the connection of information from encoder to decoder. The green part in figure 13 shows the decoder process, which can be written as follows:

$$F_{output}(x_n, x_n') = F_{main}(x_n) + F_{fine}(x_n) + F'(x_n') \tag{5}$$

$$F'(x_n') = Dropout(BN(ReLU(Conv(x_n', W)), \beta = 0.5) \tag{6}$$

As a result , the softmax layer is used to transfer the last layer into classification maps.

All sematic segmentation models in this project follows this structure. The main difference among them are the blocks(different models contain different contents). Table 1 shows the detail in every part. Each parameter is determined by experiment.

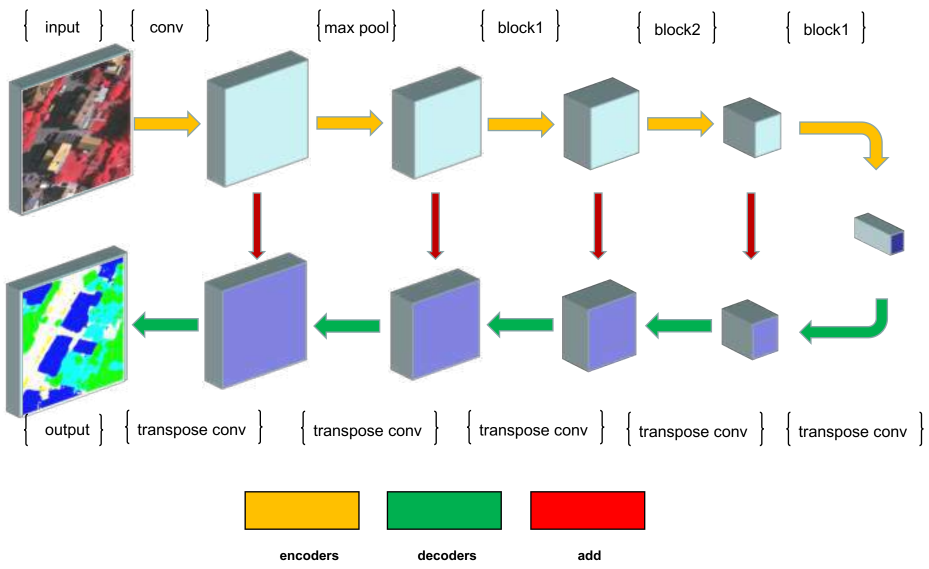


Figure 13. whole structure for lightweight semantic segmentation

Table 1. Detail in this project

Layer name	Output Size	KSize	Stride	Repeat
Original Image	128x128x3			
Conv1	64x64x24	3x3	2	1
MaxPool	32x32x24	3x3	2	
Stage2	16x16x100		2	1
	16x16x100		1	3
Stage3	8x8x200		2	1
	8x8x200		1	7
Stage4	4x4x400		2	1

	4x4x400		1	3
Transpose Conv1	8x8x200	3x3	2	1
Transpose Conv2	16x16x100	3x3	2	1
Transpose Conv3	32x32x24	3x3	2	1
Transpose Conv4	64x64x24	3x3	2	1
Output Image	128x128x6	3x3	2	1

3.4 Post Processing --- CRF

Convolutional neural networks have achieved good results in solving semantic segmentation tasks. Although deep neural networks are effective in extracting local features and using small receptive fields for good prediction, they lack the ability to utilize global context information and cannot directly model the interaction between predictions. Therefore, this project uses DenseCRF (36) to establish a global information connection.

There is a category label x_i for each pixel i. and a corresponding observation y_i. Thus each pixel point acts as a node, and the relationship between the pixel and the pixel acts as an edge, which constitutes a conditional random field. The category label xi corresponding to the pixel i.can be inferred by observing the variable y_i. The conditional random field results are shown in the figure:

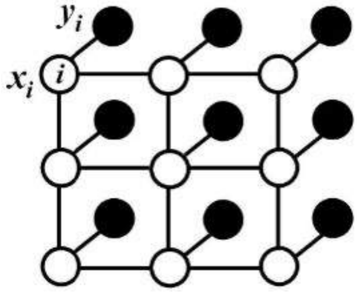


Figure 14. the structure for DenseCRF

The conditional random field conforms to the Gibbs distribution: (where x is the observation value)

$$P(X = x|I) = \frac{1}{Z(I)} e^{-E(x|I)} \tag{7}$$

Where $E(x|I)$ is the energy function：

$$E(x|I) = \sum_{i} \Psi_u(x_i) + \sum_{i<j} \Psi_p(x_i, x_j) \tag{8}$$

The one-dimensional potential function $\sum_i \Psi_u(x_i)$ is the output from the front-end FCN. The binary potential function is as follows:

$$\Psi_p(x_i, x_j) = u(x_i, x_j) \sum_{m=1}^{M} \omega^m k_G^m(f_i, f_j) \tag{9}$$

The binary potential function is to describe the relationship between pixel points and pixel points, and encourage similar pixels to assign the same label, while pixels with larger differences assign different labels. The definition of this "distance" is related to the color value and the actual relative distance, so CRF can make the image split as much as possible at the boundary.

Chapter 4: Results and Discussion

4.1 Overview

The next past will show several experiments in two special database. And discuss them based on the result.

The project is conducted on the TensorFlow deep learning framework. It took 4 hours on the two ISPRS VHR(Very High Resolution) image datasets on GPU GeForce GTX 1080. The aim of the experiments is to classify all the pixels in the digital orthophoto maps into six categories.

The whole of our experiments are divided into two parts. In the first part, we describe our processes to find the best mini batch size, group numbers and stage channel numbers of the model in this project on Vaihingen dataset. Then contrast the model with some other models.

The second part demonstrates the state-of-art performance of our network on Potsdam dataset.

In the implementation, each model will keep training until no better result appear within 20 epochs. To address the vanishing gradient problem in training process[Fig.15], we use Adadelta method(37) [38] because of its unique updating quality(decrease from 0.5 to 0). In addition, we set weight decay to 4e-5. Other hyper-parameters follows(38) [39]. The assessment metrics utilized in our experiments are overall accuracy(OA), kappa, model size, computation complexity(MAdds).

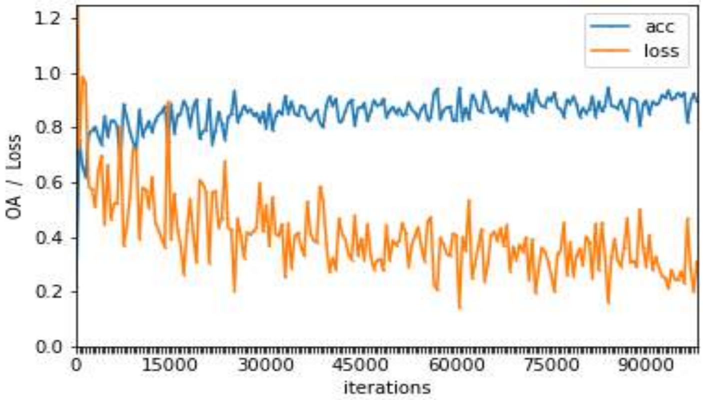


Figure 15. Monitoring training of DSFCN.

4.2 Experiments on Vaihingen dataset

4.2.1 Data details

The Vaihingen dataset[Fig. 16(a)-(c)] contains three spectral bands: red (R), green (G), and near infrared (IR). In our training process, we only utilize digital orthophoto maps(DOMs) without digital surface models(DSMs). Among the DOMs, we use ID 1, 5, 7, 13, 15, 17, 21, 23, 28, 32,34, 37 as training data and ID 3, 11, 26, 30 as validation. Then we sliced these data into the images with a shape of 128 × 128 × 3 and each two adjacent small training images have 64% overlap. In addition, we also flip our original data horizontally and vertically to increase the training data. As a result, 23474 patches exist as training dataset and 1023 patches are used as validation dataset.

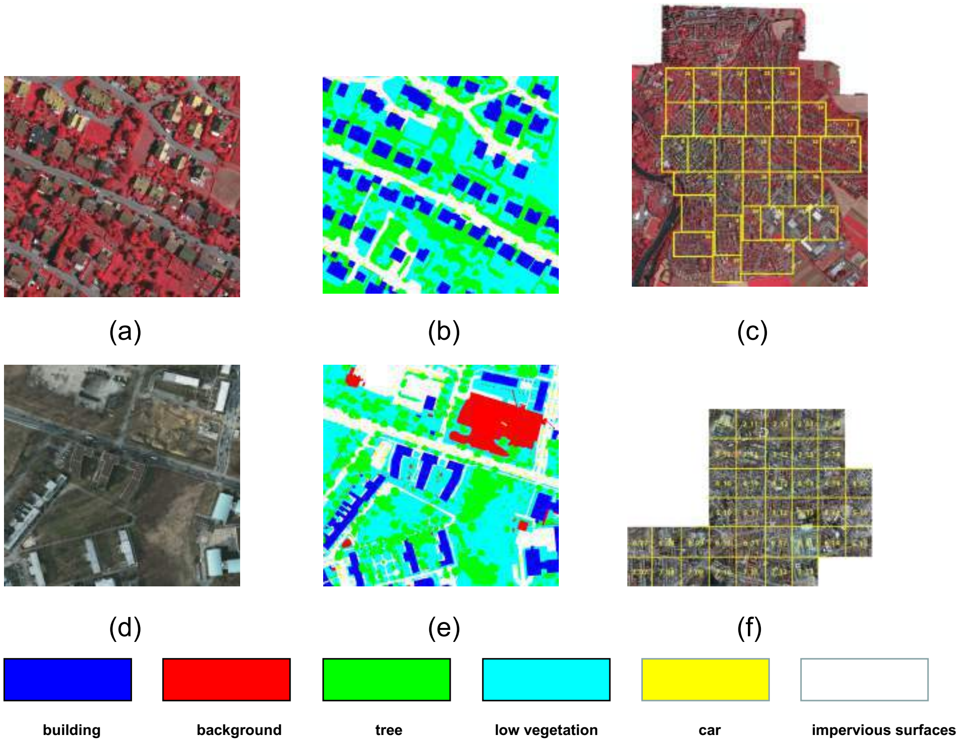


Figure 16. (a) Tile 33 in Vaihingen dataset. (b)Ground truth. (c) Vaihingen dataset. (d)Tile 3-12 in Potsdam. (f)Ground truth. (e)Potsdam dataset.

4.2.2 Minibatch size

In this part, to find the relationship between batch size and performance, our model is evaluated with different batch size while the group number is 1 and stage channels are 100, 200, 400. The Validation OA, Validation Kappa and Training OA are listed in Table 2. The result indicates that under the same settings, our framework with batch size 5 have a better performance on validation datasets, although it may not perform best on training datasets.

Table 2. Accuracy assessment of different batch size on Vaihingen dataset. Group number =1, Stage depth = 100,200,400. Without CRF

Minibatch Size	Validation OA	Validation Kappa	Training OA
3	0.84762	0.79321	0.88315
4	0.84631	0.79117	0.87212
5	0.85141	0.80209	0.90681
6	0.84266	0.78535	0.87978
10	0.84465	0.78908	0.89839
15	0.83845	0.78472	0.92140
20	0.84103	0.78853	0.92631
25	0.83944	0.78591	0.87832
30	0.83391	0.77865	0.88941
35	0.83830	0.78439	0.89267

4.2.3 Group number

when group number is larger, this model with channel shuffle perform better than the counterparts. In this part, to get the best result, we evaluate the model with different group numbers(1,2,4,8) on constant condition that batch size is 5 and stage channels are 100,200,400. The complete results are exhibited in Table 3. It shows that when choosing the group number as 1, the performance outperform others both in accuracy and model size. That is because models with other group numbers block the process of information and representation flowing between channel groups.

Table 3. Accuracy assessment of different group numbers on Vaihingen dataset. Batch size = 5, Stage depth = 100,200,400. Without CRF

Group Number	Validation OA	Validation Kappa	Training OA	Model Size
1	**0.85141**	**0.80209**	**0.90681**	**17.2MB**
2	0.84583	0.79474	0.87913	19.3MB
4	0.84656	0.79565	0.87608	23MB
8	0.83838	0.78457	0.87408	29.4MB

4.2.4 Stage depth

As Section 3 mentions, the stage depth plays a key role in the memory and computation complexity. In order to find the best depth which can achieve relatively small model size and maintain accurate, we adapt our experiment with three different depths while the batch size is 5 and group number is 1. The result in Table 4 indicates that as the stage depth goes deeper, its relatively size will be larger. Moreover, the Validation OA, Kappa and Training OA of Stage depth with 100, 200, 400 perform best accuracy while maintain a relatively small model size.

Table 4. Accuracy assessment of different stage depth on Vaihingen dataset. Batch size = 5, Group number = 1. Without CRF

CNN depth	Validation OA	Validation kappa	Training OA	Model Size
72_144_288	0.84545	0.79412	0.88167	**14.2MB**
100_200_400	**0.85141**	**0.80209**	**0.90681**	**17.2MB**
256_384_512	0.85131	0.80192	0.90169	27MB
384_768_1536	**0.85231**	**0.80301**	**0.90935**	130.1MB

4.2.5 CRF Effect

Table 5 shows the effect of CRF. When adding CRF as post process, the accuracy can be improved. Fig 17 shows two examples. It is obvious that CRF can make the image split as much as possible at the boundary.

Table 5. CRF effect on Vaihingen dataset. Batch size = 5, Group number = 1. Stage depth = 100,200,400

	Validation OA	Validation kappa	Training OA
With CRF	0.87145	0.82214	0.92167
Without CRF	0.85231	0.80301	0.90935

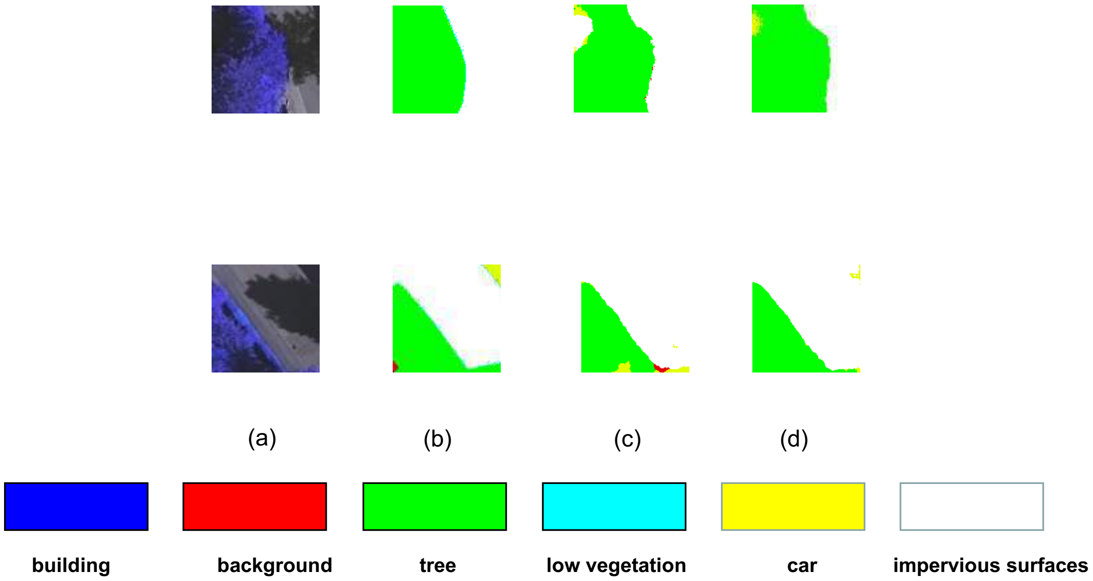


Figure 17. CRF effect example:(a) real picture. (b)Ground truth. (c) result without CRF. (d)result with CRF.

4.2.6 Comparison experiments

To emphasize our superiority, we compare our model with another lightweight net: MobileNet. It's the same to our model except that the down-sampling layers are replace. Meanwhile, we train U-Net as comparison. The results are listed in Table 6. It is important to notice that our Model with Depthwise Separable Convolution Blocks achieve slightly better accuracy than

SegNet with much less model size and MAdds, which means the pointwise group and Depthwise convolutions help to improve networks' performance. In addition, as comparison to Mobile_FCN, it shows that bottleneck in our model makes great contribution to reduce complexity and shortcut block produces better accuracy.

Table 6. Metrics of comparison experiments with different models on Vaihingen dataset.

Model	Validation OA	Validation Kappa	Training OA	Model Size
U-Net	**0.8767**	**0.8323**	**0.9290**	117.8MB
Mobile_FCN	0.8594	0.7867	0.9058	42.3MB
Our model	0.87145	0.82214	0.92167	**17.2MB**

4.2.7 Output visualization

For the purpose to access the output label more obviously and directly, we list four examples of output images in Fig. 18. In addition, our detail semantic segmentation results on Vahingen dataset are exhibited in Table 7. Among the integrated result, building, low vegetation and impervious surfaces possess better accuracy. However, we found some regions labelled with trees are misclassified to cars. Meanwhile, cars are also confused with trees.

Table 7. The result on Vaihingen dataset.

Reference/Predicted	Building	Background	Tree	Low Vegetation	Car	Surfaces
Building	**0.9227**	0.0021	0.0045	0.0206	0.0072	0.0426
Background	0.1271	**0.0012**	0.0253	0.0004	0.0321	0.8389
Tree	0.0051	0.0004	**0.8388**	0.1343	0.0051	0.0159
Low vegetation	0.0228	0.0013	0.1394	**0.7805**	0.0072	0.0486
Car	0.0255	0.0003	0.0017	0.0169	**0.7483**	0.207
Surfaces	0.0555	0.0017	0.0162	0.0516	0.0342	**0.8404**

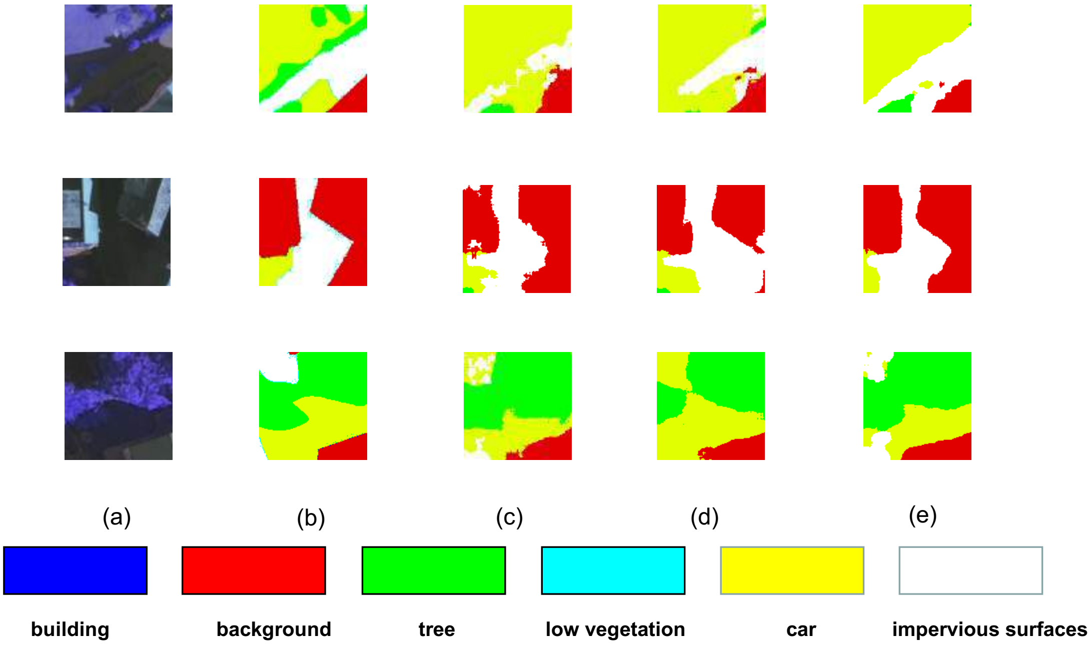


Figure 18. Example output on Vaihingen data. **(a)** Original image. **(b)** Ground Truth. **(c)** Mobile_FCN. **(d)** U-Net. **(d)** model in this project.

4.3 Experiments on Potsdam dataset

In order to demonstrate the generalization of our architecture, we compare our DSFCN with FCN-8s, SegNet and Mobile_FCN on Potsddam dataset.

4.3.1 Data details

The Potsdam dataset contains three spectral bands[Fig. 16(d)-(f)]: red (R), blue (B), green (G) and near infrared (IR). Among the DOMs, we use 18 image tiles as training data and 6 tiles as validation (the same setting as (39)). Then, we sliced these data into the images with a shape of 128 × 128 × 3. In total, 38088 patches exist as training dataset and 12696 patches are used as validation dataset.

4.3.2 Comparison experiments

The training result is listed in Table 8 and Table 9. We find that DSFCN can got better accuracy than SegNet and Mobile_FCN. Meanwhile, it maintain the least memory and computation cost.

The results again demonstrate that the model in this project with depthwise separable convolutions possesses the best performance. The result in Table 9 indicates that different from the result in Vaihingen dataset, there is no confusion between cars and trees. However, trees and vegetation are still misclassified probably.

Table 8. Metrics of comparison experiments with different models on Potsdam dataset.

Model	Validation OA	Validation Kappa	Training OA
U-Net	**0.8269**	**0.7615**	**0.8569**
Mobile_FCN	0.8200	0.7487	0.8203
My model	0.8256	0.7596	0.8559

Table 9. The result on Vaihingen dataset.

Reference/Predicted	Building	Background	Tree	Low Vegetation	Car	Surfaces
Building	**0.8834**	0.0171	0.0053	0.0094	0.0089	0.0756
Background	0.4336	**0.2002**	0.0106	0.0341	0.0424	0.2788
Tree	0.0119	0.0083	**0.6757**	0.2325	0.0046	0.0667
Low vegetation	0.0574	0.0163	0.0745	**0.7443**	0.0065	0.1007
Car	0.0102	0.0628	0.0249	0.0056	**0.81**	0.0862
Surfaces	0.0549	0.0191	0.0231	0.0241	0.0162	**0.8623**

Chapter 5: Conclusion and Further Work

In this paper, I established a lightweight semantic segmentation architecture. Our new model integrates depthwise separable convolutions and Encoder-Decoder style. According to the experiments on ISPRS Vaihingen and Potsdam datasets, I proved that my model is more efficient than the classical and U-Net. What’s more, in the comparison experiments with Mobile_FCN, the result demonstrates that our residual block makes a great contribution to improve the accuracy, reduce the memory and computation cost. In the future, we will work on to implement the following trials:

1. I will run our network in embedded device to access its performance and practicability.

References

1. McDermid GJ, Linke J, Pape AD, Laskin DN, McLane AJ, Franklin SE. Object-based approaches to change analysis and thematic map update: Challenges and limitations. Can J Remote Sens. 2008;34(5):462–6.

2. Qian Y, Zhou W, Yan J, Li W, Han L. Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. Remote Sens. 2015;7(1):153–68.

3. Wahidin N, Siregar VP, Nababan B, Jaya I, Wouthuyzen S. Object-based Image Analysis for Coral Reef Benthic Habitat Mapping with Several Classification Algorithms. Procedia Environ Sci [Internet]. 2015;24:222–7. Available from: http://dx.doi.org/10.1016/j.proenv.2015.03.029

4. Estoque RC, Murayama Y, Akiyama CM. Pixel-based and object-based classifications using high- and medium-spatial-resolution imageries in the urban and suburban landscapes. Geocarto Int. 2015;30(10):1113–29.

5. Hussain E, Shan J. Object-based urban land cover classification using rule inheritance over very high-resolution multisensor and multitemporal data. GIScience Remote Sens [Internet]. 2016;53(2):164–82. Available from: http://dx.doi.org/10.1080/15481603.2015.1122923

6. Piazza GA, Vibrans AC, Liesenberg V, Refosco JC. Object-oriented and pixel-based classification approaches to classify tropical successional stages using airborne high–spatial resolution images. GIScience Remote Sens [Internet]. 2016;53(2):206–26. Available from: https://doi.org/10.1080/15481603.2015.1130589

7. Zhang X, Chen G, Wang W, Wang Q, Dai F. Object-Based Land-Cover Supervised Classification for Very-High-Resolution UAV Images Using Stacked Denoising Autoencoders. Vol. 10, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2017. p. 3373–85.

8. Audebert N, Le Saux B, Lefèvre S. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In: Asian Conference on Computer Vision. Springer; 2016. p. 180–96.

9. Marmanis D, Schindler K, Wegner JD, Galliani S, Datcu M, Stilla U. Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS

J Photogramm Remote Sens. 2018;135:158–72.

10. Marmanis D, Wegner JD, Galliani S, Schindler K, Datcu M, Stilla U. Semantic segmentation of aerial images with an ensemble of CNNs. ISPRS Ann Photogramm Remote Sens Spat Inf Sci. 2016;3:473.

11. Zhong Z, Li J, Cui W, Jiang H. Fully convolutional networks for building and road extraction: Preliminary results. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE; 2016. p. 1591–4.

12. Maggiori E, Tarabalka Y, Charpiat G, Alliez P. Convolutional neural networks for large-scale remote-sensing image classification. IEEE Trans Geosci Remote Sens. 2017;55(2):645–57.

13. Blaschke T, Strobl J. What's wrong with pixels? Some recent developments intterfacing remote sensing and GIS. Geo-Informations-Systeme [Internet]. 2001;14(6):12–7. Available from: http://www.scopus.com/inward/record.url?eid=2-s2.0-0035382401&partnerID=tZOtx3y1

14. Hay GJ, Castilla G. Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. Object-Based Image Anal. 2008;75–89.

15. Blaschke T. Object based image analysis for remote sensing. ISPRS J Photogramm Remote Sens [Internet]. 2010;65(1):2–16. Available from: http://dx.doi.org/10.1016/j.isprsjprs.2009.06.004

16. Blaschke T, Hay GJ, Kelly M, Lang S, Hofmann P, Addink E, et al. Geographic Object-Based Image Analysis - Towards a new paradigm. ISPRS J Photogramm Remote Sens. 2014;87:180–91.

17. Zhang M, Hu X, Zhao L, Lv Y, Luo M, Pang S. Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images. Remote Sens. 2017;9(5):27–9.

18. Han X, Zhong Y, Zhang L. An efficient and robust integrated geospatial object detection framework for high spatial resolution remote sensing imagery. Remote Sens. 2017;9(7).

19. Kampffmeyer M, Jenssen R. Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban. Cvpr. 2016. p. 1–9.

20. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2015;9351:234–41.

21. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–323.

22. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25 [Internet]. Curran Associates, Inc.; 2012. p. 1097–105. Available from: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

23. Fergus MDZ and R. [Occlusion] Visualizing and Understanding Convolutional Networks. Anal Chem Res. 2018;(ICLR):818–33.

24. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014;1–14. Available from: http://arxiv.org/abs/1409.1556

25. Smoluk G. Going Deeper with Convolutions Christian. Mod Plast. 1980;57(3):62–3.

26. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2015; Available from: http://arxiv.org/abs/1512.03385

27. Shotton J, Johnson M, Cipolla R. Semantic texton forests for image categorization and segmentation. 26th IEEE Conf Comput Vis Pattern Recognition, CVPR. 2008;

28. Kontschieder P, Bulò SR, Bischof H, Pelillo M. Structured class-labels in random forests for semantic image labelling. Proceedings of the IEEE International Conference on Computer Vision. 2011. p. 2190–7.

29. Wu X. Fully convolutional networks for semantic segmentation. Comput Sci. 2015;

30. Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. 2015; Available from: http://arxiv.org/abs/1511.07122

31. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017; Available from: http://arxiv.org/abs/1704.04861

32. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proc IEEE Comput Soc Conf Comput Vis Pattern

Recognit. 2018;4510–20.

33. Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2018;6848–56.

34. Ma N, Zhang X, Zheng H-T, Sun J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. 2018;11218 LNCS:122–38. Available from: http://arxiv.org/abs/1807.11164

35. Agarap AF. Deep Learning using Rectified Linear Units (ReLU). 2018;(1):2–8. Available from: http://arxiv.org/abs/1803.08375

36. Krähenbühl P, Koltun V. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. 2012;1–9. Available from: http://arxiv.org/abs/1210.5644

37. Zeiler MD. ADADELTA: An Adaptive Learning Rate Method. 2012; Available from: http://arxiv.org/abs/1212.5701

38. Vincent P, Larocheh H, Lajoie I, Bengio Y, Manzagol P-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pascal Vincent Hugo Larochelle Yoshua Bengio Pierre-Antoine Manzagol. J Mach Learn Res [Internet]. 2010;11(Dec):3371–408. Available from: http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf

39. Chen G, Zhang X, Wang Q, Dai F, Gong Y, Zhu K. Symmetrical Dense-Shortcut Deep Fully Convolutional Networks for Semantic Segmentation of Very-High-Resolution Remote Sensing Images. IEEE J Sel Top Appl Earth Obs Remote Sens. 2018;11(5):1633–44.