

# **SEMANTIC VIDEO SEGMENTATION FROM UAV**

YIWEN WANG  
February, 2109

SUPERVISORS:  
Dr. M. Y. Yang  
Dr. S. Hosseinyalamdary





# **SEMANTIC VIDEO SEGMENTATION FROM UAV**

**YIWEN WANG**  
Enschede, The Netherlands, February, 2019

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics

SUPERVISORS:  
Dr. M. Y. Yang  
Dr. S. Hosseinyalamdary

THESIS ASSESSMENT BOARD:  
Prof. Dr. ir. M.G. Vosselman (Chair)  
Dr. M.N. Koeva; University of Twente, ITC-PGM  
etc

#### DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

## ABSTRACT

With exploding researches in deep learning, semantic video segmentation achieves remarkable advances (Jang & Kim, 2017) and has been used in a wide range of academic and real-world applications. Using video semantic segmentation technique for Unmanned Aerial Vehicle (UAV) data processing is also a popular application. The UAVs could obtain high resolution images and videos from the dangerous and inaccessible areas where the manned vehicle cannot reach with relatively low cost. It's suitable for those tasks in small or dangerous areas which require high resolution images with numerous information in details.

However, the semantic segmentation mission for UAV data also meet some special challenges caused by the characteristic of UAVs. The largest challenge is the enormous change of objects in videos. Traditional methods for video semantic segmentation for UAVs don't care about the temporal information and just extend the single image segmentation method to multiple frames. In these approaches, UAV video is viewed as a collection of consecutive frames. Each frame is a static image and is segmented individually. The segmentation result of each frame can be influenced easily by the changes of the viewpoint of the object, the changes of illumination, and deformation of the object. The same object may have different appearances in different frames and would lead to different segmentation result. Hence, the accuracy of these segmentations is relatively low. To keep the temporal consistency, the pixels of the same object in different frames should be assigned the same label.

This research proposes an FCN+Conv\_LSTM framework for semantic video segmentation. The proposed algorithm tries to combine the FCN model and the Conv\_LSTM model. In this algorithm, the FCN model serves as the frame-based segmentation method which is used to segment each frame individually. The outputs of the FCN model are sent to Conv\_LSTM model. According to the different inputs of Conv\_LSTM model, this framework is divided into two methods, one uses the segmentation result of each frame another one uses the feature map of each frame. The Conv\_LSTM model serves as the post-processing method which makes use of the temporal information between consecutive frames. The inputs of this part are sequences formed by the output segmentation results or the sequences of the feature maps extracted from FCN model. Conv\_LSTM learn the temporal information of these sequences and output the final segmentation results.

The dataset used in this experiment is the UAV videos captured in Wuhan, China from June to August 2017 and in Gronau, Germany in May 2017. 27 sequences are extracted from these videos. The experimental results show the superiority of this FCN + Conv\_LSTM model especially in some classes compared to the single image segmentation model. And the feature maps are more suitable for the Conv\_LSTM model. This result shows the usefulness of temporal information in the task of semantic video segmentation.

Keywords: FCN, Conv-LSTM, semantic video segmentation, UAV

## **ACKNOWLEDGEMENTS**

First, I would like to express my sincere gratitude to my first supervisor, Dr, M. Y. Yang, for his valuable suggestions, warm encouragement, and constant guidance. He gives me much help during the whole process of my thesis. Second, I would like to say thank you to my second supervisor, Dr, S. Hosseinyalamdary. He gives great suggestions for my writing and warm encouragement every time I meet him. Also, I would like to thank all the teachers who have taught me in this university. Their enlightening teaching helps me build a solid foundation for finishing this thesis.

I am also grateful to Ye, Yaping, and Shan for their help during the thesis writing. My thanks also go to my friends Shan, Ying, and Jun, who shared with my worries and gave me encouragement.

Last but not least, I would like to thank my parents for their great confidence in me in these years and my boyfriend for his spiritual support and encouragement. Their support and motivation help me overcome the problems I meet and accomplish this thesis.

# TABLE OF CONTENTS

---

Abstract .....	i
Acknowledgements .....	ii
Table of contents .....	iii
List of figures .....	v
List of tables .....	vi
1. Introduction.....	1
1.1. Motivation.....	1
1.2. Research identification .....	3
1.2.1. Research objectives.....	3
1.2.2. Research questions.....	3
1.3. Contributions.....	3
1.4. Innovation aimed at.....	3
1.5. Outline.....	4
2. Related Work.....	5
2.1. Methods before deep learning .....	5
2.2. Deep learning methods for semantic segmentation .....	7
2.2.1. CNN.....	7
2.2.2. FCN.....	8
2.3. Methods using temporal information .....	9
2.3.1. Overview .....	9
2.3.2. Optical flow .....	9
2.3.3. RNN.....	10
2.3.4. LSTM .....	10
2.3.5. Conv_LSTM.....	11
3. Method .....	12
3.1. Overview .....	12
3.2. Proposed frameworks .....	12
3.3. Frame-based segmentation and feature extraction.....	15
3.3.1. FCN.....	15
3.3.2. Feature extraction .....	15
3.4. Sequence construction .....	16
3.5. Convolutional LSTM.....	17
3.6. Loss function .....	17
3.7. Accuracy metric.....	18
4. Experimental Results.....	19
4.1. Dataset.....	19
4.1.1. Annotation .....	19
4.2. Parameter setting.....	20
4.3. Implementation details.....	20
4.4. Comparison between FCN and method 1.....	21
4.5. Comparison between FCN and method 2.....	23
4.6. Comparison between method1 and method 2.....	25
4.7. Discussion .....	27
5. Conclusion and Future Work .....	29

---

5.1. Answers to research questions .....	29
5.2. Recommendations .....	31
List of references .....	33

## LIST OF FIGURES

---

Figure 1.1 The example of changes between the two different shapes of the road(Lyu, Vosselman, Xia, Yilmaz, & Yang, 2018).....	2
Figure 2.1 Support Vectors Network (Cortes & Vapnik, 1995) .....	5
Figure 2.2 Label propagation in video (Badrinarayanan et al., 2010).....	6
Figure 2.3 Example of the mixture of tree model with three component temporal trees(Badrinarayanan et al., 2014) .....	6
Figure 2.4 VGG-16 architecture (Noh, Hong, & Han, 2015) .....	7
Figure 2.5 The architecture of FCN (Long et al., 2015a).....	8
Figure 2.6 Architecture of FCN8s, FCN16s and FCN32s.(Long et al., 2015b).....	8
Figure 2.7 Architecture of SegNet .....	9
Figure 2.8 Example of a single RNN (Lipton, Berkowitz, & Elkan, 2015) .....	10
Figure 3.1 Framework of method 1.....	13
Figure 3.2 Framework of method 2.....	14
Figure 3.3 Formation of blocks and sequence.....	16
Figure 3.4 Conv_LSTM structure .....	17
Figure 4.1 Example of annotation .....	20
Figure 4.2 Comparison between FCN and method1.....	22
Figure 4.3 Comparison between FCN and method 2 .....	24
Figure 4.4 Comparison between method 1 and method2.....	26

## LIST OF TABLES

---

Table 3.1 Structure of FCN8s Network .....	15
Table 4.1 IoU scores for FCN8s model and method 1 .....	21
Table 4.2 IoU scores for FCN8s model and method 2 .....	23
Table 4.3 IoU scores for method1 and method 2.....	25
Table 4.4 Comparison between tree models.....	28

# 1. INTRODUCTION

## 1.1. Motivation

In recent years, with the increase of the computational ability thanks to the wide availability of graphics processing unit (GPU), computer vision has made substantial progress (Srinivas et al., 2016). As one of the key problems in both photogrammetry and computer vision areas, video segmentation has been attracting increasing amounts of attention. Video segmentation is, in general, a spatiotemporal foreground segmentation that separating foreground objects from their background. With exploding researches in deep learning, It achieves remarkable advances (Jang & Kim, 2017) and has been used in a wide range of applications, including autonomous driving, indoor navigation (Garcia-Garcia et al., 2018), surveillance (Brutzer, Hoferlin, & Heidemann, 2011), action recognition, video retrieval (Zhang et al., 2015) and many other academic and real-world applications. This research would focus on the semantic video segmentation. The aim of the semantic video segmentation is not only separating the foreground objects from their background, but also assign a correct class label to every pixel in a video (Mahasseni, Todorovic, & Fern, 2017). The video would be segmented into coherent and semantically meaningful regions (Liu & He, 2015).

Using video semantic segmentation technique for Unmanned Aerial Vehicle (UAV) data processing is also a popular application. UAV equipped with cameras has been fast deployed to a broad range of applications, such as traffic monitoring, surveillance, crop dusting, entertainment industry and filming (Valavanis & Vachtsevanos, 2015). The superiority of UAVs can be summarized as the following three points: first, they could obtain images and videos from the dangerous and inaccessible areas where the manned vehicle cannot reach. Second, compared with the images captured from satellites and aircrafts, images from UAVs contain more details and have higher resolution. This is mainly because the distance between the UAVs and the interesting regions is much closer than the satellites and other aircrafts. Moreover, the cost of UAVs data is lower than most other methods because the price of the UAVs is not that high and the cost of the flying process is low. Thus, for those tasks in small or dangerous areas which require high resolution images and numerous information in details, UAVs are the best choice.

However, because of those specific characteristics of UAVs, the segmentation tasks of it also meet some different challenges from other data. The largest challenge for video semantic segmentation from UAV is the significant change of object appearance over the video frames. These changes are mostly caused by the following three reasons. The first one is the changes of the viewpoint of the UAV cameras. The direction and position of the UAVs would change through the whole flight which would cause the giant changes between the continuous video frames. For example, as we all know, the road is normally looks like a straight line in the video frame. The shape of the road is an important feature for segmentation. However, when the UAV flights across the road, there may be just part of the road left in some frames. And in those frames, the road loses the linear shape. As Figure 1.1 shows, in the left image which is the first frame, the road has a linear shape. However, in the right image which is the second frame, only a part of the road left and the shape of it is a triangle. The second reason for the giant change of object appearance is the illumination of the scene. The object has different appearances when it under the shadow and outside the shadow. And the last one is the deformation of the object. For example, the cars with their doors open look different in other frames when their doors closed. Thus, if we segment each frame individually, the

same object may have different classification results because of their different appearances in different frames.



Figure 1.1 The example of changes between the two different shapes of the road(Lyu, Vosselman, Xia, Yilmaz, & Yang, 2018)

Traditional methods for video semantic segmentation are just extending the semantic segmentation approaches for a single image to multiple frames (Mohammad, Kaloskampis, & Hicks, 2015). The temporal dependencies are ignored in these methods. In these approaches, the video is divided into several frames and is viewed as a collection of those frames. Each frame is considered as a static image and is segmented individually. For the segmentation task of each frame, it uses the same method as the single image segmentation. The largest limitation of these methods is poor temporal consistency. Because the frames are considered as single images, the segmentation processes are just according to these frames themselves. The segmentation result of each frame can be influenced easily by the changes of the viewpoint of the object, the changes of illumination, and deformation of the object. The same object may have different appearances in different frames and would lead to different segmentation result. Hence, the accuracy of these segmentations is relatively low. To keep the temporal consistency, the pixels of the same object in different frames should be assigned the same label. Beyond, there exists huge redundancy along the time axis when the motion field is smooth. This redundancy would make the efficiency of these traditional methods relatively low. Therefore, developing an appropriate video segmentation method using temporal information for UAV is a significant and important mission.

This research proposes an FCN + Conv\_LSTM framework for semantic video segmentation. The proposed algorithm tries to combine the FCN model and the Conv\_LSTM model. In this algorithm, the FCN model serves as the frame-based segmentation method which is used to segment each frame individually. The output of this part is the segmentation result of each frame or the feature map of each frame extracted by the FCN model. The Conv\_LSTM model serves as the post-processing method which makes use of the temporal information between consecutive frames. The inputs of this part are sequences formed by the output segmentation results or the sequences of the feature maps extracted by FCN model. Conv\_LSTM learn the temporal information of these sequences and could know how to combine the information of the frames of input sequences to predict the final segmentation results.

## 1.2. Research identification

This section lists the research objectives and research questions of this thesis. The answers to the research questions would be shown in section 5.

### 1.2.1. Research objectives

This research aims at developing an efficient and novel semantic video segmentation method based on neural network and make use of the temporal information to keep the temporal consistency.

The main research objectives of this study could be divided into the following three sub-objectives:

1. Explore the appropriate frame-based semantic segmentation method.
2. Develop the appropriate multi-frame segmentation method using temporal information.
3. Develop the method to combine the multi-frames to the whole network.

### 1.2.2. Research questions

Sub-objectives 1:

- How long is the interval between each frame extracted from the video?
- How many classes would be segmented?
- Which deep learning method would be chosen for the semantic segmentation of each frame, FCN, AlexNet or ResNet?
- How to extract the features of each frame and how to represent the features?
- How to measure the accuracy of the segmentation of each frame?

Sub-objectives 2:

- How to use temporal information?
- How to connect the sequential frames to several blocks?
- What is the size of each block?
- What is the size of the coverage between the sequential blocks?

Sub-objectives 3:

- Which method would be used to connect the blocks?
- How to evaluate the segmentation of the whole video?

## 1.3. Contributions

This thesis proposes a novel semantic video segmentation method based on neural network and makes use of the temporal information of videos to keep the temporal consistency. The main contributions of this thesis are shown in follows:

1. This thesis explored several deep-learning methods and chose FCN8s model as frame-based semantic segmentation method to get segmentation or extract features of each frame individually.
2. This thesis constructs an appropriate data structure to form the individual frames extracted from the videos as sequences.
3. This thesis proposes an FCN + Conv\_LSTM framework for semantic video segmentation which could make use of the temporal information with limited computation cost.

## 1.4. Innovation aimed at

The previous researches for the video segmentation from UAV mainly focused on the frame-based method. These methods are just extending single image segmentation approaches to multiple frames. The

temporal information is omitted in these methods. Thus, the temporal consistency of the result is poor. This study would focus on the temporal consistency and make use of the temporal information to make the segmentation coherent through the video and get higher accuracy.

### **1.5. Outline**

This thesis contains 5 chapters, the roles of each chapter are described as follows:

1. Chapter 1 introduces the motivation of this thesis; identifies this research; lists the contributions; shows the innovations and gives the overview of the whole thesis.
2. Chapter 2 reviews the related work of semantic video segmentation.
3. Chapter 3 describes the frameworks of the methods used in this research and explains the details of these methods.
4. Chapter 4 presents the details and results of the experiments and gives some discussion according to the results.
5. Chapter 5 concludes this research, demonstrates the advantages and disadvantages of this research, answers the research questions and plans the future work of this researching problem.

## 2. RELATED WORK

Semantic video segmentation aims at separating the foreground objects of the video from the background and cluster them into different classes. It is a kind of semantic segmentation question. This chapter reviews the related work of semantic video segmentation. Firstly, section 2.1 generally introduce traditional learning algorithms before deep learning methods. Secondly, section 2.2 reviews the methods using deep learning for semantic segmentation. Thirdly, section 2.3 introduces the deep learning approaches using temporal information for semantic video segmentation.

### 2.1. Methods before deep learning

Before the GPU was widely used, the computational ability was poor and the research of deep learning didn't explode. The most popular learning algorithms are Support Vector Machines (SVMs) and Conditional Random Fields (CRF).

SVM was proposed by Cortes and Vapnik in 1995. It is a supervised learning model based on hand-engineering features. It could predict the label of each pixel given the set of features. The results of this model are highly related to the feature used. Because the features becoming more and more complex, and the data becoming larger and larger, this model cannot satisfy the requirement of modern tasks.

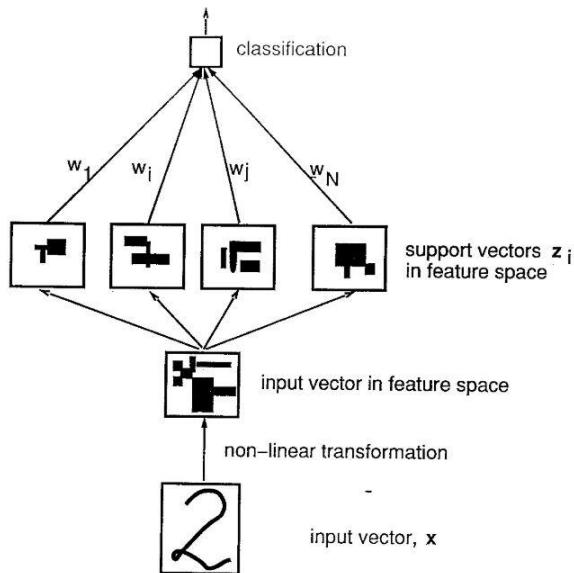


Figure 2.1 Support Vectors Network (Cortes & Vapnik, 1995)

CRF is also a supervised learning model. John Lafferty et al. (2001) presented CRF for segmenting and labeling data. The largest difference between SVM and CRF is that CRF could use the probability of a label plus other labels of the preceding observation.

Badrinarayanan, Galasso and Cipolla (2010) proposed a Label propagation method for semantic video segmentation using a coupled-HMM model. Figure 2.2 shows how this label propagation method works. Both ends of the video frames have annotation images, this method could propagate the labels to the

unlabelled frames between two ends. However, the length between the two ends of the video sequence could not be too long for better accuracy.

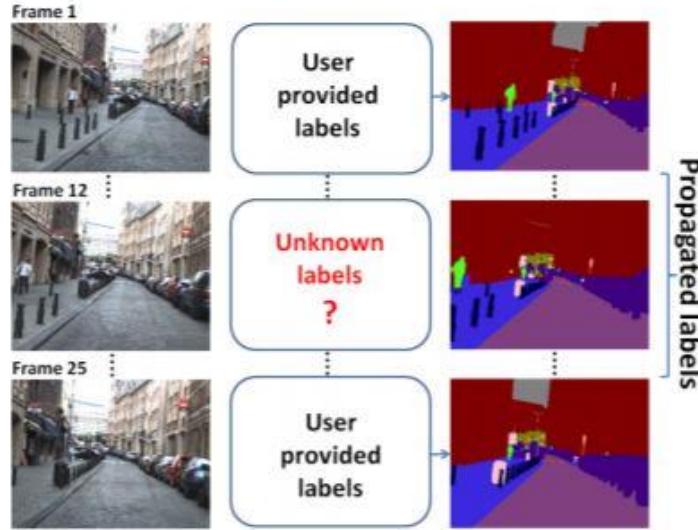


Figure 2.2 Label propagation in video (Badrinarayanan et al., 2010)

To extend the label propagation model, a method using tree structure was proposed by Badrinarayanan, Budvytis and Cipolla (Mustikovela, Yang, & Rother, 2016). The tree structure of this method not only contains the relation between the two labeled ends but also contain the relation between the unlabelled frames of the sequence. This method could extend the time-window but not good at the class with small size.

Furthermore, they extend the tree structure model to a mixture of tree probabilistic graphical model which links the super-pixels of each frame through the whole video sequence using tree structure (Badrinarayanan, Budvytis, & Cipolla, 2014). Figure 2.3 shows an example of the structure of this model. The points are super-pixels. This method could offer great segmentation accuracy but need high computational cost.

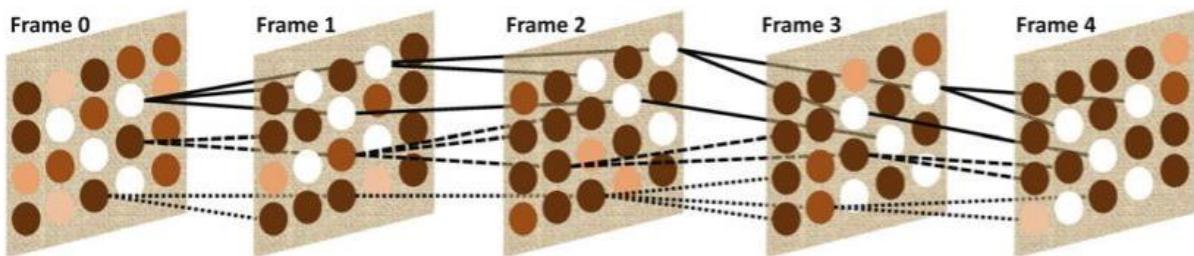


Figure 2.3 Example of the mixture of tree model with three component temporal trees(Badrinarayanan et al., 2014)

## 2.2. Deep learning methods for semantic segmentation

This section introduces the most popular deep learning networks CNN and FCN which were applied in many semantic segmentation methods.

### 2.2.1. CNN

Convolutional Neural Network (CNN) is a kind of end to end learning method. Alex Krizhevsky (2012) trained a CNN based deep learning model called AlexNet which seems like one of the milestones in the deep learning field. The AlexNet contains five convolutional layers and three fully connected layers. They used ReLu for the non-linearity function and used data augmentation and dropout to reduce the overfitting (Krizhevsky, A., Sutskever, I., & Hinton, 2012). This model could achieve much higher accuracy than the traditional hand-engineering approaches. It became one of the most influential models in video segmentation area and has been modified as the encoder in many cases.

The Visual Geometry Group (VGG) from the Department of Engineering Science, University of Oxford trained a CNN model called VGG. This model needs fewer parameters because its first layer contains a stack of convolution layers and small size receptive fields which makes the model more discriminative and easier to train (Garcia-Garcia et al., 2018).

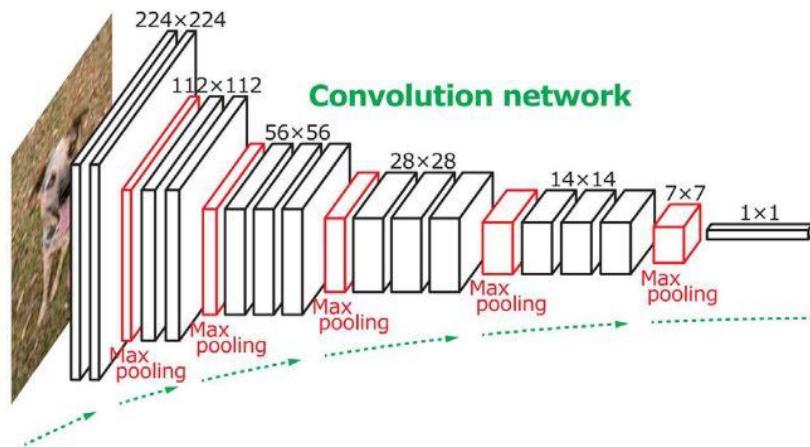


Figure 2.4 VGG-16 architecture (Noh, Hong, & Han, 2015)

AlexNet was the winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012. VGG is the first runner-up of ILSVRC in 2013. And in 2014, the winner is GoogLeNet which was proposed by Szegedy (2015). This network introduces a new efficient architecture called Inception module. The main idea of this model is discovering how to approximate the most favorable local sparse structure. The whole network contains 27 layers (including pooling layers) and the inception modules are stacked on the top of the network. This special architecture could save the computational cost because of the parameter reducing.

After the GoogLeNet, another remarkable method was proposed by He (2015) which named ResNet. It has a very deep depth with 152 layers. It also introduced a new architecture called residual learning block. It could be considered as shortcut connection which could skip one or more layers by copy the inputs. This kind of architecture could help to overcome the vanishing gradient problem (Garcia-Garcia et al., 2018).

### 2.2.2. FCN

One of the most popular deep neural networks applies CNN is the Fully convolutional network (FCN). It is the state-of-the-art techniques for video segmentation(Long, Shelhamer, & Darrell, 2015a). It was proposed by Long et al. (2015) for learning per-pixels tasks. They transformed many well-known models such as AlexNet (Krizhevsky, A., Sutskever, I., & Hinton, 2012), ResNet (Mingus, Herd, O'Reilly, Jilk, & Wyatte, 2016) and GoogLeNet (Szegedy et al., 2015) into fully convolution ones to take advantage of existing CNNs as powerful visual models that are able to learn hierarchies of features (Garcia-Garcia et al., 2018). FCN could take the arbitrary size of input and produce correspondingly-sized output with efficient inference and learning (Long et al., 2015b). The classification level of FCN is pixel level. The typical FCN structure is shown in Figure 2.5.

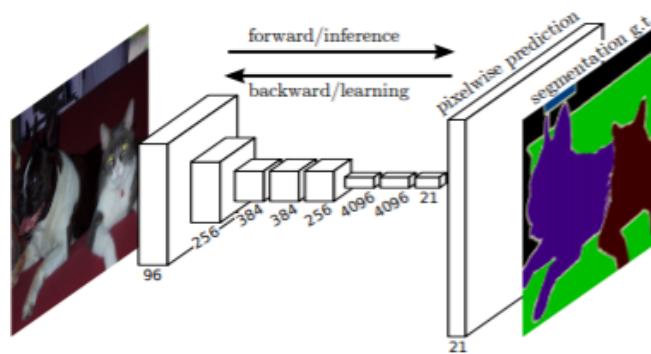


Figure 2.5 The architecture of FCN (Long et al., 2015a)

Specifically, there are several different FCN architectures. Long (2015) also proposed skip connection architectures in FCN such as FCN8s, FCN16s, and FCN32s. These models could skip some layers and fuse the features from different layers for better segmentation. For example, FCN8s model is based on the VGG-16 network and add a skip from pool3 at stride 8 to get more information from the global structure. The skip connection of FCN model is shown in Figure 2.6.

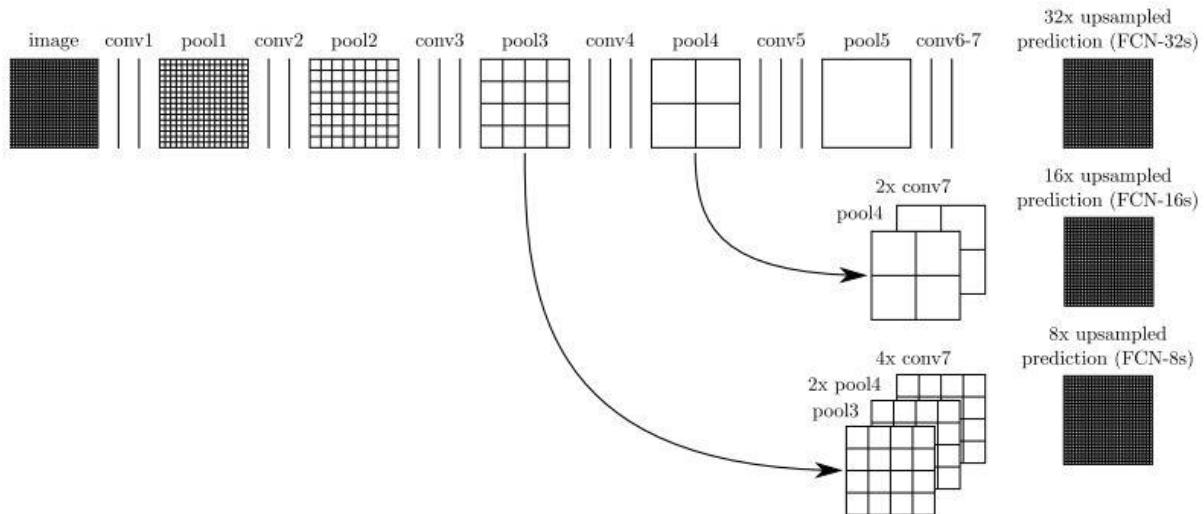


Figure 2.6 Architecture of FCN8s, FCN16s and FCN32s.(Long et al., 2015b)

In order to map low-resolution features to input resolution better, Badrinarayanan (2017) proposed an efficient FCN architecture for pixel-wise semantic segmentation which named SegNet. It contains an encoder network which is similar to the traditional FCN model and a corresponding decoder network which uses the pooling indices from pooling step to upsample the low-resolution input feature maps. It is also based on the VGG-16 network. The first 13 convolutional layers are as same as the first 13 convolutional layers of VGG-16 networks while the following layers are not fully connected layers but corresponding decoder layers. The architecture of SegNet is shown in Figure 2.7. Because its novel architecture this network is efficient in computation.

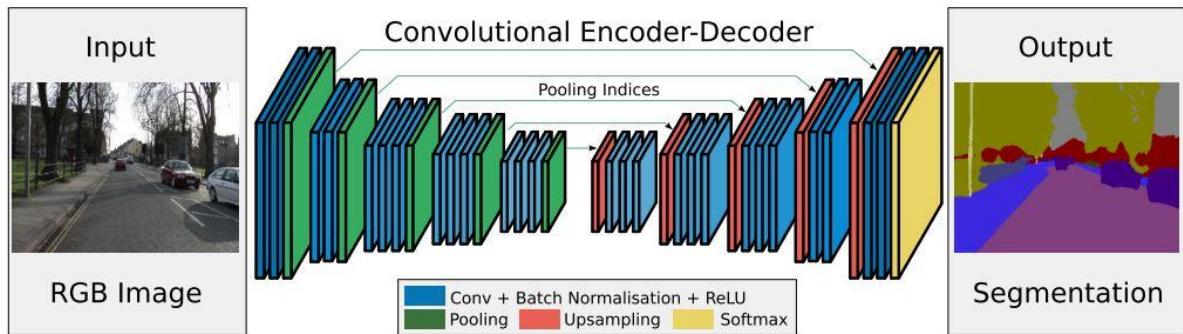


Figure 2.7 Architecture of SegNet

### 2.3. Methods using temporal information

This section introduces the methods using temporal information including the optical flow, RNN, LSTM, and Conv\_LSTM.

#### 2.3.1. Overview

Data like videos do not only contain the feature information of each frame but also contains the temporal information between frames. Traditional image segmentation methods, such as CNN and FCN are mainly focusing on the segmentation of individual frame and are not very suitable for learning sequences (Srinivas et al., 2016). The segmentation result of each frame can be influenced easily by the changes of the viewpoint of the object, the changes of illumination, and deformation of the object. The same objects may have different appearances in different frames and would lead to different segmentation result. Hence, the accuracy of these segmentations is relatively low. In order to solve this problem, several approaches using temporal information have been proposed.

#### 2.3.2. Optical flow

Varun Jampani et al. (2016) use optical flow and propagate information forward through video by combining the training of CNN between consecutive frames. The optical flow is the description of the movement of the brightness patterns of the objects between the consecutive frames (Horn & Schunck, 1981). Jang and Kim (2017) develop the convolutional trident network (CTN) and propagate the segmentation labels at the previous frame to the current frame using optical flow vectors. This method makes use of the temporal information between the consecutive image and could offer great segmentation result. However, the calculation of optical flow costs time and computation memory.

### 2.3.3. RNN

Recurrent Neural Networks (RNNs) are neural networks with a looping structure. This kind of neural network contains feedback session which enables it to have a memory of previous states and to learn temporal patterns in data (Srinivas et al., 2016). This feedback session is realized by the recurrent cells which contain the hidden layer, input layer, and output layer. The outputs of this network are based on the input and the previous memory from the hidden layer we call it hidden units. The RNN network has been applied in many other approaches. Pinheiro (2014) proposed an approach consisting of RNN to consider a large input context, at the same time, limiting the capacity of the model. Visin et al. (2015) proposed a structured prediction architecture based on RNN named ReSeg. It exploits CNN to extract the local features and RNN to retrieve the long distance pixel dependencies (Visin et al., 2016). However, the data flow between the recurrent cells in the RNN network leads to the vanishing and exploding gradients problems (Bengio, Simard, & Frasconi, 1994). Figure 2.8 shows an example of the structure of a single recurrent network.

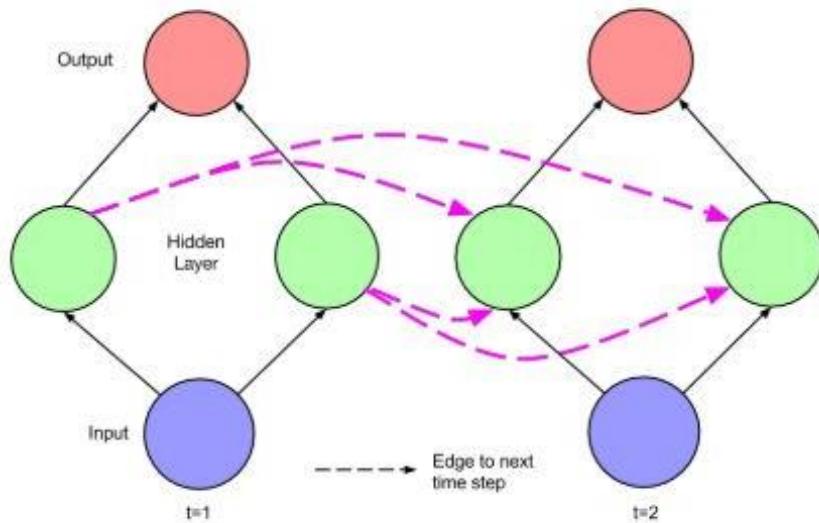


Figure 2.8 Example of a single RNN (Lipton, Berkowitz, & Elkan, 2015)

### 2.3.4. LSTM

To solve the problem of the vanishing and exploding gradient problems in RNN, some approaches using gated structures have been introduced. The gates can take control of the backpropagation process of the network (Valipour, Siam, Jagersand, & Ray, 2017). Long Short Term Memory (LSTM) is one of these approaches with gated structures. LSTM units called Constant Error Carousels to overcome the vanishing gradient problem in RNN (Srinivas et al., 2016). It contains three gates including input gate, output gate and forget gate. Each gate could learn and get weights. These gates help the network to choose the important information to remember and determine how to combine the memory and current input. Byeon (2015) investigated an LSTM to take account the local (pixel by pixel) and global (label by label) dependencies.

### 2.3.5. Conv\_LSTM

LSTM networks have been modified and applied in many other tasks. The representative LSTM application is the Convolution Long Short Term Memory (Conv-LSTM). It has convolutional structures in both the input-to-state transition and the state-to-state transition (Shi et al., 2015). And it could help to keep the temporal consistency. The convolution LSTM contains two key components, one is the convolutional layers another is the LSTM cells in RNN model. Both the input-to-state transition and the state-to-state transition have the convolutional structures (Shi et al., 2015).

## 3. METHOD

This chapter explains the key methods used in this research. Section 3.1 gives an overview of the whole method. Section 3.2 introduces the two frameworks of these methods. Section 3.3 explains the segmentation method for each frame and the process of feature extraction. Section 3.4 explains how to form the frame sequences. Section 3.5 introduces segmentation methods using temporal information. Section 3.6 introduces the loss function. Lastly, in section 3.7, the accuracy metric is explained.

### 3.1. Overview

This research aimed to develop an automatic semantic video segmentation method which could produce the segmentation result of the whole video and keep the temporal consistency at the same time. Traditional methods just extended single image segmentation approaches to multiple frames. The processing of segmentation results is just depending on the features of these frames themselves and didn't take the temporal information into account. The changes of the objects which mainly caused by the changes of the viewpoint of the object, the changes of illumination, and deformation of the object can easily influence the segmentation result of each frame. The different appearances of the same object in different frames would lead to different segmentation results. Hence, the accuracy and the temporal consistency of the segmentation video is relatively low.

To solve this problem, this thesis tries to combine the FCN network with Conv\_LSTM network to get the FCN+Conv\_LSTM network. The FCN is a widely used semantic segmentation method. And the Conv\_LSTM could make use of the temporal information.

The FCN model is implemented in TensorFlow. The Conv-LSTM model is implemented in Keras and use TensorFlow as the backbone.

### 3.2. Proposed frameworks

Two methods are proposed to combine the FCN network and the Conv\_LSTM network.

The first one is using the segmentation result of the FCN network as the bond. First, using FCN to segment the input images and send these outputs as the inputs of Conv\_LSTM network. Which means the segmentation result of the FCN network would be formed as sequence and sent into the Conv\_LSTM network.

The second method used to combine these two networks is using the feature map extracted by the FCN network as the bond. Firstly, extract the features of the input images by the FCN. The outputs of this step are feature maps of input images. Then form these extracted feature maps as sequences. After that, these feature map sequences are used as the input of the Conv\_LSTM networks.

Figure 3.1 demonstrates the framework of the first method.

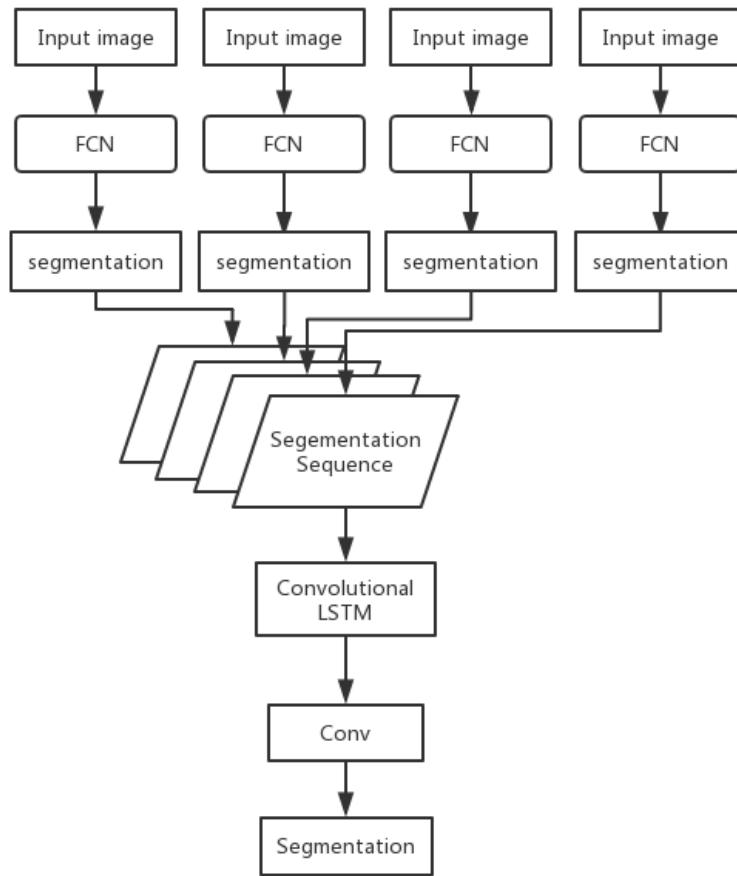


Figure 3.1 Framework of method 1

- 1) The input data of this framework are 4 images extracted from the same video. The interval between these frames is fixed. We use the FCN model as the frame-based segmentation method to get the segmentation result of these images individually. The outputs of this phase are segmentation results of the input images.
- 2) Order the output images of the FCN model from the earliest to the latest and form them as segmentation result sequences. The output segmentation images have 3 dimensions (width, height, band). The number of bands depends on the type of segmentation images. For example, when the segmentations are RGB images the band's number is 3. When the segmentations are one hot image, the band's number is equal to the class numbers. We combine these consecutive frames to form a 4 dimensions tensor (times, width, height, band). The number of the first dimension is the number of the length of the sequence
- 3) Input the segmentation result sequences to the convolution LSTM network. The corresponding ground truth images are also formed as the same sequences. The convolution LSTM contains two key components, one is the convolutional layers another is the LSTM cells in RNN model. Both the input-to-state transition and the state-to-state transition have the convolutional structures (Shi et al., 2015).
- 4) Finally, after the training of the Conv\_LSTM model, we could predict the input images through the FCN model and the Conv\_LSTM model to get the final segmentation result.

The second method uses FCN to extract the feature from the input images and send these feature maps to the Conv\_LSTM model. Figure 3.2 demonstrates the framework of the second method.

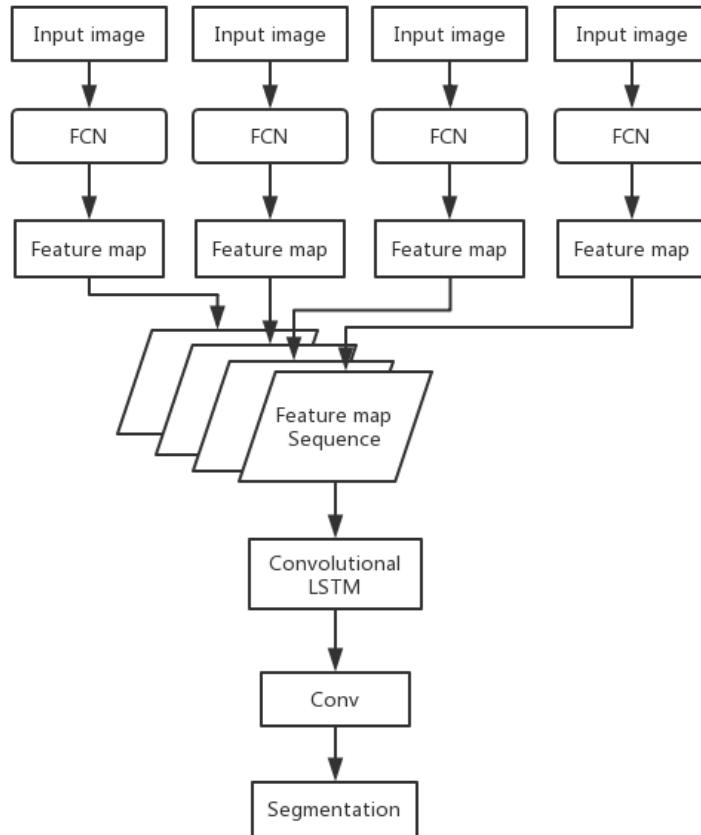


Figure 3.2 Framework of method 2

- 1) The input data of this framework is the same as the first method which is 4 images extracted from the same video with the fixed interval. We use the FCN model as the frame-based segmentation method to extract the features from the input images individually. The outputs of this phase are feature maps of the input images.
  - 2) Order the feature maps from the FCN model from the earliest to the latest and form them as feature maps sequences. The output feature maps have 3 dimensions (width, height, band). The number of the bands depends on the layers which the feature maps extracted from. The FCN network contains several layers, the depth of the feature maps is different. We combine these consecutive feature maps to form a 4 dimensions tensor (times, width, height, band). The number of the first dimension is the number of the length of the sequence.
  - 3) Input the feature maps sequences to the convolution LSTM network. The corresponding ground truth images are also formed as the same sequences. The convolution LSTM contains two key components, one is the convolutional layers another is the LSTM cells in RNN model. Both the input-to-state transition and the state-to-state transition have the convolutional structures (Shi et al., 2015).
  - 4) Finally, the tensor which has the same shape of the input image would go through the convolution layer and we could get the final segmentation result.

### 3.3. Frame-based segmentation and feature extraction

The first part of the proposed framework is the frame-based segmentation method. Frame-based method means this method is used on the individual frames.

The frame-based segmentation method is used in method 1 for getting the segmentation results of the input images. And in method 2, this method is used for extracting feature maps of the input images.

#### 3.3.1. FCN

FCN8s model is chosen as the frame-based semantic segmentation method. FCN could take the arbitrary size of input and produce correspondingly-sized output with efficient inference and learning (Long et al., 2015b). And it is able to learn hierarchies of features (Garcia-Garcia et al., 2018a) and offer the pixel level classification. FCN8s is a kind of FCN model with skip connection structure. It added a skip from pool3 at stride 8 which make it could get more information from the global structure.

The structure of FCN8s model is shown in Table 3.1. To simplify the table, the table only contains the convolutional layers and the deconvolutional layers, the dropout layers, pooling layers, and softmax layers are omitted.

Layer name	Number of filters
Conv_1	64
Conv_2	128
Conv_3	256
Conv_4	512
Conv_5	512
Conv_6	4096
Conv_7	4096
Conv_8	8
Deconv_1	8
Deconv_2	8
Deconv_3	8

Table 3.1 Structure of FCN8s Network. (Doesn't include the dropout layers, pooling layers, and softmax layers)

#### 3.3.2. Feature extraction

The weight of the FCN model would be updated during the training. When the training is finished, the saved weight could be used to predict the testing images and get the segmentation results. These weights could also be used for feature extraction. Each layer of the FCN model has its weights, and the output of these layers are the feature map we want to extract. Using the weights of several layers of the trained FCN model, it is easy to get the feature maps.

### 3.4. Sequence construction

The input of the Conv\_LSTM network should be 5 dimensions tensors. These 5 dimensions are samples, time, width, height, filters. To get the 5 dimensions tensor we need to first transform the individual images into sequences. The images have 3 dimensions which are width, height and band. The number of bands depends on the type of images. When the images are segmentation results, the number of bands could be 3 if the images are RGB images or 1 if the images are gray images. When the images are the feature maps, the number of the bands is up to the layers which the feature maps extracted from. We combine these consecutive feature maps together to form a 4 dimensions tensor. These 4 dimensions are times, width, height and band. The number of the first dimension is the number of the length of the sequence.

These 4 dimensions sequences are named as blocks in this thesis. For each video, we could extract several sequences. When the length of the sequences is more than the length of the block, it is possible to form more than one blocks of one sequence. To connect these consecutive blocks together, we would like to form a 5 dimensions tensor with the 5 dimensions samples, times, width, height, and band.

Because of the limitation of computation ability, the sequence length we set in this study is 4 frames. The Conv\_LSTM could use the features from the first three frames and the temporal information between these three frames to predict the last frame.

In order to avoid the condition that some of the frames don't been used. We need to set some overlaps between the sequences. We call the 4 frame sequences blocks. And each block has 3 frames overlap. So that we could get the predict of every frame except the first 3 frames.

The structure of the sequences is shown in Figure 3.3.

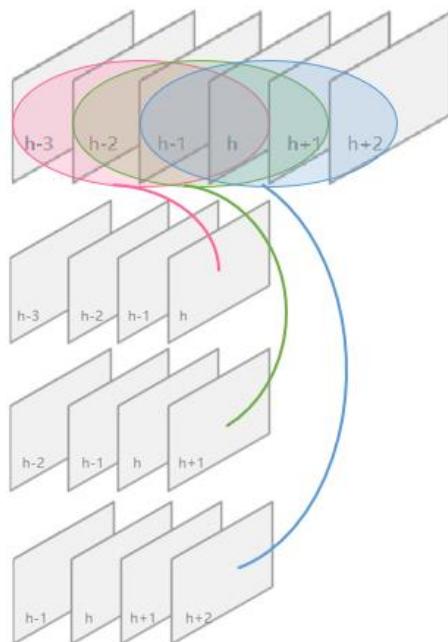


Figure 3.3 Formation of blocks and sequence

### 3.5. Convolutional LSTM

Convolutional LSTM is used to make use of temporal information. For method 1, the inputs of this network are sequences formed by the segmentation results extracted from the FCN-8s model. For method 2, the inputs are sequences formed by the feature maps extracted from the original images. This model could learn the temporal dependencies among the frames of the sequences. The convolution LSTM contains two key components, one is the convolutional layers another is the LSTM cells in RNN model. There are 3 gates in the LSTM cells, input gate, output gate and forget gates. In the process of network training, these 3 gates update the weight and could help the network to choose the important information to remember and determine how to combine the memory and current input.

The structure of Conv\_LSTM is shown in Figure 3.4.

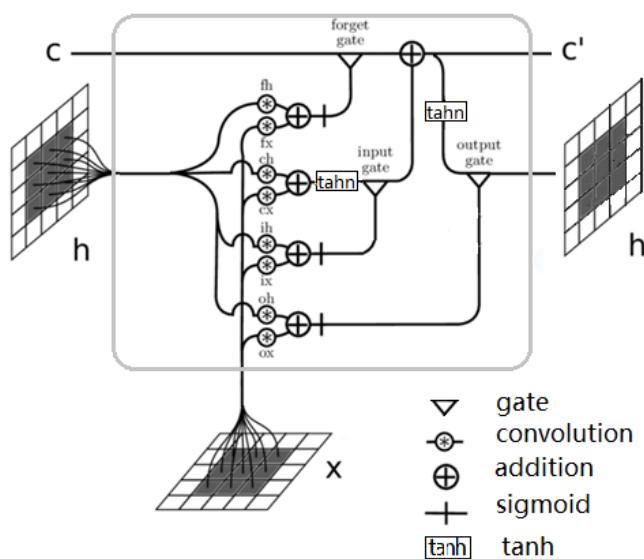


Figure 3.4 Conv\_LSTM structure

### 3.6. Loss function

We used categorical cross-entropy as the loss function in the implementation of the Conv\_LSTM model. It is a special kind of the cross-entropy loss function which could also be called Softmax Loss. It is defined as follows:

$$L(x) = H(p, \hat{p}) = - \sum_{c \in C} p(x=c) \log \hat{p}(x=c) \quad (1)$$

$p$  means the true probability and  $\hat{p}$  means the predicted probability. When we use this loss function, the format of the input data should have the same dimension as the number of the class which could be called as a categorial format.

### 3.7. Accuracy metric

To evaluate the experimental results, using the two trained model to segment the test image. The results would be compared with the ground truth images. Intersection over union (IoU) score is defined as follows:

Intersection over Union (IoU) score is used. It is defined as follows:

$$\text{IoU} = \text{TP}/(\text{TP} + \text{FP} + \text{FN}) \quad (2)$$

Where TP denotes true positive, FP denotes false negative and FN denotes false negative.

The mean IoU of these two models would be calculated to evaluate the accuracy of these two models.

## 4. EXPERIMENTAL RESULTS

This chapter demonstrates the details of the experiments setting including the preparation of the dataset, the setting of the two networks and the results of the different methods.

### 4.1. Dataset

The dataset used in this method is 10 videos captured by UAVs(Lyu et al., 2018). These videos were captured by UAVs in Wuhan, China from June to August 2017 and in Gronau, Germany in May 2017. We extracted 27 image sequences from the UAV videos. Totally 260 images. The extraction interval is 150 frames.

The size of these images is  $3840 * 2160$  pixels or  $4096 \times 2160$  pixels. Because of the limitation of the computation ability, these images were resized to  $512 * 288$  pixels. Normally, there are two methods to get small images from the large one, resizing method and cropping method. The cropping method could offer better image quality and get higher accuracy. It is cropping the large images to several small images, the resolutions don't change while the global feature would lose. However, the input of the Conv\_LSTM model should be sequences of frames. The global feature plays an import role in its training process. Especially in this study, the extraction interval is about 5 seconds. The movements of some objects in 5 seconds are large. For example, the location of the cars would change a lot. Also, the focus area of continuous frames also changes. The temporal consistency and temporal information among the crop image sequences are poor. At the same time, the resizing method could keep the global feature and temporal information even though it would lose some details. That's the reason for choosing the resizing method to get the small size images.

The dataset is divided into four parts. 130 images are used to train the FCN model, 75 images are used to train the Conv\_LSTM model, 45 images are used for testing and 10 images are used for validating. Because most of the videos were captured in Wuhan, only two videos were captured in Gronau. The images of Gronau are not included in test dataset.

#### 4.1.1. Annotation

The annotation tool is provided by Ye Lyu, a doctor candidate from ITC. In this research, these images are annotated into 5 classes including 4 foreground classes (building, road, cars, vegetation) and 1 background class (clutter). The road class doesn't include the bike lane and the walk lane. Those are labeled as clutter. The cars class include static cars and moving cars. The vegetation class includes the trees, shrub, and grass. Figure 4.1 shows an example of annotation.

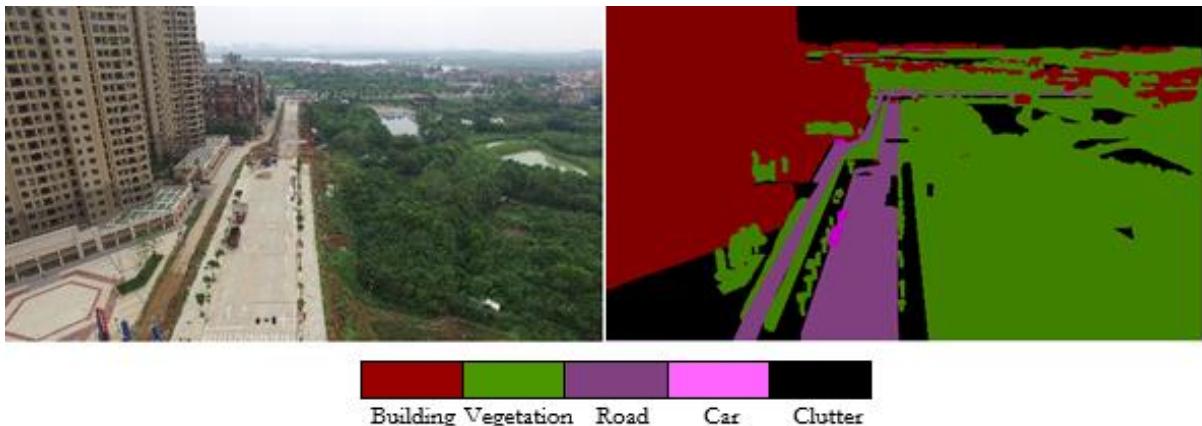


Figure 4.1 Example of annotation

#### 4.2. Parameter setting

The proposed method contains two networks, we first train the FCN model with the extracted images from UAV videos. The sizes of the original images are  $3840 \times 2160$  pixels or  $4096 \times 2160$  pixels. To save the GPU memory, the images are resized to  $512 \times 288$  pixels. The reason why using the resize method not crop method is explained in section 4.1. The ground truth images are also resized to the same size as the corresponding original images. This model is implemented in TensorFlow.

We train the Conv-LSTM model with the sequences formed by the output segmentation results of FCN model or the feature maps extracted by the FCN model. Due to the limitation of the GPU memory, the segmentation results keep the  $512 \times 288$  pixels size and these sequences are clipped into several blocks. The length of each block is 4 frames. The overlap of the consecutive blocks is 3 frames. The ground truth images are also formed in the same way.

The Conv-LSTM model is implemented in Keras and use TensorFlow as the backbone.

#### 4.3. Implementation details

The first step of this research is annotating the images. After that, the original images and their corresponding ground truth images are resized to the appropriate size. The resized process is realized by the python code.

The original images and ground truth are sent into the FCN model for training. The learning rate is set as 0.001, Adam is used as optimizer. Training of the FCN model is done with 1 image per batch. Then use it to get the segmentation result of the original images. And, it could also be used to extract the feature maps of the original images. These segmentation result images and feature maps are sent to the Conv\_LSTM model. The optimizer used in Conv\_LSTM model is Adadelta and the learning rate is 0.01.

Both the segmentation result images and the feature maps have 3 dimensions (width, height, band). The number of the bands depends on the type of the segmentation images or the extraction layers of the feature maps. These images are transferred into one-hot images and be formed as form 4 dimensions (times, width, height, band) sequences with the order from the earliest to the latest. After that, these sequences are combined to get the 5 dimensions (samples, times, width, height, band) sequences. The corresponding ground truth images are also formed as the same sequences.

These sequences are sent to the convolution LSTM network. After the training of the Conv\_LSTM model, we could predict the input images through the FCN model and the Conv\_LSTM model to get the final segmentation result.

#### 4.4. Comparison between FCN and method 1

45 images are used for testing. For the Conv\_LSTM model, these 45 images could form 7 sequences and get 24 outputs. We calculate the IoU of 5 specific classes and the mean IoU of all classes to compare the segmentation results of the FCN model and the segmentation results of method 1 model. The IoUs are shown in Table 4.1. The segmentation results of these two models are shown in Figure 4.2.

From Table 4.1, we can see the proposed method 1 provides better mean IoU than the FCN8s method. The mean IoU improves almost 2%. The IoU of road, cars and clutter increase and could provide better visualization in the segmentation results. The IoU of road increase by 2.6%, the IoU of cars increase by 0.23% and the IoU of clutter increase by 12.28%.

From Figure 4.2, we can see the result of FCN model wrongly assign the road label to the clutter object. From the original images, it is easily to find that those areas have the similar color and the similar shape to the road. It may produce some trouble for the FCN model to distinguish them and cause the wrong segmentation in FCN results. The Conv\_LSTM model could combine the information from the previous frames and could correct these mistakes in some areas.

However, at the same time, if the whole sequences have the same segmentation mistakes at the same objects, the Conv\_LSTM model cannot figure those mistakes because those mistakes have the temporal consistency too. And in some cases, different frames of the sequences have different segmentation mistakes in different areas and in different objects which may add more noisy information to the final prediction and cause more segmentation mistakes in the result. And this may be the reason why the IoU of building and vegetation decrease.

classes	FCN	Method 1
Building	61.90	59.10
Vegetation	68.45	66.14
Road	43.24	45.88
Car	0	0.23
Clutter	16.17	28.45
Mean IoU	37.95	39.96

Table 4.1 IoU scores for FCN8s model and method 1

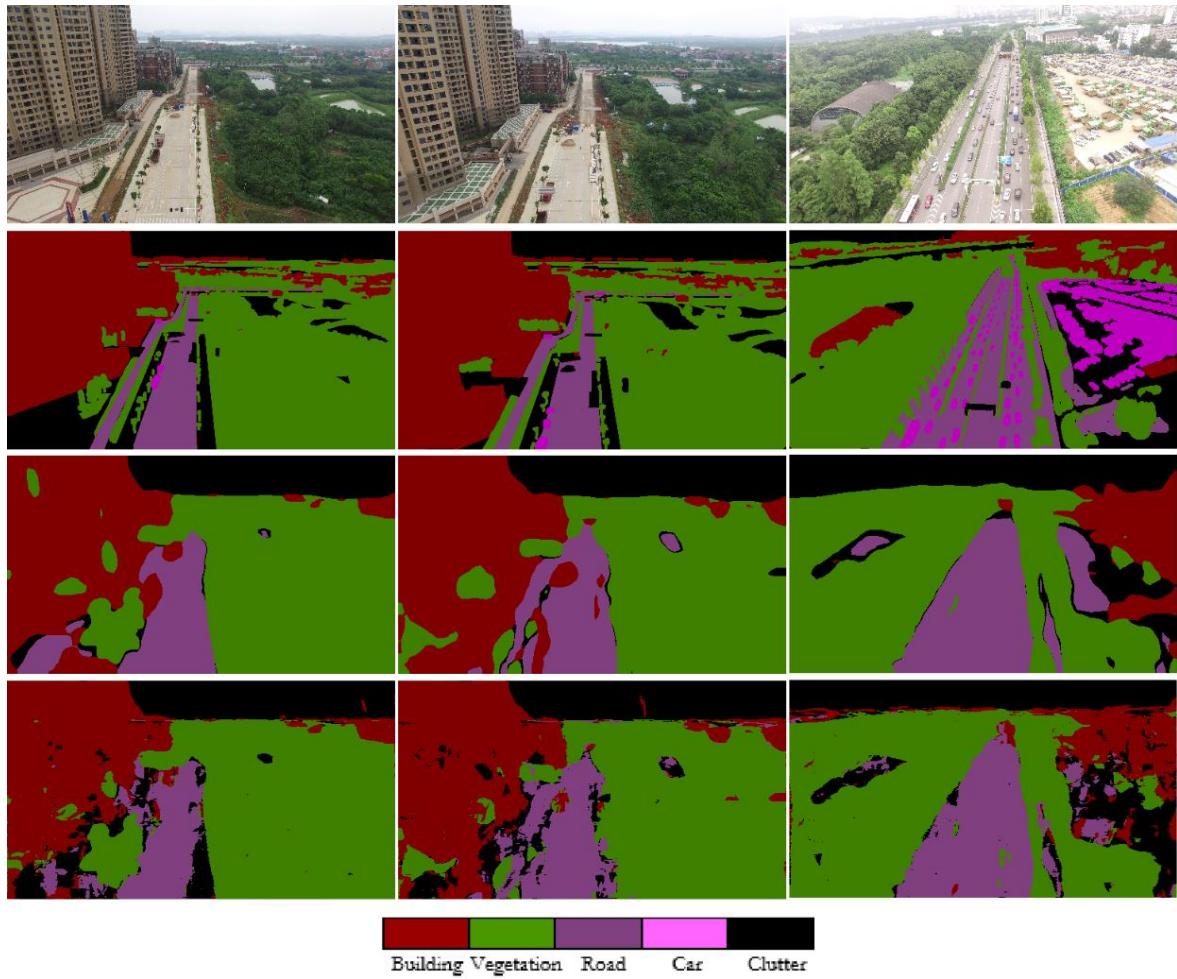


Figure 4.2 Comparison between FCN and method1. The first row shows the original image extracted from the video. The second row shows the corresponding ground truth images. The Third row shows the segmentation results of the FCN model. The fourth row shows the segmentation results of method 1.

#### 4.5. Comparison between FCN and method 2

Table 4.2 presents the results of FCN model and method 2. The IoU of 5 specific classes and the mean IoU of all classes are calculated to compare the segmentation results of these two models. The example of the segmentation results of these two models is shown in Figure 4.3.

Table 4.2 shows that the mean IoU of method 2 is increased by 2.83% from the FCN method which shows the superiority of the proposed method2. The IoU of road, cars and clutter increase and could provide better visualization in the segmentation results. The IoU of road increase by 3.37%, the IoU of cars increase by 1.97% and the IoU of clutter increase by 15.44%.

The class clutter improves a lot and it may because the Conv\_LSTM could make use of the temporal information and fix the wrong segmentation of the clutter objects in the FCN model. As it is shown in Figure 4.2Figure 4.3, a lot of clutter objects was wrongly labeled as road and building because of the similarity of the color and shape. Because the images we used in this study is resized from  $3840 \times 2160$  pixels to  $512 \times 288$  pixels. The resolution of the images is decreased a lot and some details are lost. For example, in the third line of the Figure 4.3, the cars in the images are difficult to recognize because of the poor resolution. While when the UAV flights close to the parking lot in the right part of the image, the cars in the images would be clearer so that some of the frames could recognize the cars. And that's may be the reason why Conv\_LSTM model could offer higher accuracy in the class car.

The IoU of building and IoU of vegetation are decreased. This may because the building and the vegetation could have variable shapes. For example, in this study, in the right part of the first two lines and the left part of the third line of Figure 4.3, the shape of the vegetation and building are irregular. The previous frames of the sequences may bring more noisy information to the final prediction and cause more segmentation mistakes in the result.

classes	FCN	Method 2
Building	61.90	58.58
Vegetation	68.45	65.14
Road	43.24	46.61
Car	0	1.97
Clutter	16.17	31.61
Mean IoU	37.95	40.78

Table 4.2 IoU scores for FCN8s model and method 2

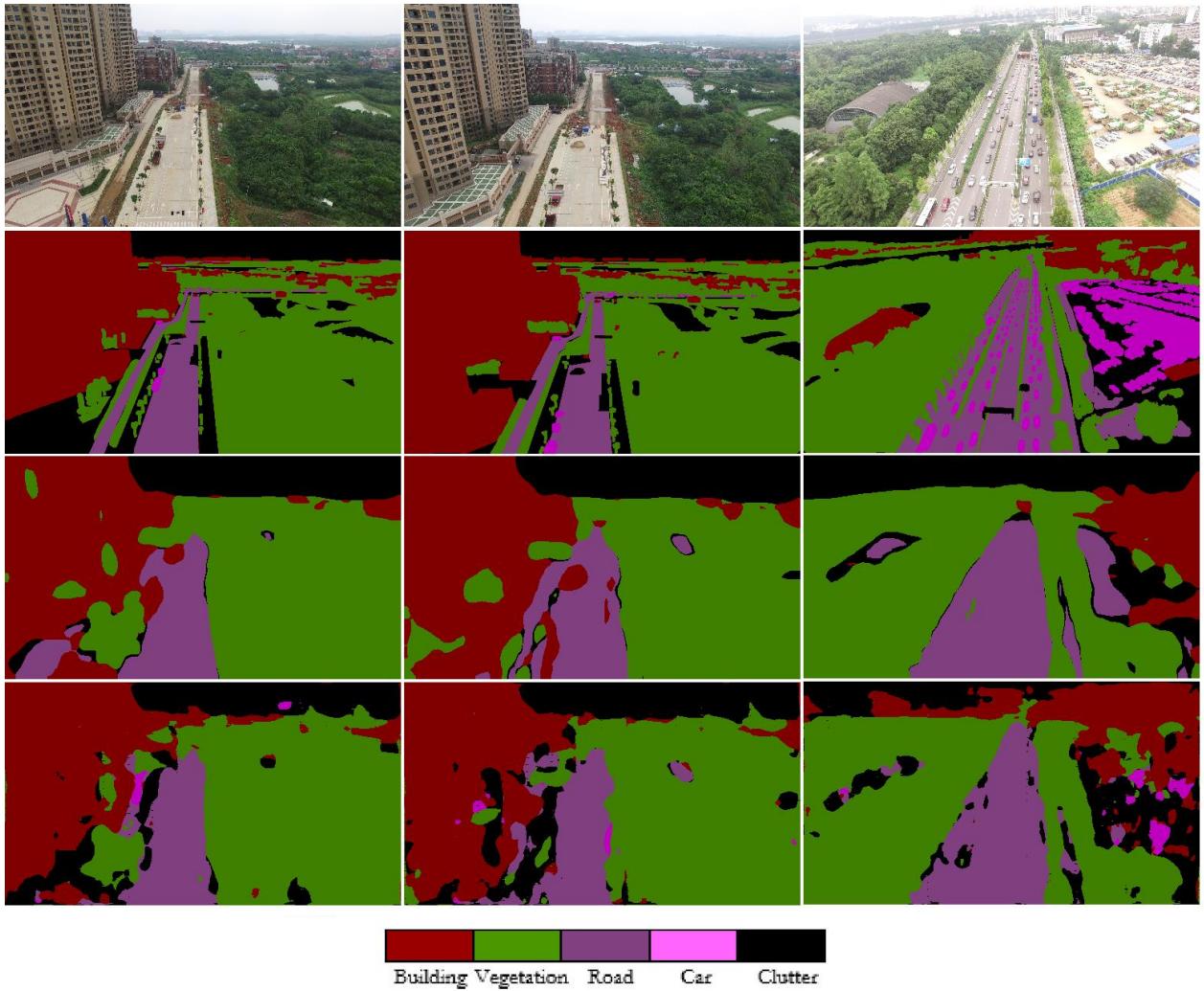


Figure 4.3 Comparison between FCN and method 2. The first row shows the original image extracted from the video. The second row shows the corresponding ground truth images. The Third row shows the segmentation results of the FCN model. The fourth row shows the segmentation results of method 2.

#### 4.6. Comparison between method1 and method 2

In this section, we compare the results of method 1 and method 2 and discuss the influence of the input data of Conv\_LSTM model. 7 sequences which include 45 images are used as a testing dataset. and get 24 outputs. The IoU of 5 specific classes and the mean IoU of all classes are shown in Table 4.3. The segmentation results of these two models are shown in Figure 4.4.

The difference between method 1 and method 2 is the input data of Conv\_LSTM model. Segmentation results which only contains the class label in each pixel are used in method 1. Feature maps which contain the feature information of each pixel are used in method2. The segmentation results make full use of the segmentation ability of FCN model but containing very limitation information. The feature maps containing more information while the segmentation ability of FCN is not used enough.

From Table 4.3, we can see the method 2 provides better mean IoU than method 1 which means the feature maps are more suitable for Conv\_LSTM training. This may because the feature maps could offer more information for Conv\_LSTM and the model could learn more. The IoU of road, cars and clutter increase and could provide better visualization in the segmentation results. The IoU of road increase by 0.8 %, the IoU of cars increase by 1.74% and the IoU of clutter increase by 3.16%. The IoU of the Building was decreased by 0.52% and the IoU of the vegetation is decreased by 1%. The increase and decrease trends are similar to the trends of these two methods compared with FCN model which may because method 2 could learn temporal information better. Therefore it could improve more in those classes which could get useful information from the previous images and at the same time be influenced more in those classes which get more noisy information from the previous images.

From Figure 4.4, we can see the result of method 1 contains a lot of small holes, especially in the road and building. This may be caused by the poor of the information of the input data. The input data of method 2 contains more information and could offer smoother results. Most of the holes are filled in the result of method 2.

classes	Method 1	Method 2
Building	59.10	58.58
Vegetation	66.14	65.14
Road	45.88	46.61
Car	0.23	1.97
Clutter	28.45	31.61
Mean IoU	39.96	40.78

Table 4.3 IoU scores for method1 and method 2

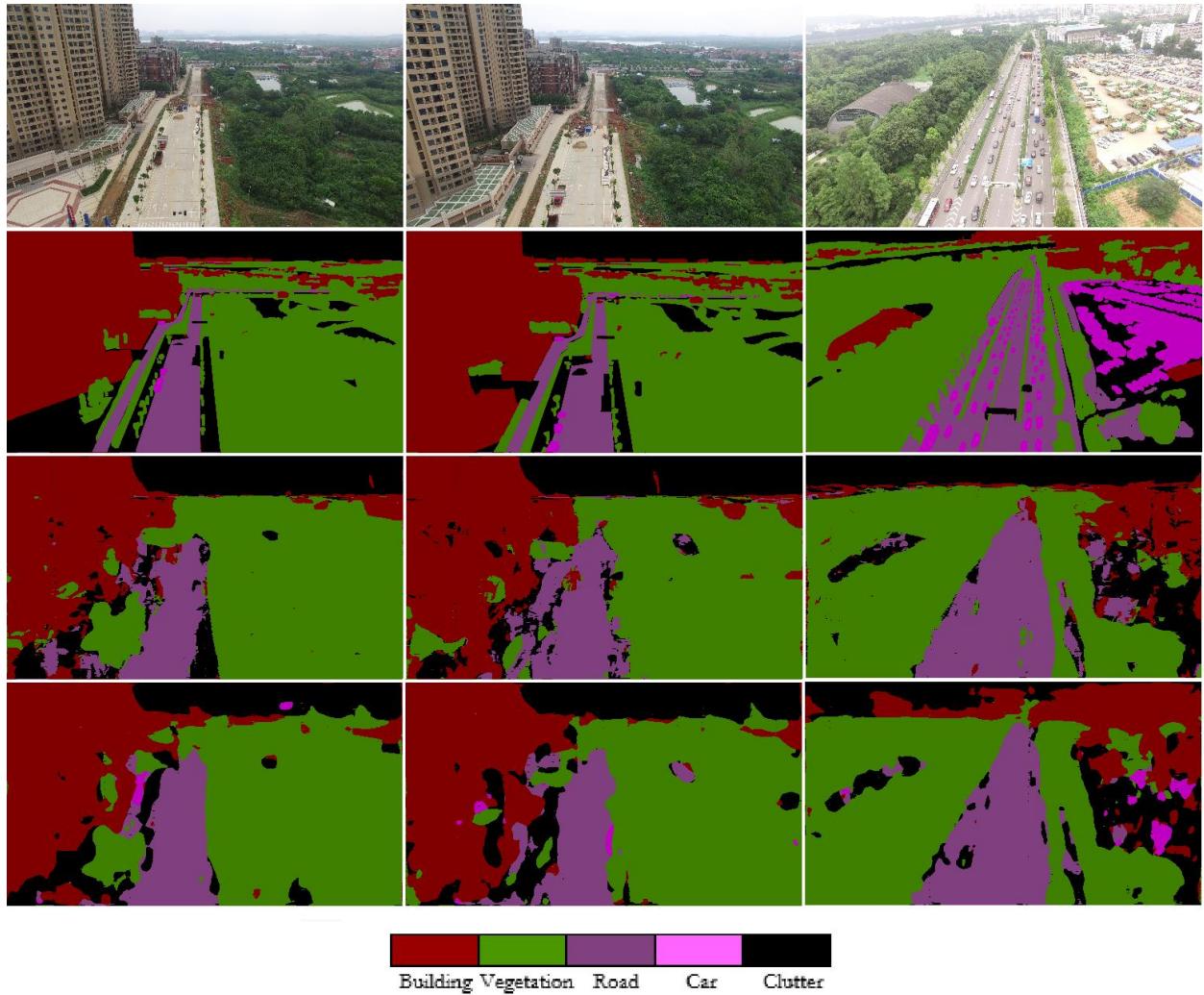


Figure 4.4 Comparison between method 1 and method2. The first row shows the original image extracted from the video. The second row shows the corresponding ground truth images. The Third row shows the segmentation results of the method 1. The fourth row shows the segmentation results of the method 2.

#### 4.7. Discussion

Previous three sections compare the FCN model, method 1 and method 2 separately. From these sections, we can know that method 1 and method 2 could improve the IoU of class road, car and clutter. The optimization ability of method 2 is better than method 1. The IoU of class building and class vegetation are decreased in method 1 and method 2 which may be caused by the noisy information from the previous frames. But in general, both method 1 and method 2 could improve the mean IoU of these 5 classes.

Table 4.4 shows the comparison of the IoUs of these three models. The Conv\_LSTM model could make use of the temporal information between the sequences, in other words, it could combine the information from the previous images and the current images to get the final prediction. The information from the previous images could help to improve the final prediction because the mistakes caused by the viewpoint of UAVs in some frames could be fixed. That may be the reason why the results of road, cars, and clutter could be improved. However, the information from the previous images also may bring some noisy information and influence the results. This may be the reason why the results of the building and vegetation decrease.

From the Table 4.4, it could be found that the increasing and decreasing degree of method 2 are both higher than method 1 which means method 2 is more sensitive to the influence of the information from the previous images. In other words, method 2 could make use of more temporal information than method 2. Also, the mean IoU of method 2 is higher than method 1. The reason for this result may be that the feature maps contain more information than the segmentation results so that could offer more information from the previous images.

In conclusion, both method 1 and method 2 could make use of the temporal information and improve the result of the segmentation which means the FCN + Conv\_LSTM model works. The improvements are mainly in those class that may be wrongly segmented because of the poor resolution and viewpoint of UAVs. Using feature maps as input could offer a better result than using the segmentation results as input both in the IoU and the smoothness of segmentation results.

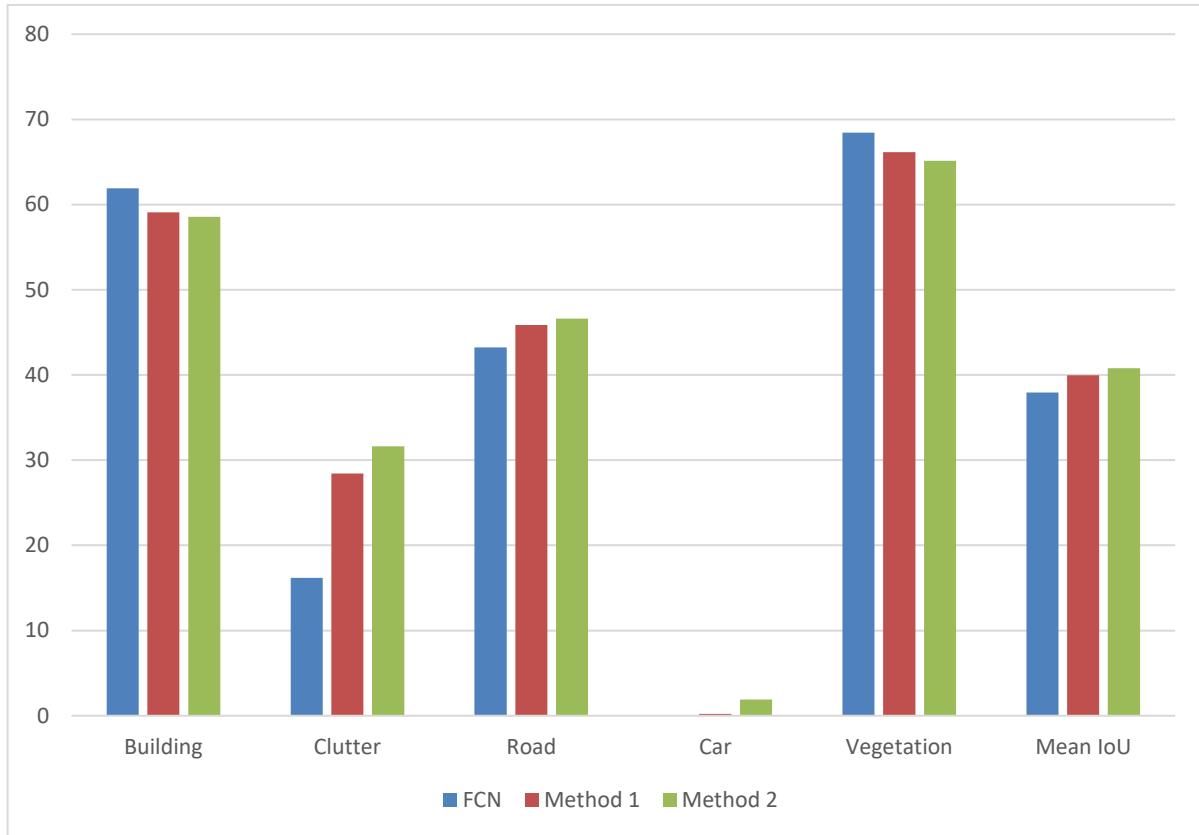


Table 4.4 Comparison between tree models

## 5. CONCLUSION AND FUTURE WORK

This paper explores the method for semantic video segmentation using temporal information and proposes the FCN+Conv-LSTM framework. The proposed method improves the segmentation result especially improves the result of the class road, clutter and car.

The proposed algorithm tries to combine the FCN model and the Conv\_LSTM model together. In this algorithm, the FCN model serves as the frame-based segmentation method which is used to segment each frame individually. The output of this part is the segmentation result of each frame or the feature map of each frame. The Conv\_LSTM model serves as the post-processing method which makes use of the temporal information between consecutive frames. The inputs of this part are sequences formed by the output segmentation results or the sequences of the feature maps extracted from FCN model. Conv\_LSTM learn the temporal information of these sequences and output the final segmentation results.

The experiments are done on the UAV videos captured in Wuhan, China from June to August 2017 and in Gronau, Germany in May 2017. 27 sequences are extracted from these videos and used to train the FCN model and the Conv\_LSTM model. The experimental results show the superiority of this FCN+Conv\_LSTM model especially in the class road, clutter and car compared to the single image segmentation model. Also, using feature maps as the input of Conv\_LSTM model could offer a better result than using the segmentation results as input. There are also some limitations to this method. The IoU of the class building and vegetation are decreased which may be caused by the influence of the noisy information introduced by the previous frames of the sequences.

For the whole research, there are some aspects that could be improved in the future. Firstly, because of the limitation of the computation ability, the original images are resized from  $3840 \times 2160$  pixels to  $512 \times 288$  pixels. The resolution of the images is decreased a lot and some details are lost. Secondly, the length of the blocks is only set as 4 due to the limitation of the GPU. Thirdly, the size of the dataset is not large enough because of the limitation of time for annotation.

For future work, we will extend this method to improve its accuracy and overcome the problems caused by the influence of the noisy information from the previous images.

### 5.1. Answers to research questions

Sub-objectives 1:

- How long is the interval between each frame extracted from the video?

Answer: The interval is 150 frames. During the data preparation phase of this research, time for labeling is limited. Therefore, the interval between each image is set as 150 to cover the whole videos.

- How many classes would be segmented?

Answer: 5 classes are segmented including building, road, vegetation, car and clutter.

- Which deep learning method would be chosen for the semantic segmentation of each frame, FCN, AlexNet or ResNet?

Answer: FCN8s is used in this research. FCN8s model is based on the VGG-16 network and adds a skip from pool3 at stride 8 to get more information from the global structure.

- How to extract the features of each frame and how to represent the features?

Answer: Using FCN8s to extract the feature map. The FCN8s contains 9 layers and each layer could extract different feature maps. These feature maps represent the features of the original images and are sent into the Conv\_LSTM network.

- How to measure the accuracy of the segmentation of each frame?

Answer: We calculate the IoU of different classes and the mean IoU of them.

Sub-objectives 2:

- How to use temporal information?

Answer: The Conv\_LSTM network is used to make use of the temporal information.

- How to connect the sequential frames to several blocks?

Answer: The original frames have 3 dimensions (width, height, band). We combine these consecutive frames together to form a 4 dimensions tensor (times, width, height, band). This 4-dimensions tensor is named as block.

- What is the size of each block?

Answer: Each block contains 4 frames. This number could be changed, while in this research we set it as 4 frames because of the limitation of computation ability.

- What is the size of the coverage between the sequential blocks?

Answer: The coverage between the sequential blocks is 3 frames. The network we used in this research uses the previous frames to predict the last frame. We set the coverage between the consecutive blocks as 3 to make sure that each frame (except the first three) could be predicted.

Sub-objectives 3:

- Which method would be used to connect the blocks?

Answer: The blocks have 4 dimensions (times, width, height, band). We combine these consecutive blocks together to form a 5 dimensions tensor (samples, times, width, height, band).

- How to evaluate the segmentation of the whole video?

Answer: We extracted test frames from the whole video. The test dataset is also formed as sequences and be predicted by the network. The predicted result of each sequence is the segmentation of the last frame. We calculate the IoU of these frames to measure the accuracy of them.

## 5.2. Recommendations

- Decreasing the interval between the consecutive frames could increase the temporal continuity between them.
- Increasing the size of the dataset could help to improve the result of our network.
- Other recurrent neural networks could be used to make use of temporal information such as the Conv\_GRU network.
- Increasing the length of the sequence block is a good direction to improve the performance of the Conv\_LSTM network. The longer the sequence the more temporal information the sequence contains.
- Combine the FCN and the Conv\_LSTM networks. In this research, these two networks are separate. The connection between them is based on output and input. The backpropagation processes are also separate. To improve the result, it is better to build a network contains two parts of layers. The first stack of layers works as the FCN model and the following layers work as the Conv\_LSTM model.



## LIST OF REFERENCES

---

- Badrinarayanan, V., Budvytis, I., & Cipolla, R. (2013). Semi-supervised video segmentation using tree structured graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11), 2751–2764. <https://doi.org/10.1109/TPAMI.2013.54>
- Badrinarayanan, V., Budvytis, I., & Cipolla, R. (2014). Mixture of trees probabilistic graphical model for video segmentation. *International Journal of Computer Vision*, 110(1), 14–29. <https://doi.org/10.1007/s11263-013-0673-5>
- Badrinarayanan, V., Galasso, F., & Cipolla, R. (2010). Label propagation in video sequences. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 3265–3272). IEEE. <https://doi.org/10.1109/CVPR.2010.5540054>
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481–2495. <https://doi.org/10.1109/TPAMI.2016.2644615>
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>
- Brutzer, S., Hoferlin, B., & Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1937–1944. <https://doi.org/10.1109/CVPR.2011.5995508>
- Byeon, W., Breuel, T. M., Raue, F., & Liwicki, M. (2015). Scene Labeling with LSTM Recurrent Neural Networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3547–3555). <https://doi.org/10.1109/CVPR.2015.7298977>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Garcia-Garcia, A., Orts-Escalano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70, 41–65. <https://doi.org/10.1016/j.asoc.2018.05.018>
- Horn, B. K. P., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1–3), 185–203. [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2)
- Jang, W.-D., & Kim, C.-S. (2017). Online Video Object Segmentation via Convolutional Trident Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5849–5858). <https://doi.org/10.1109/CVPR.2017.790>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems* (pp. 1097–1105). <https://doi.org/10.1101/j.protcy.2014.09.007>
- Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML* (pp. 282–289). <https://doi.org/10.1038/nprot.2006.61>
- Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. Retrieved from <http://arxiv.org/abs/1506.00019>
- Liu, B., & He, X. (2015). Multiclass semantic video segmentation with object-level active inference. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4286–4294). IEEE. <https://doi.org/10.1109/CVPR.2015.7299057>
- Long, J., Shelhamer, E., & Darrell, T. (2015a). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 07–12 June, pp. 3431–3440). <https://doi.org/10.1109/CVPR.2015.7298965>
- Long, J., Shelhamer, E., & Darrell, T. (2015b). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3431–3440). IEEE. <https://doi.org/10.1109/CVPR.2015.7298965>
- Lyu, Y., Vosselman, G., Xia, G., Yilmaz, A., & Yang, M. Y. (2018). The UAVid Dataset for Video Semantic Segmentation. *Arxiv.Org*, 2018, 1–9. Retrieved from <https://research.utwente.nl/en/publications/the-uavid-dataset-for-video-semantic-segmentation>
- Mahasseni, B., Todorovic, S., & Fern, A. (2017). Budget-Aware Deep Semantic Video Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2077–2086). IEEE. <https://doi.org/10.1109/CVPR.2017.224>
- Mingus, B., Herd, S., O'Reilly, R. C., Jilk, D. J., & Wyatte, D. (2016). Deep residual learning for image

- recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. <https://doi.org/10.3389/fpsyg.2013.00124>
- Mohammad, M. A., Kaloskampis, I., & Hicks, Y. (2015). New Method for Evaluation of Video Segmentation Quality. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (pp. 523–530). SCITEPRESS - Science and Technology Publications. <https://doi.org/10.5220/0005306205230530>
- Mustikovela, S. K., Yang, M. Y., & Rother, C. (2016). Can Ground Truth Label Propagation from Video help Semantic Segmentation? Retrieved from <http://arxiv.org/abs/1610.00731>
- Noh, H., Hong, S., & Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1520–1528). IEEE. <https://doi.org/10.1109/ICCV.2015.178>
- Pinheiro, P. O., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. JMLR.org. Retrieved from <https://dl.acm.org/citation.cfm?id=3044816>
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., & Woo, W. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 802–810. Retrieved from <http://arxiv.org/abs/1506.04214>
- Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S., & Babu, R. V. (2016). A Taxonomy of Deep Convolutional Neural Nets for Computer Vision. *Frontiers in Robotics and AI*, 2. <https://doi.org/10.3389/frobt.2015.00036>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 07–12–June, pp. 1–9). <https://doi.org/10.1109/CVPR.2015.7298594>
- Valavanis, K. P., & Vachtsevanos, G. J. (2015). UAV Applications: Introduction. In *Handbook of Unmanned Aerial Vehicles* (pp. 2639–2641). Dordrecht: Springer Netherlands. [https://doi.org/10.1007/978-90-481-9707-1\\_151](https://doi.org/10.1007/978-90-481-9707-1_151)
- Valipour, S., Siam, M., Jagersand, M., & Ray, N. (2017). Recurrent fully convolutional networks for video segmentation. In *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017* (pp. 29–36). <https://doi.org/10.1109/WACV.2017.11>
- Visin, F., Ciccone, M., Romero, A., Kastner, K., Cho, K., Bengio, Y., ... Courville, A. (2016). ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 426–433). <https://doi.org/10.1109/CVPRW.2016.60>
- Zhang, G., Yuan, Z., Liu, Y., Ma, L., & Zheng, N. (2015). *Video object segmentation by integrating trajectories from points and regions*. *Multimedia Tools and Applications* (Vol. 74). <https://doi.org/10.1007/s11042-014-2145-5>