This thesis is presented for the degree of Doctor of Philosophy of The University of Adelaide

Deep Learning for 2D and 3D Scene Understanding

Yu Liu
Jan 2020

School of Computer Science

Supervisors:
Prof. Ian Reid
Dr. Lingqiao Liu
Dr. Cesar Cadena (ETH Zurich)

Thesis Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works. I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signature:

Date: 1/10/2020

Abstract

This thesis comprises a body of work that investigates the use of deep learning for 2D and 3D scene understanding. Although there has been significant progress made in computer vision using deep learning, a lot of that progress has been relative to performance benchmarks, and for static images; it is common to find that good performance on one benchmark does not necessarily mean good generalization to the kind of viewing conditions that might be encountered by an autonomous robot or agent. In this thesis, we address a variety of problems motivated by the desire to see deep learning algorithms generalize better to robotic vision scenarios. Specifically, we span topics of multi-object detection, unsupervised domain adaptation for semantic segmentation, video object segmentation, and semantic scene completion.

First, most modern object detectors use a final post-processing step known as Non-maximum suppression (GreedyNMS). This suffers an inevitable trade-off between precision and recall in crowded scenes. To overcome this limitation, we propose a Pairwise-NMS to cure GreedyNMS. Specifically, a pairwise-relationship network that is based on deep learning is learned to predict if two overlapping proposal boxes contain two objects or zero/one object, which can handle multiple overlapping objects effectively.

A common issue in training deep neural networks is the need for large training sets. One approach to this is to use simulated image and video data, but this suffers from a domain gap wherein the performance on real-world data is poor relative to performance on the simulation data. We target a few approaches to addressing so-called domain adaptation for semantic segmentation: (1) Single and multi-exemplars are employed for each class in order to cluster the per-pixel features in the embedding space; (2) Class-balanced self-training strategy is utilized for generating pseudo labels in the target domain; (3) Moreover, a convolutional adaptor is adopted to enforce the features in the source domain and target domain are closed with each other.

Next, we tackle the video object segmentation by formulating it as a meta-learning problem, where the base learner aims to learn semantic scene understanding for general objects, and the meta learner quickly adapts the appearance of the target object with a few examples. Our proposed meta-learning method uses a closed-form optimizer, the so-called "ridge regression", which is conducive to fast and better training convergence.

One-shot video object segmentation (OSVOS) has the limitation to "overemphasize" the generic semantic object information while "diluting" the instance cues of the object(s), which largely block the whole training process. Through adding a common module, video loss, which we formulate with various forms of constraints (including weighted BCE loss, high-dimensional triplet loss, as well as a novel mixed instance-aware video loss), to train the parent network, the network is then better prepared for the online fine-tuning.

Next, we introduce a light-weight Dimensional Decomposition Residual network (DDR) for 3D dense prediction tasks. The novel factorized convolution layer is effective for reducing the network parameters, and the proposed multi-scale fusion mechanism for depth and color image can improve the completion and segmentation accuracy simultaneously.

Moreover, we propose PALNet, a novel hybrid network for Semantic Scene Completion(SSC) based on single depth. PALNet utilizes a two-stream network to extract both 2D and 3D features from multi-stages using fine-grained depth information to efficiently capture the context, as well as the geometric cues of the scene. Position Aware Loss (PA-Loss) considers Local Geometric Anisotropy to determine the importance of different positions within the scene. It is beneficial for recovering key details like the boundaries of objects and the corners of the scene.

Finally, we propose a 3D gated recurrent fusion network (GRFNet), which learns to adaptively select and fuse the relevant information from depth and RGB by making use of the gate and memory modules. Based on the single-stage fusion, we further propose a multi-stage fusion strategy, which could model the correlations among different stages within the network.

Acknowledgments

I am first and foremost very grateful to my principal supervisor, Prof. Ian Reid, for inspiring this work, and for giving me invaluable guidance throughout the time of my Ph.D., which made this work possible and has helped and inspired me to grow as a researcher. I greatly value the opportunity that was given to me to be a part of a great research team.

My sincere thanks also go to my co-supervisor, Dr. Lingqiao Liu, and Dr. Cesar Cadena, for the immense guidance and support, and the countless fruitful discussions leading to the ideas in this work.

I am also very thankful to Dr. Hamid Rezatofighi, Dr. Anton Milan, and Dr. Thanh-Toan Do, for the invaluable discussions, introduction into the world of multi-object tracking and instance segmentation, and great support towards this work.

I also like to thank my former co-supervisor, Prof. Javen Shi, for providing me with valuable insights and advice, Dr. Saroj Weerasekera, Dr. Yasir Latif for the great explanations, support, and advice.

I wish to thank all my lab mates, both former and present, for the stimulating discussions, the support, and the lasting memories. I especially like to thank Haokui Zhang, Jie Li, Anh-Dzung Doan, and Qingsen Yan for the great help and discussions towards this work. And also my friends Zhi Tian, Qiaoyang Luo, Wei Liu, Mingyu Guo, Violetta Schevchenko accompany and support me in the hard times.

I also like to acknowledge Prof. Roland Siegwart for the memorable opportunity to visit Autonomous Systems Lab and its research group at ETH Zurich. It is a wonderful journey that I live with some friends include Zetao Chen, Marius Fehr, Margarita Grinvald. I also like to thanks Xiaoying Shen and my best friend.

I am grateful for the support and memories to have given by the past and current members of the School of Computer Science at the University of Adelaide, the Australian Centre for Robotic Vision, and the Australian Institute for Machine Learning, and last but not least, to my family for their timeless support, and always being there to share the experiences from home and my Ph.D. journey.

Dedicated to —
— my mom.

Publications

This thesis contains the following work that has been published or prepared for publication:

- **Y. Liu***, J. Li*, Q. Yan, X. Yuan, C. Zhao, I. Reid, C. Cadena, "3D Gated Recurrent Fusion for Semantic Scene Completion", *arXiv:2002.07269*
- **Y. Liu***, J. Li*, X. Yuan, C. Zhao, R. Siegwart, I. Reid, C. Cadena, "Depth Based Semantic Scene Completion with Position Importance Aware Loss", *ICRA 2020*
- J. Li*, **Y. Liu***, D. Gong, Q. Shi, X. Yuan, C. Zhao and I. Reid, "RGBD Based Dimensional Decomposition Residual Network for 3D Semantic Scene Completion", *CVPR 2019*
- **Y. Liu**, L. Liu, H. Zhang, H. Rezatofighi, Q. Yan, I. Reid, "Meta Learning with Differentiable Closed-form Solver for Fast Video Object Segmentation", *IROS 2020*
- **Y. Liu**, L. Liu, H. Rezatofighi, T. Do, Q. Shi and I. Reid, "Learning Pairwise Relationship for Multi-object Detection in Crowded Scenes", *ArXiv: 1903.00620*
- **Y. Liu**, Y. Dai, A. Doan, L. Liu, I. Reid, "In Defense of OSVOS", *ArXiv: 1908.06692*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The goal of computer vision is to enable an intelligent system to see and understand the surrounding environment like human-beings. In computers, the eye is replaced by a similar optical device – a camera – that produces a projection of the 3D world onto a 2D image. Although 3D information is lost in this projection, it remains a rich information source and the goal of visual scene understanding is to extract that information that can lead to suitable knowledge and representations of the surrounding environment. Computer vision can also benefit from sensors that provide information that the eye does not, such as RGB-D sensors that deliver 2.5 D information – image+depth – directly, which is also called 3D vision. Both 2D and 3D vision are valid sensors for scene understanding. The former is mostly good for detection and recognition – what is where – at the object and pixel level. The latter (3D) gives 2.5D images – depth as well – which means that it is more amenable to geometric tasks (as well as semantic) such as scene completion.

The aim of this thesis is to address a number of fundamental problems in scene understanding, including object detection, scene segmentation, and scene completion. In turn advances in these areas begin to deliver the goal of scene understanding and consequently enable diverse robotic tasks such as driverless cars, augmented reality, and mobile robots.

Modern computer vision has made significant strides towards solving many of these problems above, not least because of breakthroughs associated with *deep learning* – the idea that a neural network with many layers can be configured to solve a visual task by training using a large dataset of input and ground-truth output values. By making use of the huge amount of the training data and the high-speed GPU computation, the performance of many vision tasks has been largely boosted by deep learning technology. However, there are some unsolved problems in deep learning. Firstly, current state-of-the-art methods are usually fully supervised, which means they need at least thousands of images (and sometimes millions) to be annotated for training the network. Meanwhile, extensive experiments prove that the deeper the network utilized, the better performance can be reached as well as much more training data are needed to avoid the over-fitting. Secondly, the trained network usually is task-specific, which means it has a poor generalization ability, therefore, how to quickly adapt to similar tasks is another problem to be solved.

However although there has been significant progress made in computer vision using deep learning, a lot of that progress has been relative to performance benchmarks, and for static images; it is common to find that good performance on one benchmark does not necessarily mean good generalization to the kind of viewing conditions that might be encountered by an autonomous robot or agent. In this thesis, we address a variety of problems motivated by the desire to see deep learning algorithms generalize better to robotic vision scenarios.

1.2 Approaches

One of the early successes of the deep learning revolution was object detection. Object detection is trying to find what is the object and where is it in the 2D image coordinate. There are now a number of high-quality object detectors, many of which can be used "off the shelf" so it might be assumed that this is a solved problem. However such standard solutions rarely work well in scenes with a lot of overlapping objects (eg pedestrians in a street scene), because almost all rely on a post-processing stage known as "Non-maximal suppression" which is essentially a non-deep "hack" to remove false positives. In crowded scenes, this hack performs poorly, since it can not reach a good balance between precision and recall. In order to address this problem, we propose a pairwise relationship network to determine if two overlapped proposals contain two different objects or zero/one object, which can handle multi-object detection under heavy occlusion effectively.

Another fundamental problem in scene understanding is semantic segmentation. Specifically, it aims to classify each pixel to a specific category in the image. As pointed out before, one necessary requirement for training a network is the need for a vast amount of training data. Acquiring training data is especially labor-consuming and time-consuming for segmentation since it is in the pixel-level dense annotation. Current state-of-the-art segmentation models are usually trained and evaluated on public benchmarks such as COCO Lin et al. [2014], PASCAL VOC Everingham et al. [2010], and they can achieve satisfactory performance on these two datasets correspondingly. In fact, the real-world scenarios vary a lot including the illumination, scene structure, pose, etc, and this creates a need for a network that has the strong ability to quickly adapt and get well segmentation for the unseen objects and scenarios. However, it is not practical to annotate new objects each time a new scenario is encountered. We observe, however, that among different datasets, although the appearance information varies from scene to scene, the low-level and the mid-level features can potentially be shared among different domains. This observation motivates approaches we propose for domain adaption, and for video object segmentation.

The first of these, domain adaptation, aims to address the issue of a dearth of training data via transfer learning. Specifically, domain adaptation aims to deliver algorithms which trained in one or multiple "source domain" to a related but different "target domain", and the source domain and target domain usually have different distribution but same feature space. For unsupervised domain adaptation (UDA), the setting is the labeled data are available in the source domain, and it aims to achieve good performance in the target domain which has no annotations available. Specifically, in UDA for segmentation, the source domain usually makes use of the synthetic data which can be generated easily from a game engine, and the target domain is usually a real-world dataset. One limitation of current UDA for segmentation methods is that they usually assume there is only one data distribution center for both the source and target domain and trying to enforce they are close with each other via adversarial learning. However, that may not be the case, so in this thesis, we propose a method that allows for a multi-modal distribution via a novel clustering approach.

For 2D scene understanding, video object segmentation is another interesting topic and fundamental problem. The goal of video object segmentation is to segment the rest frames of the same video given the annotation of the first frame. This potentially differs from static image segmentation because of the availability of temporal consistency between frames. Most of the state-of-the-art methods are based on a three-step online fine-tuning strategy like OSVOS Caelles et al. [2017], which is essentially over-fitting on the appearance of the target object(s). Compared with the traditional dense-tracking-based methods, it has the advantage to handle abrupt moving and heavy occlusion. Although good performance can be achieved, online fine-tuning-based methods are quite slow during inference, usually take around 8 to 10 seconds per image, which can not satisfy the purpose of practical use. To this end, we deliver a meta learning-based method that achieves fast video object segmentation with comparable accuracy.

The final topic considered in this thesis is a 3D scene understanding area called Semantic Scene Completion (SSC). Scene completion is the process by which a scene that has missing geometry is filled in (eg an RGB-D camera will deliver depths at most pixels, but some areas will be unknown because the sensor failed at that location because of poor reflection). Semantic scene completion aims to solve both geometric and semantic completion together, and our insight is that the two processes can aid one another. Previous methods are solely based on the depth as input, which can not fully utilize the semantic information. Based on our observation, semantic segmentation mainly benefits from the RGB image since it contains much more information about color and texture, and scene completion mainly benefits from depth since it contains much more information about geometry and shape. Therefore, rather than using depth information solely, we instead of making use of both RGB and depth as input for the network, and fuse the two-modal of features in different stages. Moreover, the state-of-the-art methods use a standard cross-entropy loss which treats each voxel equally, however, in a 3D scenario, there are a huge amount of voxels within the object which carry redundant information, and relatively few voxels which located in the edges, corners of scene play the key role for the scene segmentation and completion. We propose a position-aware importance loss to boost the task performance and speed up the training convergence. Finally, because simple fusion (such as summation, average) of the two modes of data, RGB and depth, fails to achieve satisfactory performance, in this thesis we propose the use of a Gated Recurrent Unit as a means to perform a better fusion of RGB features + Depth features.

1.3 Summary of Contributions

This thesis presents the following main contributions:

C1 Non-Maximum Suppression: In chapter 3, An end-to-end Pairwise-NMS relationship network is proposed for replacing the GreedyNMS, which achieves better performance for object detection without losing efficiency. We evaluate the proposed method on three public datasets, both qualitative and quantitative results demonstrate that our method performs better than GreedyNMS, especially in the crowded scene. The proposed method also achieves better performance than the recent Soft-NMS Bodla et al. [2017]. Thanks to its flexibility, Pairwise-NMS can be integrated into learning-based detectors (e.g., Faster-RCNN Ren et al. [2017] and DPM Felzenszwalb et al. [2010]) neatly, and pave the way for the end-to-end learning detectors.

C2 Domain adaptation for semantic segmentation. In chapter 4, we propose a novel UDA method for semantic segmentation with three contributions: (1) Using the multi-exemplars to replace the commonly used FC classifier for better representing the data distribution of each class, and combined with the proposed clustering loss to make the exemplars which belong to different classes are separable. (2) Taking use of the class-balanced self-training strategy, which will generate pseudo labels in the target domain, and incorporate the appearance information into the training loop. Iteratively optimize the segmentation model and the generated pseudo labels. (3) With the help of the designed convolutional adaptors, features from different domains can be better aligned. Therefore, achieve better segmentation performance in real data.

C3 Video object segmentation: In chapter 5, we deliver a meta-learning-based method for video object segmentation, using a closed form solver (ridge regression) as the internal optimizer. This is capable of performing fast gradient back-propagation and can adapt to previously unseen objects quickly with very few samples. Inference (i.e. segmentation of the video) is a single forward pass per frame with no need for fine-tuning or post-processing. Ridge regression in high-dimensional feature spaces can be very slow, because of the need to invert a large matrix. We address this by using a novel block splitting mechanism, which greatly accelerates the training process without damaging the performance. We demonstrate the state-of-the-art video segmentation accuracy relative to all others methods of comparable processing time, and even better accuracy than many slower ones.

C4 Video object segmentation: In chapter 6, we propose video loss for video object segmentation which can speed up the training process of current online fine-tuning-based methods. In the neural network of video object segmentation, the low-level layers have relatively large spatial resolutions, and carry more details about the object instance, while the high-level layers have stronger abstract and generalization ability, leading to carry more category information. Especially, in the second phase of OSVOS Caelles et al. [2017], i.e. training the *parent network*, it actually tries to fine-tune the network to acquire the ability to distinguish the objects from the background. However, it dilutes the 'instance' information. And quickly adapts to the specific (target) instance, which is exactly the need of the third phase (i.e. *online finetuning*). Video loss can effectively 'rectify' the training process of *parent network*, and make it be better prepared for the *online fine-tuning*.

Specifically, each video is supposed to maintain an *average object*, and through mapping, we expect that the objects from the same video are close to each other in the embedding space, while the objects from different videos are far away from each other. In this way, video loss can help the network to maintain an *average object* for each video sequence.

C5 Semantic scene completion: In chapter 7, we propose DDR-Net for semantic scene completion. Firstly, we propose the dimensional decomposition residual (DDR) blocks for 3D convolution, which dramatically reduces the model parameters without performance degradation. Secondly, 3D feature maps of RGB and depth are fused in multi-scale seamlessly, which enhances the network representation ability and boosts the performance of SC and SSC tasks. Thirdly, the proposed end-to-end training network achieves state-of-the-art performance on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets.

C6 Semantic scene completion: In chapter 8, we propose PALNet for semantic scene completion. Firstly, we propose a novel PA-Loss for the SSC task, which emphasizing the rare voxels that are located on the surface or corners of the scene, while diluting the voxels which carry redundant information within the objects. Our experiments indicate that PA-Loss has the benefit of slightly faster convergence for training and can achieve better performance than previous works. Secondly, based on single-view depth, we propose a hybrid network, which takes full advantage of both fine-grained depth and TSDF. The detailed information extracted from the full resolution depth is beneficial for semantic labeling as well as scene completion, which also distinguishes our approach from the existing mainstream approaches that only use the voxelized TSDF as the sole input.

C7 Semantic scene completion: In chapter 9, an end-to-end 3D-GRF based network, GRFNet, is presented for fusing RGB and depth information in the SSC task, through employing *gate* and *memory* components, the selection, and fusion between two modalities can be conducted effectively. To the best of our knowledge, this is the first time that a gated recurrent network is employed for data fusion in the SSC task. Within the framework of GRFNet, single-stage and multi-stage fusion strategies are proposed. While outperforming existing fusing strategies in the SSC task already with the single-stage, the multi-stage fusion proves to give the best results. Extensive experiments demonstrate that the proposed GRFNet achieves superior performance on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets.

The developed methods cover the aspect of 2D and 3D scene understanding and will be useful in diverse automated tasks, including auto-driving, mobile robots, and augmented reality.

Bibliography

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS– improving object detection with one line of code. In *ICCV*, 2017.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2241–2248. IEEE, 2010.

M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, pages 5431–5440, 2016.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, June 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2577031.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

Chapter 2

Literature Review

This chapter aims to provide a more detailed review on 2D and 3D scene understanding with the focus on topics on relevance to the work in chapters 3-9. Specifically, we discuss the problems of multi-object detection, semantic segmentation, video object segmentation and semantic scene completion. We also provide background material on topics of relevance to these problem areas, including non-maximum suppression, unsupervised domain adaptation, and meta learning.

2.1 Multi-object Detection

As a fundamental research topic in computer vision, object detection aims to classify and locate the objects in the 2D image coordinate, multi-object detection is possibly more than just running a detector everywhere, and multi-object detection can be viewed as the cornerstone for instance-level segmentation Arnab and Torr [2017]; He et al. [2017a]; Urtasun [2017] and multi-object tracking Leibe et al. [2007]; Tang et al. [2013]; Zamir et al. [2012]. In multi-object detection, several objects are usually overlapping with each other as a cluster, which is called co-occurrence Ouyang and Wang [2013]; Rodriguez et al. [2011]; Sadeghi and Farhadi [2011]. The problems that arise with co-occurrence are challenging such as occlusion, illumination changing, texture inconsistency, etc. Among these problems, occlusion is the most notorious issue. Based on whether the clusters of objects belong to the same category or not, the co-occurrence can further be classified into intra-class co-occurrence (also called crowded scene) and inter-class co-occurrence. Because different object categories usually contain different discriminative features, the problem of inter-class co-occurrence is relatively easier to solve than the one of intra-class co-occurrence. In particular, due to the incredible contribution of deep neural network and huge training data Ref; He et al. [2016a]; Krizhevsky et al. [2012a]; Russakovsky et al. [2015], significant improvements have been achieved in object detection Girshick [2015]; Girshick et al. [2012]; He et al. [2014]; Lin et al. [2017b]; Ren et al. [2015] based on the extremely accurate per-category classification in recent years, thus inter-class co-occurrence is not a big deal anymore for object detection. However, intra-class co-occurrence still remains as a challenging problem, and most of the popular detectors fail when they come to multiple overlapping objects in crowded scenes.

In order to understand the essence of the problem brought by intra-class co-concurrence in multi-object detection, it is necessary to know the work principle and evaluation criteria of object detection. With the developments for object detection over many years, in general, a common pipeline for

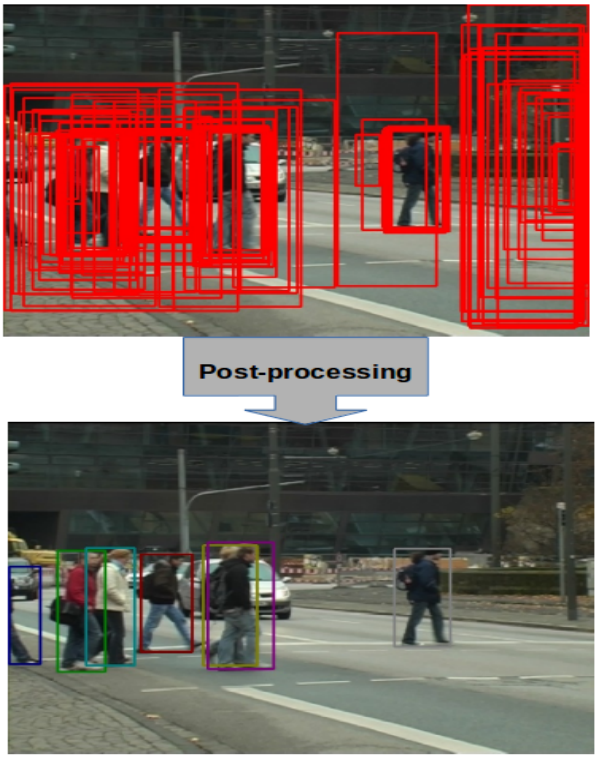


Figure 2.1: Detection proposals to final bounding boxes: for object detection, on the top, (100) detection proposals (generated by Faster-RCNN) detector, will pass through a post-processing step (GreedyNMS is used by default), and acquire the final bounding boxes as shown at the bottom.

object detection includes three stages: proposals generation, object detection, and post-processing. A commonly used post-processing step for object detection is non-maximum suppression, which is also called GreedyNMS Felzenszwal et al. [2010]; Girshick [2015]; Girshick et al. [2012]; Ren et al. [2015]. As the name suggests, non-maximum suppression tries to suppress above-threshold values that are not local maxima based on the assumption that non-maximal values usually arise as false positives associated with the true maximum. Specifically, for NMS in object detection, all the proposals are sorted according their confidence score before post processing, and in each step, non-maximum suppression always pick the proposal with the highest score and suppress proposals which have overlap of more than some threshold and a detection score that is lower than the local maximum.

Meanwhile, for evaluation, Mean Average Precision (MAP) is used as the standard criterion Everingham et al. [2015]; Lin et al. [2014]. The reason that detectors fail in crowded scenes is that several objects are often close by each other and heavily occluded. The post-processing step with GreedyNMS is not suitable in such scenarios as it is only capable of achieving good performance for isolated objects in sparse scenes. Specifically, the de facto used GreedyNMS usually suppresses harshly, leading to many missed detections in crowded scenes. Despite this limitation, GreedyNMS is efficient and achieves good performance on sparse scenes, so it has long been the default post-processing step in object detection. In the following section, we list out several methods for the post-processing step of object detection.

Non-maximum Suppression From the mathematical perspective, non-maximum suppression can be viewed as a quadratic problem. In object detection, the inverse scores of the proposals are treated as the unary terms and IoUs between the proposals are treated as the binary terms Rezatofighi et al. [2017]. The scores are generated by detector based on the encoding of localizations between proposals and ground-truth bounding boxes. The role of GreedyNMS is to assign zero if the IoU between two proposals is smaller than a NMS threshold or infinite vice versa. By minimizing an energy function, the problem can be solved through a quadratic optimization solver. As pointed out by previous works Barinova et al. [2012]; Chan et al. [2008]; Pham et al. [2016]; Pirsiavash et al. [2011], instead of using GreedyNMS, which greedily iterates and acquires local minimum, using other solvers can provide better local minima. While theoretically appealing, there is an increase in the complexity in both memory and time by using quadratic optimization solvers, especially when the number of proposals is large Barinova et al. [2012]; Chan et al. [2008]; Pham et al. [2016]; Pirsiavash et al. [2011]. GreedyNMS can not reach a good balance between precision and recall for multi-object detection under heavy occlusion.

Message Passing. Targeting at object detection in crowded scenes, Rasmus *et al.* Rothe et al. [2014] propose to use a latent structured SVM to learn the weights of affinity propagation clustering between proposals. By passing messages between windows, the merging step can be executed by using clustering methods, and proposals can be re-weighted through minimizing energy function. This kind of method can handle cases which contain two objects that are very close to each other. However, it is hard to distinguish the boundary between one object and two objects.

Soft-NMS. Both sub-optimal solver and message passing methods belong to the branch of improving the binary term, *i.e.* re-weight. There are several works that focus on improving the unary term, that is re-score, of the proposals. One of the interesting works is Soft-NMS Bodla et al. [2017]. Comparing to GreedyNMS, Soft-NMS decreases the detection scores as an increasing function of IoUs between the target proposals and the overlapped ones, rather than setting scores of the overlapped proposals to zero. Although just one line of code is added, some gains in performance among several detection datasets were acquired. To be specific, a linear or gaussian function is employed to re-score the non-maximal proposals rather than to totally suppress them. The limitation of this method is that it has to manually tailor a corresponding function to alleviate GreedyNMS. This is troublesome since the balance between recall and precision is usually trade-off and data driven. Furthermore, as showed in the experiment section, it can not handle the heavily occluded cases very well.

ConvNet NMS. Very recently, Jan Hoang *et al.* Hosang et al. [2016, 2017] propose to use a neural network to re-score the proposals. Being different from other works, they argue that NMS should be totally replaced with the convolutional neural network. The approach in Hosang et al. [2016, 2017] can be used to learn an end-to-end solution for object detection. However, we note that for the for sparse scenes, GreedyNMS performs well and efficient. Hence GreedyNMS should not be totally removed from an object detection pipeline. In addition, Learning-NMS Hosang et al. [2017] requires a large memory cost, so it limits the number of the sampling proposals for training the network.

Pairwise Prediction. Based on the same application scenario as Rothe et al. [2014], Tang *et al.* Tang et al. [2013] use a SVM to handle the object pairs in the crowded scene, but with manually designed features. It is hard to handle the case when two nearby objects with the similar appearance.

Compared to previous works, our work also focuses on predicting object pairs like Tang et al. [2013],

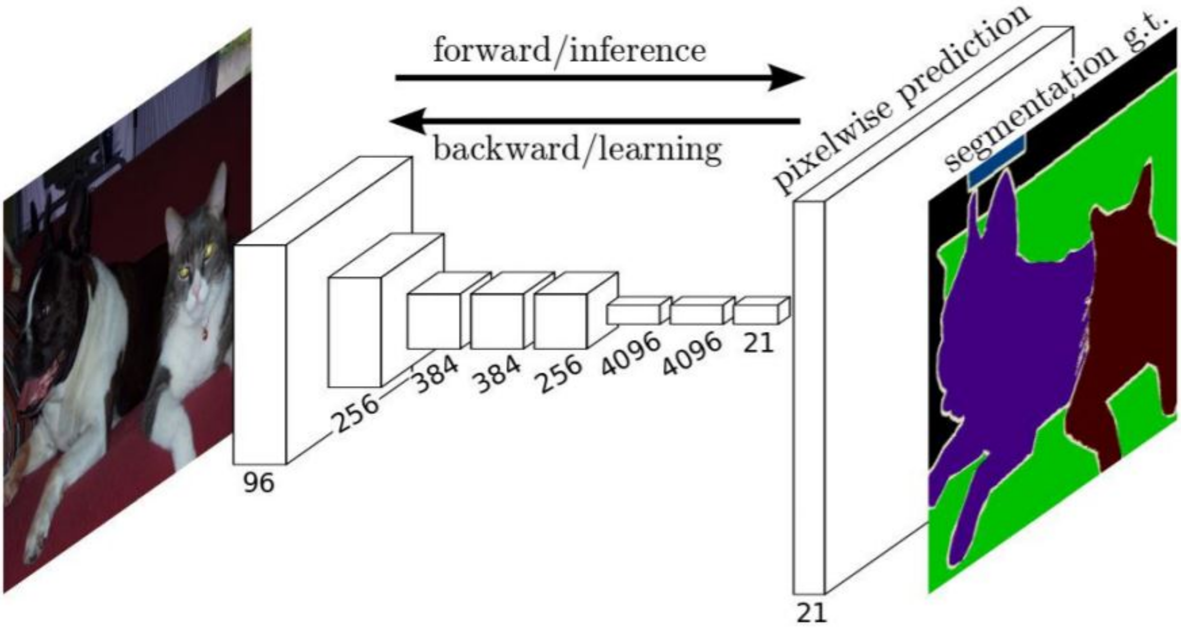


Figure 2.2: Semantic segmentation is a per-pixel dense prediction task. The figure comes from Long et al. [2015] by using a fully convolutional network.

but we employ a deep learning based network. In addition, rather than throwing away NMS framework totally like Learning-NMS Hosang et al. [2017], the good merits of NMS are retained while the hard constraints are mitigated. Specifically, we utilize the features of the region of interest (ROI) to pass messages between pairs. After the pairwise relationship matrix of the detections within one image was learned, only two lines of codes are added to improve the GeedyNMS. The proposed method is also simple as Soft-NMS Bodla et al. [2017], but it provides a more robust solution for multi-object detection under heavy occlusions.

2.2 Semantic Segmentation

The performance of fully supervised semantic segmentation has long been boosted up since the invention of the work by long et al Long et al. [2015], which include some well-known works such as RefineNet Lin et al. [2017a], DeepLab Chen et al. [2017a] and their derivatives. The overall trend is using more labeled data and much heavy networks to push the extreme performance to a limit. However there is significant cost involved in this exercise, and in some cases it is not even possible. This has therefore sparked a range of work in semantic segmentation aimed at reducing the amount of hand-labelled training data required.

2.2.1 Unsupervised domain adaptation

Unsupervised domain adaptation (UDA) is somehow a new problem which has attracted increasing attention recently. UDA is a learning framework to transfer knowledge learned from source domains with a large number of annotated training examples to target domains with unlabeled data only. This is important for applications like driverless cars because of the difficulty of acquiring sufficient relevant training data for all possible scenarios the car would encounter. Since in real scenarios, the ubiquitous environment changes due to the illumination, pose, sensor various. It is impossible to develop an unified network which is capable of handling all of the new scenarios, therefore, how to take use of the avaliable annotated labels to quickly adapt to the new environment become a hot topic in the computer vision community. Domain gap refers that models trained in one domain generally have poor performance on other domains. Therefore, it is very important to endow the model with the ability to adapt quickly (eg using meta-learning) in new scenarios.

Style transfer learning methods assume the semantic/structural context/meaning of two domains are similar/same, and try to transfer the feature of appearance from the source to the target. The representatives are CycleGAN Zhu et al. [2017] and UNIT Liu et al. [2017]. Although it can incorporate the appearance information of the target domain, one limitation that style transfer based methods suffer is the "unreal" effects of transferred style, which make these kind of methods still have gaps from the real data. Through using discriminator and generator, adversarial learning enforce the features from two domains are closed with each other, which can be done in pixel/feature/prediction Busto et al. [2018]; Chen et al. [2017b]; French et al. [2017]; Grandvalet and Bengio [2005] levels. Specifically, by taking use of circle consistency loss, CycleGAN Zhu et al. [2017] tries to find the good matches between similar features. BDL Li et al. [2019b] employs adversarial loss and reconstruction loss to conduct domain-transfer based semantic segmentation. However, BDL Li et al. [2019b] suffers the problem that the network structure and loss term are too complex and is not elegant at all.

2.2.2 Semi-supervised Learning Based Methods

Pseudo labelling refers to a process whereby a trained model (trained fully supervised with labeled data) is used to predict labels in unlabeled data, which are then used to augment the training set. Self-training strategy seeks to generate pseudo labels in the target domain based on the fully-supervised model trained in the source domain, which can effectively boost the performance of the UDA task. One limitation of the previous self-training based methods is that they are prone be overwhelmed by the easy classified pixels which have a relatively high confidence score during the process of generating pseudo labels, while ignoring the hard-to-classify classes. CBST Zou et al. [2018] proposes to use a class-balanced self-training strategy to generate pseudo labels in the target domain based on the self-training strategy, which can effectively boost the performance of the UDA task, since it can generate some confident labels in the target domain, which is helpful for the UDA task to capture the appearance information from the target domain. However, self-training can essentially be viewed as a joint learning problem to optimize the model as well as the pseudo labels in the target domain, so it cannot ensure the generated pseudo labels are accurate as expected. That means, as the training process is underway, more and more generated pseudo labels will be added into the training annotations, the remaining labels will have much less confidence regarding to the prediction accuracy.

2.2.3 Multi-domain Learning Based Methods

Multi-domain learning (MDL) aims at obtaining a model with minimal average risk across multiple domains. Inspired by the work of Rebuffi et al. [2017], we apply the multi-domain learning strategy

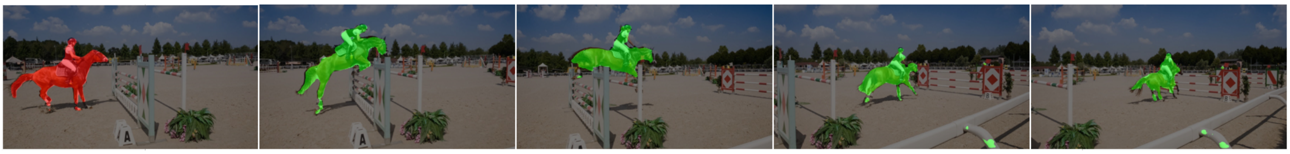


Figure 2.3: Video object segmentation: The per-pixel dense annotation of the first frame (red) is given, the goal for video object segmentation is to acquire the segmentation mask in the rest of the frames independently (green).

in the UDA task. Being different from the transfer learning and multi-task learning, multi-domain learning aims to achieve feature sharing among domains, with the goal to quickly adapt to the new task while still retain the good performance in the old task. Specifically, assume the learning function from input X, can be represented by $F(X) = \alpha * F * x$, where F store most of the mid-level and low-level features learned from different domains, which are class-agnostic. α is the class-determined weight, which only take a small ratio of the overall parameters, while can quickly adapt to the target task. In Chapter 4 we follow the philosophy of residual adaptor which is delivered by Rebuffi et al. [2017], that are serial adaptor and parallel adapter, we adapt these two kind of convolutional adapters based on VGG Simonyan and Zisserman [2014b] backbone in our task to better align the features from source domain to the target domain.

2.3 Video Object Segmentation

The goal of video object segmentation is to 'cutout' the target object(s) from the entire input video sequence. For semi-supervised video object segmentation, the annotated mask of the first frame is given, and the algorithm is designed to predict the masks of the rest frames in the video. There are three categories in this spectrum.

Tracking based Methods In this category, one stream of methods employ the optical flow to track the mask from the previous frame to the current frame, including MSK Perazzi et al. [2017], MPNVOS Sun et al. [2018] etc, one limitation of those methods is that they can not handle heavy occlusion and fast moving. Most recently, there are an emergence of methods which use the ReID technique to conduct the video object segmentation, including PReMVOS Luiten et al. [2018] and FAVOS Cheng et al. [2018]. Specifically, FAVOS using ReID to tackle the part-based detection box first, and through merging the (box) region based segments to form the final segmentation. PReMVOS firstly generate instance segmentation proposals in each frame, and then take use of the ReID technique to do data association to pick the correct segments in temporal domain, which can largely reduce the background noises brought by other nearby or overlapped object(s).

Adaptation based Methods For this category of methods, the core idea is utilizing the mask priors acquired from the previous frame(s) to be the guidance, to supervise the prediction in the current frame. Specifically, Segflow Cheng et al. [2017a] takes use of a parallel two-branch network to predict the segmentation as well as optical flow, through the bidirectional propagation between two frames, calculating optical flow and segmentation together and expecting them to benefit from each other. RGMP Wug Oh et al. [2018] takes both annotations of the first frame and predicted mask of the previous frame as guidance, employs a siamese encoder-decoder to conduct the mask propagation as well as detection, and with synthetic data to further boost the segmentation performance. OSMN Yang et al. [2018] shares the similar design principle with RGMP, while the difference is that it uses an modulator to quickly adapt the first annotation to the previous frame, which can then be used by the segmentation network as the spatial prior.

Fine-tuning based Methods Besides the aforementioned two categories of methods, there are some fine-tuning based methods which achieve the top performance in video object segmentation benchmark are OSVOS-S Maninis et al. [2017], OnVOS Voigtlaender and Leibe [2017], CINM Bao et al. [2018] etc, and all of them are derived from OSVOS Caelles et al. [2017]. Specifically, OSVOS-S Maninis et al. [2017] aims to solve the problem of removing noisy object(s) with the help of instance segmentation. While OnAVOS Voigtlaender and Leibe [2017] tries to enhance the network's ability for recognizing the new appearance of the target object(s) as well as suppressing the similar appearance carried by the noisy object(s). CINM Bao et al. [2018] is also initialized with the fine-tuning model, and employ a CNN to infer the markov random field (MRF) in spatial domain, and with optical flow to track the segmented object(s) in temporal domain.

2.3.1 Fast Video Object Segmentation

A few previous methods proposed to tackle fast video object segmentation. In particular, FAVOS Cheng et al. [2018] first tracks the part-based detection. Then, based on the tracked box, it generates the part-based segments and merges those parts according to a similarity score to form the final segmentation results. The limitation of FAVOS is that it can not be learned in an end-to-end manner, and heavily relies on the part-based detection performance. OSNM Yang et al. [2018] proposes a model which is composed of a modulator and a segmentation network. Through encoding the mask prior, the modulator can help the segmentation network quickly adapt to the target object. RGMP Wug Oh et al. [2018] shares the same spirit with OSNM. Specifically, it employs a siamese encoder-decoder structure to utilize the mask propagation, and further boosts the performance with synthetic data. The most similar work to ours is PML Chen et al. [2018b], which formulates the problem as a pixel-wise metric learning problem. Through the FCN Long et al. [2015], it maps the pixels to high-dimensional space, and utilizes a revised triplet loss to encourage pixels belonging to the same object much closer than those belonging to different objects. Nearest neighbor (NN) is required for retrieval during inference. In contrast our meta-learning approach acquires a mapping matrix between the high-dimensional feature and annotated mask in reference image using ridge regression, and then can be adapted rapidly to generate the prediction mask. Compared to baseline method PML Chen et al. [2018b], the method we propose in Chapter 5 is twice faster and achieves a 3.8% gains in segmentation accuracy. And with the same efficiency, the J *mean* of our method is 3.4% better than OSNM Yang et al. [2018] on the DAVIS2016 Perazzi et al. [2016] validation set.

2.3.2 Meta Learning

Meta learning can be used as an effective way to handle the problem of unsupervised domain adaptation for semantic segmentation. Meta learning is also named learning to learn Naik and Mammone [1992]; Schmidhuber [1987], it is an alternative to the de-facto solution that has emerged in deep learning of pre-training a network using a large, generic dataset (eg ImageNet Deng et al. [2009]) followed by fine-tuning with a problem-specific dataset. Meta-learning aims to replace the fine-tuning stage (which can still be very expensive) by training a network that has a degree of plasticity so that it can adapt rapidly to new tasks. For this reason it has become a very active area recently, especially with regard to one-shot and few-shot learning problems Fei-Fei et al. [2006]; Lake et al. [2015].

Recent approaches for meta-learning can be roughly put into three categories: (i) metric learning for acquiring similarities; (ii) learning optimizers for gaining update rules; and (iii) recurrent networks for reserving the memory. In Chapter 6, we adopt the meta-learning algorithm that belongs to the category of learning optimizers. Specifically, inspired by Bertinetto et al. [2018] which was originally designed for image classification, we adopt ridge regression, which is a closed-form solution to the optimization problem (explained in more detail in Ch6). The reason for using it is because, compared with the widely-used SGD LeCun et al. [1998] in CNNs, ridge regression can propagate gradient efficiently, which is matched with the goal of *fast mapping*. Through extensive experiments, we demonstrate that the proposed method is in the first echelon regarding to speed for fast video object segmentation, while obtaining more accurate results without any post-processing.

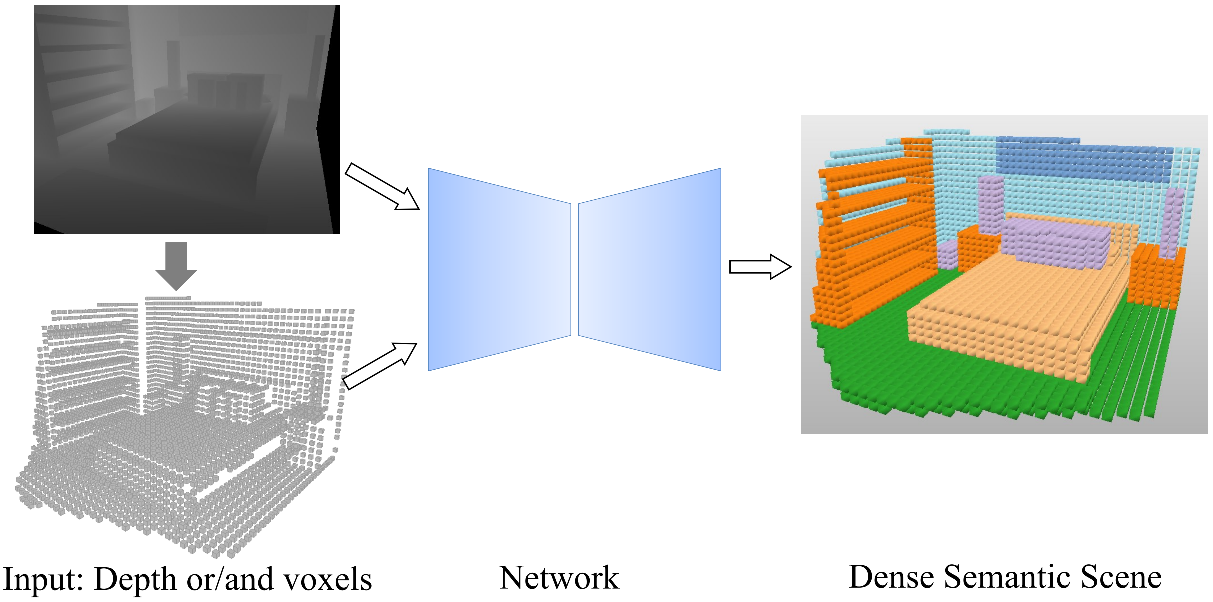


Figure 2.4: Semantic Scene Completion: Based on input depth or its corresponding 3D voxel volume, the goal of SSC is to simultaneously complete the partial 3D shapes and predict the dense semantic labels of both observed and unobserved parts in the view frustum.

2.4 Semantic Scene Completion

Semantic Scene Completion (SSC) refers to the task of inferring the 3D semantic segmentation of a scene while simultaneously completing the 3D shapes. Recently several methods have been proposed for SSC using deep learning techniques Garbade et al. [2018]; Guo and Tong [2018]; Song et al. [2017]; Zhang et al. [2018a]. Among them, the most representative work is the SSCNet Song et al. [2017] which conducts the semantic labeling and scene completion simultaneously and also proves that these two tasks can benefit from each other. SSCNet takes advantage of TSDF and uses voxels to represent the 3D space. Although better results have been achieved compared with the previous methods, SSCNet ignores the fine-grained information of depth. Zhang *et al.* Zhang et al. [2018a] introduces spatial group convolution (SGC) to reduce the computation costs but with poor performance than SSCNet Song et al. [2017]. SEGCloud Tchapmi et al. [2017] employs fine-grained 3D point as input but the computing and memory costs are incredibly high. VVNet Guo and Tong [2018] combines 2D-CNN and 3D-CNN by replacing some 3D volume layers with the corresponding 2D view network layers which leads to a much accurate and efficient network compared with SSCNet. However, it discards TSDF, which can provide the prior knowledge about the space encoding and is vital to distinguish between the empty and occupied parts of the scene. Two representives of generative models for SSC are 3D-GAN Wu et al. [2016] and ASSC Wang et al. [2018]. 3D-GAN Wu et al. [2016] uses volumetric convolutional networks to generate 3D objects from a probabilistic space. ASSC Wang et al. [2018] applies auto-encoder to encode the latent context of the single-view depth and uses a 3D generator to rebuild the 3D complete scene. To enrich the input information and boost the accuracy of SSC, TS3D Garbade et al. [2018] proposed to add a RGB branch in addition to the voxel branch, which introduce extra network or parameters, and are less accurate than our method.

Different from the previous methods, our proposed *Position Aware Loss Network (PALNet)* (described in Chapter 8) takes advantages of both the fine grained depth and the TSDF encoded 3D volume.

2.4.1 Loss Function for 3D Dense Prediction

Compared to 2D image segmentation, 3D dense prediction has three characteristics in general. Firstly, the number of voxels in 3D dense prediction is much larger than the amount of pixels in the 2D image segmentation. Secondly, the number of voxels ranges a lot among objects with different sizes. Finally, the number of voxels outside the objects is far beyond that of inside the objects. And we further observed that in 3D semantic scene completion, voxels at different positions make different contributions as well as deliver various training difficulties to the scene understanding. Based on the above observations, it is thus vital to choose a suitable loss function to train the 3D network effectively with the consideration of voxel-wise data-balance. There are many classic loss functions available to train 3D networks.

Cross-entropy Loss Extended from 2D vision tasks, cross-entropy loss can be used in 3D dense prediction. In essence, it treats all the predicted targets equally and is proved less efficient in 3D tasks Milletari et al. [2016]; Song et al. [2017].

Weighted Cross-entropy Loss Weighting factor $w_c \in [0, 1]$ is introduced based on cross-entropy loss to handle the imbalance problem. Weighted cross-entropy loss can emphasize the importance of classes with rare samples, while it relies on manually set weight parameters. A compromise approach instead of manual selection of weights is to set w_c as the inverse frequency for the corresponding class. And it can only handle category-level data imbalance, but not voxel-wise imbalance.

Focal Loss Focal loss aims to address the data imbalance in object detection especially when the dataset contains too many easy negatives that contribute no meaningful learning signals. However, one limitation of using focal loss is that it underestimates the importance of well classified samples Nguyen et al. [2018]; Redmon and Farhadi [2018]. Also, the training process becomes sensitive to incorrectly labeled samples Guo et al. [2018].

Dice Loss Dice loss Milletari et al. [2016] is proposed to address the data imbalance in volumetric medical image segmentation. It is a good choice to address the imbalance between the foreground and background in binary segmentation, but does not generalize well to multi-category segmentation. Besides, it is not as easy to optimize as cross-entropy loss, as its gradient may blow up to some enormous value when both the value of the target label and the prediction are small.

In summary, none of those loss functions can take into account the importance of different positions. In this thesis, we propose Local Geometric Anisotropy (LGA) to determine the importance of geometric information contained in different voxels, and LGA is then used to calculate the weight factor for cross-entropy loss to form the proposed PA-Loss, which fully considers the impact of each element, leads to the better performance compared to other loss functions.

2.4.2 Computation-efficient Networks

As a milestone in deep learning architectures, ResNet He et al. [2016b] uses a residual block to prevent the performance degradation that occurs when network layers become deep. The extreme deep network leads to the state-of-the-art performance in many tasks including image classification Krizhevsky et al. [2012b], object detection Liu et al. [2019]; Redmon et al. [2016]; Ren et al. [2017] and segmentation Chen et al. [2018a]; He et al. [2017b]. However, this is very expensive concerning computation resource and heavy-burden He et al. [2016b]; Krizhevsky et al. [2012b]. To cater to the appeal for real-world applications, there is a trend to tailor the heavy-burden networks to the light-weight network in recent years.

Feature Representation Considering the redundant information contained in the 3D scene completion, the first spectrum of work try to model the scene with sparse feature representation. Specifically, OctNet Riegler et al. [2017] and O-CNN Wang et al. [2017] utilize the Octree-based CNN to represent the 3D object shape. PointNet Charles et al. [2017] and Kd-Networks Klokov and Lempitsky [2017] employ point clouds to indicate the occupation of the scene. Although saving the memory and computation, the neighbor pixels are usually mapped to the same voxel, which inevitably causes the detail missing for semantic labeling and scene understanding.

Group Convolution Recently, there are several popular light-weight networks have been proposed, include MobileNet Howard et al. [2017]; Sandler et al. [2018] and ShuffleNet Zhang et al. [2018b]. In MobileNet, depth-wise convolutions and point-wise convolutions are utilized to separate the channels as well as reduce the parameters and the calculations. In ShuffleNet, besides the group point-wise convolution and depth-wise convolution are adopted, shuffle layer is developed for information exchange between different shuffle units. However, the above models heavily rely on depth-wise convolution and group convolution, and mainly target at 2D networks thus can not directly be applied for 3D tasks.

Spatial Group Convolution To improve the computing efficiency of the 3D network. EsscNet Zhang et al. [2018a] is introduced, rather than to conduct the group convolution on feature channel dimension, which adopts the group convolution on the spatial aspect. The limitation of spatial group convolution is that it splits the features manually into separate parts, which cause the performance drops. Meanwhile, the splitting process involves hash table maintaining and coordinate with other blocks, and is cumbersome for transplantation. On the contrary, the proposed DDR (proposed in Chapter 7) block is much flexible, and it can be planted to any network which contains the 3D modules.

2.4.3 Modality Fusion in SSC

There are many works focused on RGBD fusion in 2D applications Chang et al. [2017]; Gupta et al. [2015]; Park et al. [2017]; Qi et al. [2017]; Wang et al. [2016]. RGBD sensor can capture the depth and color images simultaneously, although depth can be used to infer the geometry of the scene, which is too sparse to reconstruct the occluded parts of the scene. Compared with depth, color image carries more cues about texture, color, and reflections, which can be viewed as an essential complement to the depth for SSC task. Following the design philosophy of SSCNet, TS3D Garbade et al. [2018] adds the color image into the work-flow. However, the scene labeling needs to be estimated twice, and the depth flow and color flow are still apart from each other from the essential.

In Guedes et al. [2018], two fusion strategies were proposed, one is early-stage fusion which concatenates the feature at the first layer, and another is mid-level fusion which concatenates the features before the output layer. Although follow the overall design and reuse the features of SSCNet, the performance of adopting both fusion strategies are unexpectedly worse than that of SSCNet.

The most related work for feature fusion is RDFNet Park et al. [2017], which utilizes multi-scale fused features from color images, and aims to build a 2D segmentation framework. However, fusing the features in the 3D network is much more challenging as mentioned before. In chapter 9, we propose a novel fusion strategy which effectively fuses the 3D depth and color features on multi-scales without bringing in extra parameters.

RGB-D information fusion is vital to many vision applications. In general, the fusion scheme can be divided into three categories, *e.g.*, early fusion Couprie et al. [2013], middle fusion Ren et al. [2012], and late fusion Simonyan and Zisserman [2014a]; Yue-Hei Ng et al. [2015]. According to the stages of fusion, these schemes can also be divided into single-stage fusion and multi-stage fusion Hazirbas et al. [2016]; Park et al. [2017].

Single-Stage Fusion There are several general patterns for single-stage fusion. Specifically, Sum fusion Hazirbas et al. [2016]; Li et al. [2019a] computes the sum of the two feature maps at the same spatial locations. Average fusion is essentially a weighted sum fusion with equal weights. Max fusion Kang et al. [2014] takes the feature with the maximal value from multiple feature maps. Concatenate fusion stacks the features with channels Couprie et al. [2013]; Guo and Chen [2018]. Wang *et al.* (Wang et al. [2016]) propose an encoder-decoder architecture that exchanges the information of multi-modal data in the latent space. Bilinear fusion Lin et al. [2015] computes an outer matrix product of the two features at each pixel location.

There are a few methods that consider the complementary and selectivity of data fusion. Specifically, Li *et al.* (Li et al. [2016]) develop a novel LSTM model to fuse scene contexts adaptively. Cheng *et al.* (Cheng et al. [2017b]) use the concatenated feature maps of RGB and depth to learn an array G to weight the contribution of one input modality and $1 - G$ to weight the other input modality. Wang *et al.* (Wang and Gong [2019]) use the same strategy as Cheng et al. [2017b] to fuse the feature maps from RGB and Depth in saliency detection. However, Li *et al.* (Li et al. [2016]) only consider the complementarity of information but ignore the selectivity of the data; the other two methods only consider the selectivity of information and cannot guarantee the complementarity of information. Moreover, these methods are single-stage fusion and lack of scalability.

Multi-stage Fusion According to the way for aggregating multi-modal information, we divides multi-stage fusion algorithms into merge fusion, cross fusion, and external fusion. Hazirbas *et al.* (Hazirbas et al. [2016]) adopt the merge fusion structure to fuse the two branches of features extracted from RGB and depth images. The feature maps from depth are fused into the RGB branch by stages with an element-wise summation. Wang *et al.* (Wang et al. [2016]) use cross fusion to merge the common features of RGB and depth and keep the modality-specific features separated from each other. Both Park *et al.* (Park et al. [2017]) and Li *et al.* (Li et al. [2019a]) use an external fusion mechanism. Specifically, Li *et al.* (Li et al. [2019a]) capture features of RGB and depth image at different levels, these features at each level are fused separately and then assembled all at once before the reconstruction part. Park *et al.* (Park et al. [2017]) propose RDFNet to fuse multi-modal features separately using multiple fusion blocks, and refine the fused features one by one through a set of refining blocks. In RDFNet, each fusion introduces an additional fusion block with a new set of extra parameters. The artificially designed fusion blocks are complex and require multiple parameters that are not easy to migrate to other applications. These multi-stage fusion methods use high-level and low-level features achieving high accuracy. However, each fusion block within the multi-stage mostly adopts concatenation or summation, ignoring the adaptive selection of the multi-modal data.

In Chapter 9 we propose a GRF fusion block to extend the standard gated recurrent unit (GRU), where the gate and the memory structures can adaptively select and preserve valid information. Besides, GRFNet adopts the form of a recurrent network. When performing multi-stage fusion, GRF modules exploit parameter sharing. Our experiments show that both of the proposed single- and multi-stage GRFNets achieve better accuracy than previous methods.

Bibliography

A. Arnab and P. H. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017.

L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018.

O. Barinova, V. Lempitsky, and P. Kholi. On detection of multiple object instances using hough transforms. *TPAMI*, 34(9):1773–1784, 2012.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS– improving object detection with one line of code. In *ICCV*, 2017.

P. P. Busto, A. Iqbal, and J. Gall. Open set domain adaptation for image and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserv- ing crowd monitoring: Counting people without people mod- els or tracking. In *CVPR*, 2008.

A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv:1709.06158*, 2017.

R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85, 2017.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018a.

Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018b.

Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017b.

J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017a.

J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7415–7424, 2018.

Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, pages 3029–3037, 2017b.

C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

P. Felzenszwal, R. Girshick, and D. McAllester. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

M. Garbade, J. Sawatzky, A. Richard, and J. Gall. Two stream 3d semantic scene completion. *arXiv:1804.03550*, 2018.

R. Girshick. Fast R-CNN. In *ICCV*, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2012.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

A. B. S. Guedes, T. E. de Campos, and A. Hilton. Semantic scene completion combining colour and depth: preliminary experiments. *arXiv preprint arXiv:1802.04735*, 2018.

J. Guo, P. Ren, A. Gu, J. Xu, and W. Wu. Locally adaptive learning loss for semantic image segmentation. *arXiv preprint arXiv:1802.08290*, 2018.

Y. Guo and T. Chen. Semantic segmentation of rgbd images based on deep depth regression. *Pattern Recognition Letters*, 109:55–64, 2018.

Y. X. Guo and X. Tong. View-volume network for semantic scene completion from a single depth image. *arXiv:1806.05361*, 2018.

S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *IJCV*, 112(2):133–149, 2015.

C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *ACCV*, pages 213–228. Springer, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016a.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *ICCV*, 2017a.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017b.

J. Hosang, R. Benenson, and B. Schiele. A convnet for non-maximum suppression. In *GCPR*, 2016.

J. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. In *CVPR*, 2017.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740, 2014.

R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *ICCV*, pages 863–872, 2017.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012a.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012b.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

J. Li, Y. Liu, D. Gong, Q. Shi, X. Yuan, C. Zhao, and I. Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019a.

Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019b.

Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, pages 541–557. Springer, 2016.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017a.

T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar. Microsoft COCO: common objects in context. In *ECCV*, 2014.

T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015.

T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017b.

M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.

Y. Liu, L. Liu, H. Rezatofighi, T.-T. Do, Q. Shi, and I. Reid. Learning pairwise relationship for multi-object detection in crowded scenes. *arXiv preprint arXiv:1901.03796*, 2019.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. *arXiv preprint arXiv:1807.09190*, 2018.

K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *arXiv preprint arXiv:1709.06031*, 2017.

F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. IEEE Conf. 3D Vision*, pages 565–571. IEEE, 2016.

D. K. Naik and R. J. Mammone. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE, 1992.

T. Nguyen, T. Ozaslan, I. D. Miller, J. Keller, G. Loianno, C. J. Taylor, D. D. Lee, V. Kumar, J. H. Harwood, and J. Wozencraft. U-net for mav-based penstock inspection: an investigation of focal loss in multi-class segmentation for corrosion identification. *arXiv preprint arXiv:1809.06576*, 2018.

W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. In *CVPR*, 2013.

S. J. Park, K. S. Hong, and S. Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

T. T. Pham, S. H. Rezatofighi, I. Reid, and T.-J. Chin. Efficient point process inference for large-scale object detection. In *CVPR*, 2016.

H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *CVPR*, pages 5199–5208, 2017.

S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, June 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2577031.

X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, pages 2759–2766, 2012.

S. H. Rezatofighi, V. Kuma, A. Milan, E. Abbasnejad, A. Dick, and I. Reid. DeepSetNet: predicting sets with deep neural networks. In *ICCV*, 2017.

G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, volume 3, 2017.

M. Rodriguez, I. Laptev, and J. Sivic. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

R. Rothe, M. Guillaumin, and L. V. Gool. Non-maximum suppression for object detection by passing messages between windows. In *ACCV*, 2014.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014a.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014b.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198, 2017.

J. Sun, D. Yu, Y. Li, and C. Wang. Mask propagation network for video object segmentation. *arXiv preprint arXiv:1810.10289*, 2018.

S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *ICCV*, 2013.

L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision*, pages 537–547, 2017.

M. B. R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

J. Wang, Z. Wang, D. Tao, S. See, and G. Wang. Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks. In *ECCV*, pages 664–679, 2016.

N. Wang and X. Gong. Adaptive fusion for rgb-d salient object detection. *arXiv:1901.01369*, 2019.

P. S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *TOG*, 36(4):72, 2017.

Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Adversarial semantic scene completion from a single depth image. In *International Conference on 3D Vision*, pages 426–434. IEEE, 2018.

J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, pages 82–90, 2016.

S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018.

L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.

A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: global multi-object tracking using generalized minimum clique graphs. *ECCV*, 2012.

J. Zhang, H. Zhao, A. YaoE, Y. Chen, L. Zhang, and H. LiaoE. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, pages 733–749, 2018a.

X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018b.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Y. Zou, Z. Yu, B. Kumar, and J. Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

Chapter 3

Learning Pairwise Relationship for Multi-object Detection in Crowded Scenes

As the post-processing step for object detection, non-maximum suppression (GreedyNMS) is widely used in most of the detectors for many years. It is efficient and accurate for sparse scenes, but suffers an inevitable trade-off between precision and recall in crowded scenes. To overcome this drawback, we propose a Pairwise-NMS to cure GreedyNMS. Specifically, a pairwise-relationship network that is based on deep learning is learned to predict if two overlapping proposal boxes contain two objects or zero/one object, which can handle multiple overlapping objects effectively. Through neatly coupling with GreedyNMS without losing efficiency, consistent improvements have been achieved in heavily occluded datasets including MOT15, TUD-Crossing and PETS. In addition, Pairwise-NMS can be integrated into any learning based detectors (Both of Faster-RCNN and DPM detectors are tested in this paper), thus building a bridge between GreedyNMS and end-to-end learning detectors.

3.1 Introduction

Multiple Object detection is a vital problem in computer vision, and can be viewed as the cornerstone for instance-level segmentation Arnab and Torr [2017]; He et al. [2017]; Urtasun [2017] and multi-object tracking Leibe et al. [2007]; Tang et al. [2013]; Zamir et al. [2012]. In multi-object detection, several objects are usually overlapping with each other as a cluster, which is called co-occurrence Ouyang and Wang [2013]; Rodriguez et al. [2011]; Sadeghi and Farhadi [2011]. The problems that arise with co-occurrence are challenging such as occlusion, illumination changing, texture inconsistency, etc. Among these problems, occlusion is the most notorious issue. Based on whether the clusters of objects belong to the same category or not, the co-occurrence can further be classified into intra-class co-occurrence (also called crowded scene) and inter-class co-occurrence. Because different object categories usually contain different discriminative features, the problem of inter-class co-occurrence is relatively easier to solve than the one of intra-class co-occurrence. In particular, due to the incredible contribution of deep neural network and huge training data Ref; He et al. [2016]; Krizhevsky et al. [2012]; Russakovsky

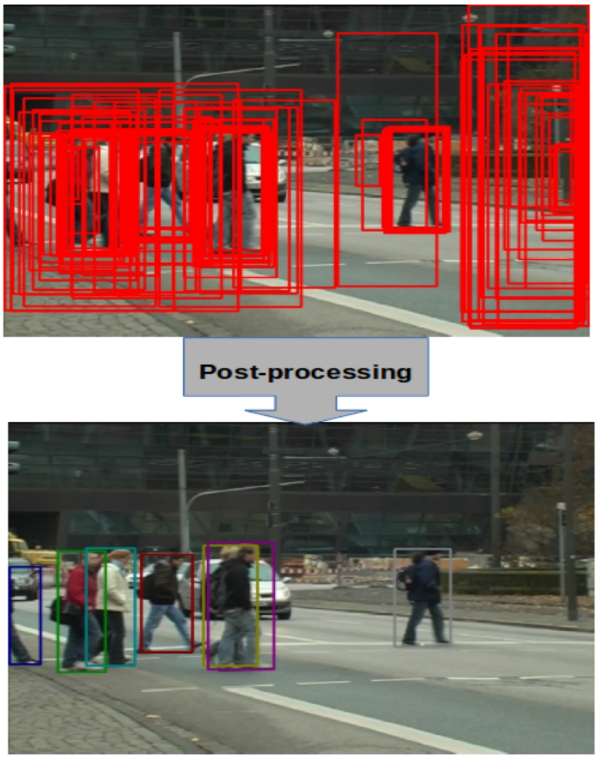


Figure 3.1: Detection proposals to final bounding boxes: for object detection, on the top, (100) detection proposals (generated by Faster-RCNN) detector, will pass through a post-processing step (GreedyNMS is used by default), and acquire the final bounding boxes as shown at the bottom.

et al. [2015], significant improvements have been achieved in object detection Girshick [2015]; Girshick et al. [2012]; He et al. [2014]; Lin et al. [2017]; Ren et al. [2015] based on the extremely accurate per-category classification in recent years, thus inter-class co-occurrence is not a big deal anymore for object detection. However, intra-class co-occurrence still remains as a challenging problem, and most of the popular detectors fail when they come to multiple overlapping objects in crowded scenes.

In order to understand the essence of the problem brought by intra-class co-concurrence in multi-object detection, it is necessary to know the work principle and evaluation criteria of object detection. With the developments for object detection in many years, in general, a common pipeline for object detection includes three stages: proposals generation, object detection, and post-processing (using GreedyNMS by default) Felzenszwal et al. [2010]; Girshick [2015]; Girshick et al. [2012]; Ren et al. [2015]. Figure 3.1 shows an example from the detection proposal to the final bounding boxes. Meanwhile, for evaluation, Mean Average Precision (MAP) is used as the standard criterion Everingham et al. [2015]; Lin et al. [2014]. The reason that detectors fail in crowded scenes is that several objects are often closed by each other and heavily occluded. The post-processing step with GreedyNMS is not suitable in such scenarios as it is only capable of achieving good performance for isolated objects in sparse scenes. Specifically, the de facto used GreedyNMS usually suppresses harshly, leading to many missed detections in crowded scenes. Despite this limitation, GreedyNMS is efficient and achieves good performance on sparse scenes, so it has long been the default post-processing step in object detection.

Targeting a more robust detector for multi-object detection, and overcoming the drawback of GreedyNMS, we propose a new NMS algorithm called Pairwise-NMS. Besides efficiently handling isolated objects like GreedyNMS does, it also takes the heavily occluded case into consideration, and dramatically improves the detection performance in crowded scenes. Specifically, for dealing with two nearby proposals, Pairwise-NMS inherits the GreedyNMS framework when the intersection over union (IoU) is smaller than a pre-defined NMS threshold N_t, since GreedyNMS is efficient and accurate in this case. However, when the IoU is larger than the NMS threshold N_t, it uses a pairwise-relationship network to predict how many objects the two proposals contain, which helps to handle the multiple overlapping objects effectively. As can be seen in the workflow of Pairwise-NMS in Figure 3.2, comparing to GreedyNMS, Pairwise-NMS is essentially a more general form to handle multi-object detection. To sum up, the contributions of this paper are as follows.

- An end-to-end Pairwise-NMS relationship network is proposed for replacing the GreedyNMS, which achieves better performance for object detection without losing efficiency.
- We evaluate the proposed method on three public datasets, both qualitative and quantitative results demonstrate that our method performs better than GreedyNMS especially in the crowded scene. The proposed method also achieves better performance than the recent Soft-NMS Bodla et al. [2017].
- Thanks to its flexibility, Pairwise-NMS can be integrated into learning based detectors (e.g., Faster-RCNN and DPM) neatly, and pave the way for the end-to-end learning detectors.

3.2 Related works

3.2.1 Non-Maximum Suppression (NMS)

Sub-Optimal Solver. From the mathematical perspective, non-maximum suppression can be viewed as a quadratic problem. In object detection, the inverse scores of the proposals are treated as the unary terms and IoUs between the proposals are treated as the binary terms Rezatofighi et al. [2017]. The scores are generated by detector based on the encoding of localizations between proposals and ground-truth bounding boxes. The role of GreedyNMS is to assign zero if the IoU between two proposals is smaller than a NMS threshold or infinite vice versa. By minimizing an energy function, the problem can be solved through a quadratic optimization solver. As pointed out by previous works Barinova et al. [2012]; Chan et al. [2008]; Pham et al. [2016]; Pirsiavash et al. [2011], instead of using GreedyNMS, which greedily iterates and acquires local minimum, using other solvers can provide better local minima. While theoretically appealing, there is an increasing in the complexity in both memory and time by using quadratic optimization solvers, especially when the number of proposals is large Barinova et al. [2012]; Chan et al. [2008]; Pham et al. [2016]; Pirsiavash et al. [2011].

Message Passing. Targeting at object detection in crowded scenes, Rasmus *et al.* Rothe et al. [2014] propose to use a latent structured SVM to learn the weights of affinity propagation clustering between proposals. By passing messages between windows, the merging step can be executed by using clustering methods, and proposals can be re-weighted through minimizing energy function. This kind of method

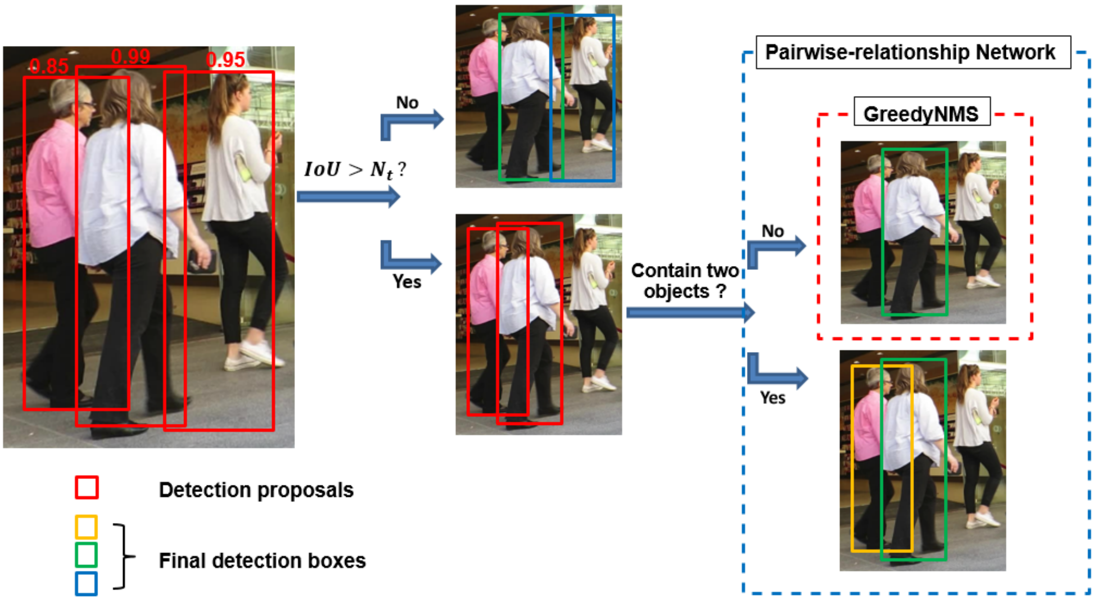


Figure 3.2: Workflow of Pairwise-NMS. Left: Detection proposals with confident scores are acquired by the detector, the one with highest score will be picked as the final detection result (the middle one, also called 'the maximal' proposal), and other proposals which are overlapped with the 'the maximal' one will be checked (also called 'non-maximal' proposals. Middle: The proposal pairs (between the maximal one and non-maximal ones) will flow into two branches depend on if their IoUs are larger than NMS threshold N_t. Right: For the heavily overlapped proposal pairs (in which IoU of the proposal pair is larger than N_t), GreedyNMS will supress the non-maximal proposal without consideration. Pairwise-NMS will smartly decide if the non-maximal proposal should be kept depend on the number of objects that the proposal pairs contain.

can handle cases which contain two objects that are very close to each other. However, it is hard to distinguish the boundary between one object and two objects.

Soft-NMS. Both sub-optimal solver and message passing methods belong to the branch of improving the binary term, *i.e.* re-weight. There are several works focus on improving the unary term, that is re-score, of the proposals. One of the interesting works is Soft-NMS Bodla et al. [2017]. Comparing to GreedyNMS, Soft-NMS decreases the detection scores as an increasing function of IoUs between the target proposals and the overlapped ones, rather than setting scores of the overlapped proposals to zero. Although just one line of code is added, some gains in performance among several detection datasets were acquired. To be specific, a linear or gaussian function is employed to re-score the non-maximal proposals rather than to totally suppress them. The drawback of this method is that it has to manually tailor a corresponding function to alleviate GreedyNMS. This is troublesome since the balance between recall and precision is usually trade-off and data driven. Furthermore, as showed in the experiment section, it can not handle the heavily occluded cases very well.

ConvNet NMS. Very recently, Jan Hoang *et al.* Hosang et al. [2016, 2017] propose to use a neural network to re-score the proposals. Being different from other works, they argue that NMS should be totally replaced with the convolutional neural network. The approach in Hosang et al. [2016, 2017] can be used to learn an end-to-end solution for object detection. However, we note that for the for sparse scenes, GreedyNMS performs well and efficient. Hence GreedyNMS should not be totally removed from an object detection pipeline. In addition, Learning-NMS Hosang et al. [2017] requires a large memory cost, so it limits the number of the sampling proposals for training the network.

Pairwise Prediction. Based on the same application scenario as Rothe et al. [2014], Tang *et al.* Tang et al. [2013] use a SVM to handle the object pairs in the crowded scene, but with manually designed features. It is hard to handle the case when two nearby objects with the similar appearance.

Compare to previous works, our work also focuses on predicting object pairs like Tang et al. [2013], but we employs a deep learning based network. In addition, rather than throwing away NMS framework totally like Learning-NMS Hosang et al. [2017], the good merits of NMS are retained while the hard constraints are mitigated. Specifically, we utilize the features of the region of interest (ROI) to pass messages between pairs. After the pairwise relationship matrix of the detections within one image was learned, only two lines of codes are added to improve the GeedyNMS (Please refer to Section 3.3.6 for details). The proposed method is also simple as Soft-NMS Bodla et al. [2017], but it provides a more robust solution for multi-object detection under heavy occlusions.

3.3 Methodology

3.3.1 Overview

The standard way to evaluate the inference performance of an object detection method is via mean average precision (MAP). Therefore, both precision and recall are taken into account. Precision is defined as the ratio between the number of true positives and the number of detections, whereas recall is the ratio between the number of true positives and the number of ground-truth bounding boxes. The goal of detection to achieve good performance is to acquire more true positives and less false positives. As the post-processing step of detection, GreedyNMS solely relies on the detection scores to pick the highest-scored proposals while suppressing the overlapping non-maximal proposals in the surrounding. In sparse scenes, GreedyNMS is capable of achieving good performance as there are only very few overlapping objects, in which cases, the highest-scored proposals will be retained to assign to the target objects. However, in crowded scenes, several objects usually are very close to each other. As a result, a bunch of proposals which are likely to be true positives always have similar scores and form a cluster. In this case, GreedyNMS will inevitably lead to a sacrifice between precision and recall, and dramatically harm the overall performance.

In order to generate a more robust detector to handle the general scenes, a new NMS algorithm for post-processing of object detection is required. There are three points should be taken into consideration:

- The algorithm is not computationally expensive.
- Heavily occluded objects should be taken into considerations.
- It does not bring in more false positives when improving the recall.

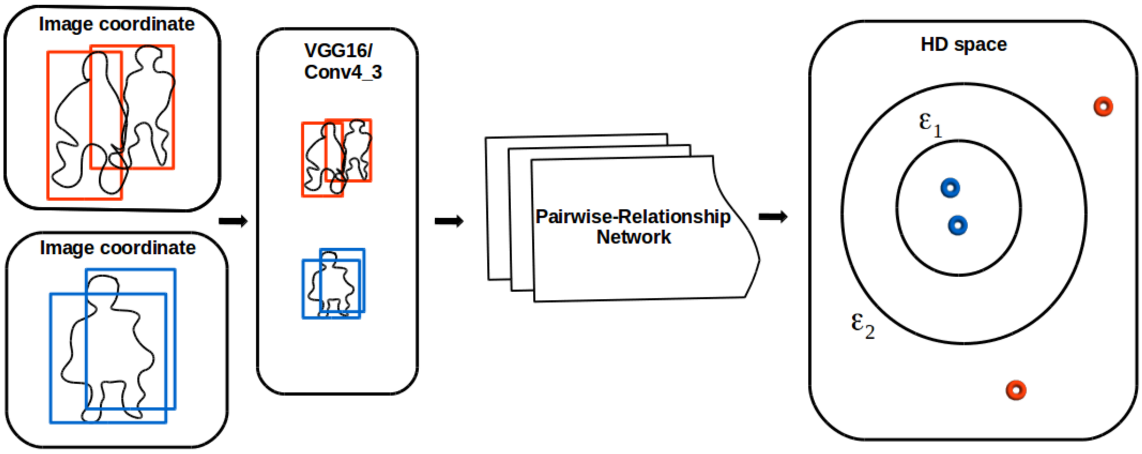


Figure 3.3: Mapping two nearby proposals from image coordinate to high-dimension space. From left to right: The overlapped proposal pairs in image coordinates are first fed into the backbone network (VGG16), their corresponding features extracted from layer Conv4-3 are formed the RoI feature pairs, and used as the inputs of our pairwise-relationship network. Each pair will be mapped into the high-dimention (HD) space through the neural network, and the proposal pair which contain two objects will have a large distance in HD space than the proposal pair which contains zero/one object.

In this paper, a new NMS framework is explored, that is Pairwise-NMS. Unlike GreedyNMS, Pairwise-NMS can hold the detection boxes for the nearby objects even when they heavily occluded with each other. Through neatly couping with GreedyNMS without losing efficiency, Pairwise-NMS provides a robust and general solution to the post-processing step for object detection.

3.3.2 Pairwise-NMS

Figure 3.2 is the workflow of Pairwise-NMS. The algorithm is fed with the detection proposals before NMS as inputs. It is then divided into two branches based on NMS threshold N_t. For the top branch, when the IoU of two overlapping proposals is smaller than N_t, Pairwise-NMS inherits from GreedyNMS directly because it performs well and very efficient. For the bottom branch, when the IoU of two overlapping proposals is larger than N_t, GreedyNMS has a poor recall. Instead of suppressing all of the surrounding non-maximal proposals as GreedyNMS does, the proposal pairs will be fed into a pairwise-relationship network. It will make the Pairwise-NMS to know whether the proposal pairs contain two objects or other cases (contain zero or one object). If the proposal pair contains two objects, then both of the proposals should be kept no matter how close they are. If the proposal pair contains zero or one object, then the non-maximal proposal would be merged like GreedyNMS does. By doing so, multiple overlapping objects can also be handled effectively. Therefore, Pairwise-NMS can consistently improve the recall and MAP, and considerably increase the performance especially in heavily occluded scenes.

Table 3.1: Relationships between two proposals

B_i, B_j	nearby or not	objects	constraints in HD-space
case1	no	0	$\parallel Z_i - Z_j \parallel \geq \epsilon_2$
case2	no	1	$\parallel Z_i - Z_j \parallel \geq \epsilon_2$
case3	no	2	$\parallel Z_i - Z_j \parallel \geq \epsilon_2$
case4	yes	0	$\parallel Z_i - Z_j \parallel \leq \epsilon_1$
case5	yes	1	$\parallel Z_i - Z_j \parallel \leq \epsilon_1$
case6	yes	2	$\parallel Z_i - Z_j \parallel \geq \epsilon_2$

3.3.3 Pairwise Relationship Network

The goal of pairwise-relationship network is to tell Pairwise-NMS how many objects the two overlapping proposals contain, and let the Pairwise-NMS to decide if the non-maximal proposals should be merged or not. It is challenging to differentiate two heavily overlapping objects which belong to the same category in an image, especially when they have similar shapes as well as appearances. Based on this observation, the pairwise-relationship network is designed. We expect that after mapping, two closed objects in 2D image coordinate will have a large distance in high dimensional space, as shown in Fig 3.3. Then, based on the distance that two proposals have in high-dimension (HD) feature space, it can easily infer whether the two proposals contain two objects or not. The input of pairwise-relationship network are features of the proposals from the backbone network (here we use VGG16 Simonyan and Zisserman [2014] as an example, because its simplicity, while it can be replaced with a deeper network structure like ResNet He et al. [2016], GooleNet Szegedy et al. [2016]), and the output is a distance value between the two HD vectors learned from the pairwise-relationship network.

3.3.4 Define Pairs

Considering how many objects two proposals contain and whether they are close to each other (Specifically, if IoU of two proposals is larger than NMS threshold, then we define them as “nearby” or vice versa), there are six cases as shown in Table 3.1. Case1∼case3 are not under considerations due to the two proposals are not overlapped with each other. For better elucidation, we define case4 and case5 as similar pairs and case6 as dissimilar pairs regarding how many objects the two proposals contain. For similar pairs, in which two proposals contain zero or one object, they should be merged into one bounding box. For dissimilar pairs, where two proposals contain two objects, both of the two proposals should be kept. The pairwise-relationship network mainly focus on predicting if two proposals are similar pairs or dissimilar pairs.

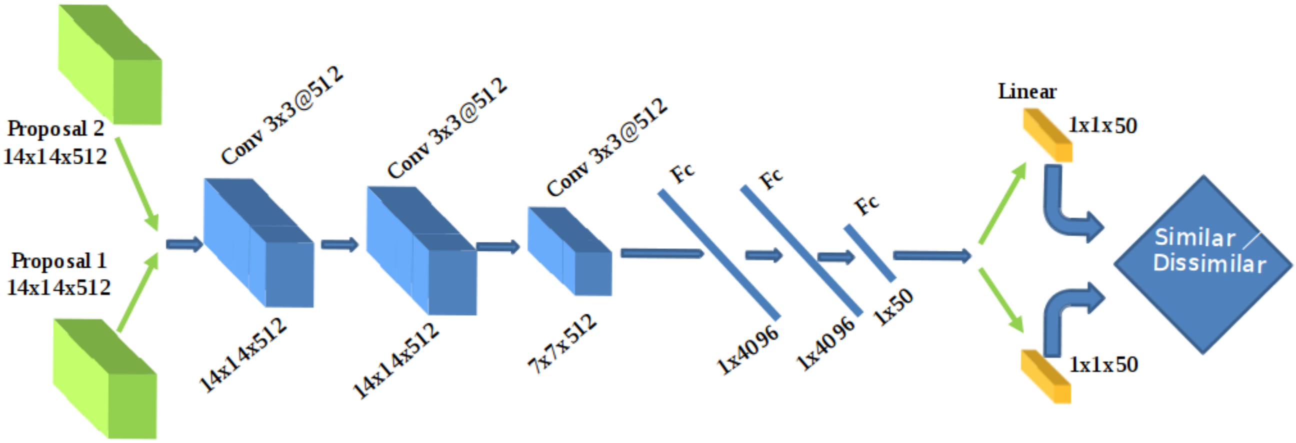


Figure 3.4: Pairwise-relationship network structure. From left to right: The 512-dimension RoI feature pairs are used as input, after three Convolution layers, three FC layers, and one Linear layer (global average pooling Lin et al. [2013] is used in our experiments), they are mapped into a 50-dimension space. Relu and Batch Normalization are applied all of the layers except the last one.

3.3.5 Network and Loss

Features of Proposal Pairs The goal of pairwise-relationship network is to learn the relationship between two overlapping proposals. The inputs to the network are the corresponding ROI features of the proposal pairs. We extract ROI features from the layer Conv4_3 of VGG16 Simonyan and Zisserman [2014], the size of ROI feature is 1/8 of the corresponding proposal in raw image. The reason for sampling ROI features from Conv4_3 is because it is small enough, which is good for the training speed and memory cost, and meanwhile carrying enough feature information that the network requires. For the purpose of training the network using the batch operation, ROIAlign He et al. [2017] is used to transfer the ROI proposals to the 14 × 14 fixed size tensors.

Network Layers As shown in Figure 3.4, the main body of the network structure is composed of three convolutional layers, three fully connected layers and a global average pooling layer Lin et al. [2013]. Each convolutional layer is followed by a Relu Krizhevsky et al. [2012] and Batch Normalization Ioffe and Szegedy [2015] layer. From Conv2 to Conv3, Max-pooling is used. The feature dimension for convolutional layers is 512, and the kernel size and stride are 3 and 1, respectively. In the last FC layer, the dimension of the feature vector is reduced to 50, which is the dimension of the HD space being used to calculate the distance of the two mapped vectors.

Global Average Pooling Global average pooling(GAP) Lin et al. [2013] is employed after the last FC layer to transfer the tensor to a HD vector. We also conduct some tests to replace the GAP layer with FC layer. Our experimental results demonstrate that GAP layer has a better performance than FC layer for the task, and a brief analysis is provided in Section 3.4.7

Loss We use L1-norm loss because it converges fast, and the loss function is defined as following:

$$\begin{aligned} L(x_1, x_2) = & y \cdot |f(x_1) - f(x_2)| + \\ & (1-y) \cdot \max(0, \lambda - |f(x_1) - f(x_2)|) \end{aligned} \tag{3.1}$$

Table 3.2: Experiment setting of Pairwise-NMS and GreedyNMS on MOT15 and TUD-Crossing

E_t	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
N_t	0.55	0.55	0.55	0.6	0.65	0.7	0.8	0.85	1	0.95
D_t	1.25	1.25	1.25	1.4	1.3	0.65	0.9	0.5	0	0.2

In Formula 3.1, x_1 and x_2 are ROI proposals, the label $y = 1$ when the two proposals belong to similar pairs (zero or one object) and $y = 0$ when the two proposals belong to dissimilar pairs (different objects). $f(x_1)$ and $f(x_2)$ are HD vectors learned from the pairwise-relationship network. A margin $\lambda = 1$ is used to push the dissimilar pairs away from the similar pairs. $|f(x_1) - f(x_2)|$ is the L_1 distance between the two HD vectors. For similar pairs, the distance value between two HD vectors would be forced to approach to 0, while for dissimilar pairs, the distance value between two HD vectors would be forced to approach to 1.

Training and Inference For training, the optimizer is stochastic gradient descent (SGD) with momentum Krizhevsky et al. [2012]. For all of the experiments, the following setting is adopted, where learning rate, momentum, weight decay and batch size are 1e-4, 0.9, 1e-5 and 1 respectively. For inference, on each image, a symmetric and sparse pairwise relationship matrix is acquired for the detection proposals, which record the pairwise-distance of any two detection proposal within the image, and which is used for couping with GreeyNMS. ON all of the datasets, the total sample proposal pairs during training on every image are 64 for proposals generate by DPM Felzenszwal et al. [2010] and 32 for proposals generated by Faster-RCNN, and the ratio of dissimilar pairs and similar pairs is 1:3.

3.3.6 Coupling with GreedyNMS

Just like Soft-NMS Bodla et al. [2017], Pairwise-NMS can also be coupled with GreedyNMS neatly. The following is the coupling code of Pairwise-NMS, after the pairwise relationship maxtrix is learned through the pairwise-relationship network. The colored parts are the differences between GreedyNMS and Pairwise-NMS. As can be seen, in GreedyNMS, all of the non-maximal proposals would be suppress, regardless how many objects it has in the surrounding area, which will inevitably cause the performance drop under the heavily occluded cases. On the contrary, our method provide an effective complementary to the GreedyNMS, that is, even when IoU of the non-maximal proposal with the selected proposal is larger than NMS threshold, those two proposals (the proposal pair) will be passed into the pairwise-relationship network, and let the network automatically decided how many objects the two proposals contain. Compared with the 'blind' suppression of GreedyNMS, Pairwise-NMS is much flexible and robust for multi-object detection. And comparing to GreedyNMS, just one line of code is added, thus no efficiency lose.

Algorithm 1: Pusedo code of Pairwise-NMS

```
Require: $\beta = \{b_1, ...b_N\}$, $S = \{s_1, ...s_N\}$, $N_t$
    $\beta$ is the list of initial detection boxes
    $S$ contains corresponding detection scores
    $N_t$ is the NMS threshold
    $D_t$ is the distance threshold
begin
    $D \leftarrow \{\}$
    while $\beta \neq empty$ do
        $m \leftarrow argmax\ \ S$
        $M \leftarrow b_m$
        $D \leftarrow D \cup M; B \leftarrow B - M$
        for $b_i$ in $\beta$ do
            if $iou(M, b_i) \geq N_t$ then
                $\beta \leftarrow \beta - b_i; S \leftarrow S - S_i$
            end if
                                        GreedyNMS
            if $iou(M, b_i) \geq N_t \wedge$
            $pairDist(M, b_i) \leq D_t$ then
                $\beta \leftarrow \beta - b_i; S \leftarrow S - S_i$
            end if
                                        Pairwise-NMS
        end for
    end while
end
```

3.4 Experiments

3.4.1 Experiment Glimpse and Evaluation

Experiment Glimpse

In this part, we provide extensive experiments on three different datasets, including MOT15, TUD-Crossing and PETS. The experimental results firstly demonstrate how our Pairwise-NMS outperforms GreedyNMS in crowded scenes with occlusions to different extents, especially, the gains with the increasing of the occlusions that Pairwise-NMS over GreedyNMS will be emphasized. Moreover, the comparison with Soft-NMS demonstrate our method performs better in very crowded scenes.

About Evaluation

There are two parameters to be considered during the execution of NMS related to the final detection performance. One of the parameters is NMS threshold N_t, and it determines to what extent the surrounding non-maximal proposals should be suppressed by the local maximal proposal. Another parameter is the evaluation threshold E_t, and it determines to what extent a detection bounding box could be true positive (if only one detection box is aligned to the object) or false positives (if multiple detection boxes are aligned to the same object).

Table 3.3: AP of optimal GreedyNMS vs Pairwise-NMS on MOT15

Evaluation threshold	GreedyNMS	Pairwise-NMS	Outperform
AP @ 0.5	74.11	**74.15**	0.04
AP @ 0.55	76.15	**76.21**	0.06
AP @ 0.6	65.66	**65.82**	0.16
AP @ 0.65	57.86	57.86	0.0
AP @ 0.7	45.68	**46.41**	0.73
AP @ 0.75	29.45	**30.41**	0.96
AP @ 0.8	16.70	**16.82**	0.12
AP @ 0.85	5.17	**5.26**	0.09
AP @ 0.9	0.87	0.87	0.0
AP @ 0.95	0.02	0.02	0.0

When a small evaluation threshold (for example $E_t = 0.5$) is used, it is much easier for the detection proposals to be true positives, since the only condition to be satisfied is that their IoUs with ground truth boxes are bigger or equal than 0.5. Meanwhile, due to the constraint that the evaluation script asking for at most one proposal to align the same object, as a result, it is prone to high recall and lower precision. On the contrary, when a large evaluation threshold (for example $E_t = 0.95$) is used, it is more difficult for the detection proposals to be true positives because it relies on the detector to perform an extremely accurate localization. As a result, it will lead to high precision and low recall.

In order to perform an objective and comprehensive evaluation, especially in multi-object detection which often contains heavy occluded cases, we use the same evaluation criterion as Microsoft COCO evaluation Lin et al. [2014], and with different evaluation thresholds to reflect the strength of Pairwise-NMS for handling heavy occlusions in crowded scenes.

3.4.2 Results on MOT15

For each evaluation criterion, one optimal NMS threshold N_t and one distance threshold D_t are chosen for GreedyNMS and Pairwise-NMS respectively. The specific settings for MOT15 and TUD-Crossing are listed as Table 3.2. And the setting for PETS are following Hosang et al. [2017] and will be explained in the Section 3.4.4

MOT15 benchmark Leal-Taixe et al. [2015] is a public dataset, which collects 11 sequences for multi-object detection and tracking from different cities. It is challenging due to the heavy occlusions, dramatic illumination changes and clustering background. MOT15 training set is used as it provides the ground-truth annotations, which facilitates the verification of the proposed Pairwise-NMS framework. The whole training set contains 5500 images, and is split with a ratio 5:1:4 for training, validation and test. For object detector, we use the Faster-RCNN code implemented by Ross GirshickRen et al. [2015]. That detector was originally trained on COCO Lin et al. [2014] and PASCAL VOC Everingham et al. [2015], both of which lack significant numbers of occluded objects. We therefore fine-tune the detector on the MOT15 training set as mentioned above. For all of the experiments, pairwise-relationship network was trained on training set and parameters were fine-tuned on validation set only.

Table 3.3 shows the overall performance of GreedyNMS and Pairwise-NMS on test set. We use

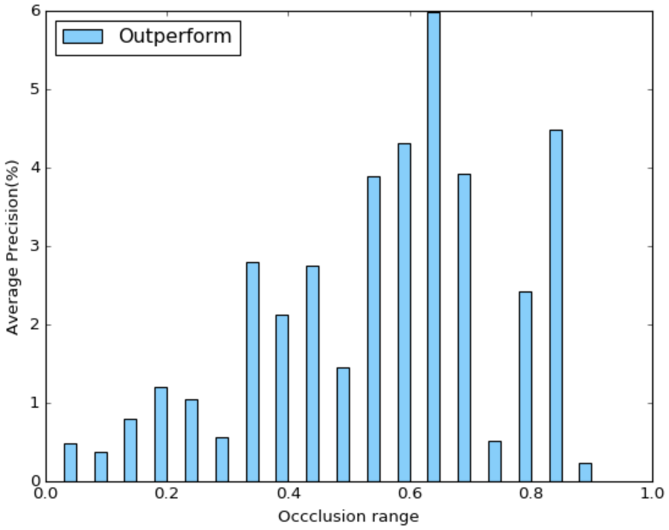


Figure 3.5: Outperform percentages of Pairwise-NMS than optimal GreedyNMS under different occlusion ranges on MOT15. Using AP@0.5 as evalution cretion, Pairwise-NMS achive 6% higher percent than that of GreedyNMS regarding to detection accuracy, and with the increasing of the occlusion range, the gains that Pairwise-NMS outperformed GreedyNMS is enlarged.

the same evaluation as Microsoft COCO Lin et al. [2014], which is $AP_{0.5}$ to $AP_{0.95}$, as the evaluation criterion. For fair comparison, the optimal NMS threshold was acquired under each evaluation threshold, and based on that, the best AP for GreedyNMS is set down as the baseline. Using the same setup with GreedyNMS, our pairwise-relationship network was trained. The pairwise matrix that is obtained using pairwise-relationship network from inference is then be coupled into GreedyNMS framework as described in Section 3.3.6. As can be seen, Pairwise-NMS outperforms GreedyNMS is eligible for $AP_{0.5}$. However, the gap is enlarged with the increasing of evaluation threshold. For example, when $AP_{0.75}$ is used, Pairwise-NMS beats GreedyNMS about 1 percent for overall performance. This improvement also double proves that our method is good at handling heavy occlusions.

Although we have achieved consistent improvements compared to GreedyNMS using different evaluation thresholds, the gains of the overall performances seem not large. In fact, it does not like so. After a thorough investigation of datasets which have heavy occlusions including TUD-Crossing and MOT15, we discover that the number of heavily occluded objects is relatively smaller comparing to the total number of ground truth bounding boxes. Besides, the brief analysis about the factors that affect the final evaluation provided in Section 3.4.1 can also provide some hints.

For better analysis about the performance of our algorithm, like the previous related papers Bodla et al. [2017]; Hosang et al. [2016, 2017], $AP_{0.5}$ for different occlusions are acquired. Here, we use the overlap between ground-truth bounding boxes as the occlusion reference. Fig 3.5 shows a full comparison of outperform percentages of Pairwise-NMS than GreedyNMS under different occlusion ranges. As can be seen in Fig 3.5, in most cases, Pairwise-NMS beat optimial GreedyNMS. The overall trend is that the heavier the occlusions, the more gains Pairwise-NMS achieve, and the max gains is 6 percent compared with GreedyNMS.

Table 3.4: F1-score of optimal GreedyNMS vs Pairwise-NMS under different occlusion ranges on MOT15

Gt overlap	GreedyNMS	Pairwise-NMS	Outperform
0.4 - 0.45	38.73	**40.25**	1.52
0.45 - 0.5	39.17	**39.18**	0.01
0.5 - 0.55	43.93	**44.72**	0.79
0.55 - 0.6	32.18	**33.28**	1.1
0.6 - 0.65	29.03	**30.0**	0.97
0.65 - 0.7	32.17	**35.90**	3.73
0.7 - 0.75	39.46	**39.73**	0.27
0.75 - 0.8	36.78	36.78	0.0
0.8 - 0.85	33.65	**34.29**	0.64
0.85 - 0.9	6.25	**6.90**	0.65

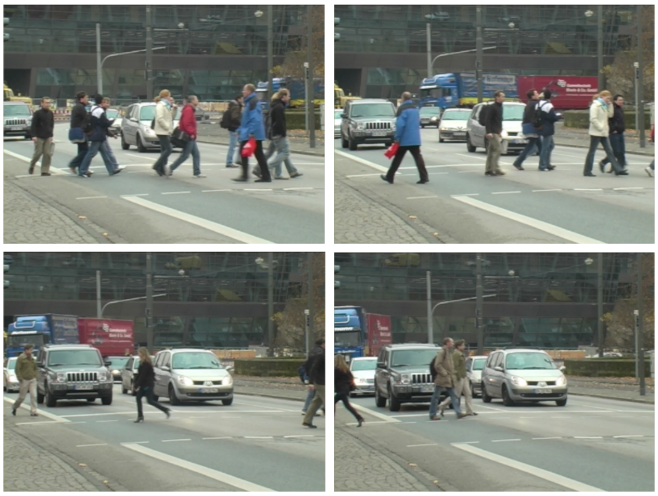


Figure 3.6: TUD-Crossing: the 50th, 100th, 150th, 200th frame of dataset, as can be seen, it is quite challenging as for object detection.

To gain further understanding on the performance of our method under heavy occlusions, F1 scores under different occlusion ranges from 0.4 to 0.9 with interval 0.05 are outlined in Table 3.4. As can ben seen, Pairwise-NMS has better F1-score than GreedyNMS in all of the thresholds. Pairwise-NMS outperforms GreedyNMS more than 3.7 percent under occlusion range between 0.65 and 0.7.

3.4.3 TUD-Crossing for Generalization

In order to verify the generalization ability of the proposed method, we use TUD-Crossing Andriluka et al. [2008] dataset, which is composed of 201 heavily occluded images, to perform the inference based on the trained model which is obtained from MOT15, without any fine-tuning on TUD-Crossing.

In Figure 3.6, there are four images with frame no 50, 100, 150, 200 are shown out. Since TUD-Crossing dataset are capture from the side view when pedestrians are coming across the traffic light, as

Table 3.5: AP of optimal GreedyNMS vs Pairwise-NMS on TUD-Crossing

Evaluation threshold	GreedyNMS	Pairwise-NMS	Outperform
AP @ 0.5	**79.07**	78.86	-0.21
AP @ 0.55	**75.58**	75.25	-0.33
AP @ 0.6	69.46	**69.47**	0.01
AP @ 0.65	57.95	**58.07**	0.12
AP @ 0.7	42.33	**42.35**	0.014
AP @ 0.75	24.41	**25.36**	0.95
AP @ 0.8	12.40	**12.56**	0.16
AP @ 0.85	4.02	**4.10**	0.07
AP @ 0.9	0.51	0.51	0.0
AP @ 0.95	0.01	0.01	0.0

can be seen, in all of the images, there are always heavily occluded cases happen, regardless how many people within the camera frustum. And for the first half of the video sequence, the camera frustum usually contain a relatively higher density. Moreover, the crowding, small scale, fractional and low resolution also make the task more challenging.

The results are shown in Table 3.5, average precision (AP) is used as the measurement metric. And for most of the evaluation criterions, Pairwise-NMS performs better than GreedyNMS regarding to detection accuracy. Specifically, the heavier occlusions are, the more gains Pairwise-NMS achieves when compared with that of using GreedyNMS, and the maximumal overall gain is 0.95 percent, and considering the ratio of heavily occluded cases just take up small part of the overall bounding boxes, this is a large improvements for object detection. To sum up, the detection results on TUD-Crossing dataset present the similar trend as the one on MOT15, which confirms that our pairwise-relationship network has acquired the ability for handling with the heavy occlusions regarding the texture and pattern changing of the image sequence.

3.4.4 Results on PETS

PETS Ellis and Ferryman [2010] dataset is another public dataset which consists of eight sub-sequences that are captured from different angles and views. We use the same setup to split the dataset as Hosang et al. [2017]; Tang et al. [2013], and the detection proposals before NMS were generated by DPM Felzenszwal et al. [2010]. Table 3.6 provides the statistics of PET dataset, including the sub-sequence splits, number of frames, detection proposals before NMS and the ground truth bounding boxes for training, validation and testing.

A same workflow as MOT15 for training and inference is shared. Table 3.7 are the results we obtain using Pairwise-NMS and GreedyNMS with different NMS thresholds, in order to be consistent with Tang et al. [2013], all of the results, with $AP_{0.5}$ as the evaluation criterion. As can be seen, our method outperforms all of the GreedyNMS results, and beats GreedyNMS with optimal NMS threshold ($N_t = 0.4$) more than 1.5 percent. It proves that, as the post-processing step, Pairwise-NMS is a good replacement for GreedyNMS and it can be integrated into any detector.

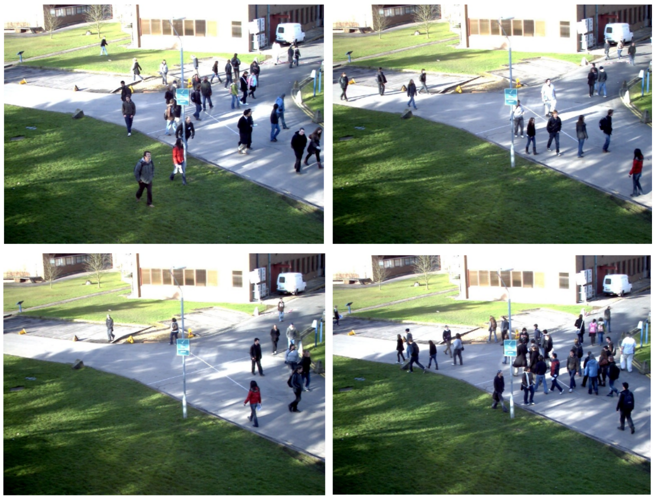


Figure 3.7: PETS test images: The 100th, 200th, 300th, 400th frame of PETS test set (S2L2 sequence) are shown out. As can be seen, the people in this scene is very crowded, quite small and affected by the light conditions, thus is extremely challenging for object detection.

Table 3.6: Statistic of PETS Dataset

splits	sequences	frames	proposals	gt boxes
training	S1L1-1, S1L1-2, S1L2-1 S1L2-2, S2L1, S3MF1	1696	5421758	23107
validation	S2L3	240	727964	4376
test	S2L2	436	2112655	10292

3.4.5 Comparison with Soft-NMS

In this section, we conduct the comparison experiments between the proposed Pairwise-NMS and the state-of-the-art method Soft-NMS Bodla et al. [2017] on PETS test dataset.

Since the most outstanding advantage of Pairwise-NMS is capable of handling with multi-object detection especially under the heavy occluded cases. PETS dataset is chosen for the reason that it is one of the most crowded and widely used datasets for multi-object detection. Thus it should be representative and is reasonable to reflect the robustness of algorithms.

According to the Soft-NMS Bodla et al. [2017] paper, instead of suppressing all of the proposals which have heavy IoUs with the targeting (maximal scored) proposal, Soft-NMS replace the hard NMS threshold used in GreedyNMS with a linear or non-linear function to re-score the overlapped proposals. And the general principle that guiding the re-scoring process is that, the IoU between the two overlapped proposals should be act as a weight for the 'score-decay' of the non-maximal one. Either the linear function or Gaussian function could be utilized to execute the re-scoring process depends on their performance regarding to the dataset.

Since for both of the linear function based Soft-NMS and Gaussian function based Soft-NMS, the

Table 3.7: $\text{AP}_{0.5}$ of Pairwise-NMS vs GreedyNMS on PETS

Method	Pairwise-NMS	GreedyNMS			
AP @ 0.5	**78.1**	N_t >0.0	N_t >0.1	N_t >0.2	N_t >0.3
		55.0	66.0	71.4	75.0
		N_t >0.4	N_t >0.5	N_t >0.6	
		76.6	73.4	64.8	

Table 3.8: Comparison results with Soft-NMS on PETS

methods	thre	tp	fp	dt	gt	rec	prec	AP@0.5
Soft-NMS_L	0.0	10119	606298	616417	10292	98.32	1.64	37.60
Soft-NMS_L	0.1	8402	11992	20394	10292	81.64	41.20	60.75
Soft-NMS_L	0.2	7601	2942	10543	10292	73.85	72.10	**63.27**
Soft-NMS_L	0.3	6645	540	7185	10292	64.56	92.48	61.94
Soft-NMS_L	0.4	5494	192	5686	10292	53.38	96.62	52.40
Soft-NMS_G	0.1	7216	1383	8599	10292	70.11	83.92	68.05
Soft-NMS_G	0.2	7543	2100	9643	10292	73.29	78.22	**70.24**
Soft-NMS_G	0.3	7756	3115	10871	10292	75.36	71.35	69.00
Soft-NMS_G	0.4	7914	4561	12475	10292	76.89	63.44	64.73
Soft-NMS_G	0.5	8012	6014	14026	10292	77.85	57.12	59.84
Pairwise-NMS	0.7	9477	66718	76195	10292	92.08	12.44	**78.11**

scores of the non-maximal proposals are continuous penalized, as a result, there will be no proposal removed from each suppression round, which will cause a long excution time compared with GreedyNMS, thus there is a parameter θ used in practical. The role of θ is to filter out the proposals which have smaller scores regardless what their originals scores are. By this way, Soft-NMS can keep the same level of efficiency with GreedyNMS. For Gaussian function based Soft-NMS, there is an extra parameter σ which is the standard deviation of the Gaussian function, and can be viewed as a 'controller' of the distribution shape of the penalty function.

In Table 3.8, on PETS dataset, we list the detection results of Soft-NMS under different settings and that of Pairwise-NMS. Specifically, with Soft-NMS_L denotes linear function based Soft-NMS, and with Soft-NMS_G denotes Gaussian function based Soft-NMS. For easy comparison, we unified use the 'thre' in the second column of Table 3.8 to denote the parameter θ, σ and N_d employed in linear function based Soft-NMS, Gaussian function based Soft-NMS and Pairwise-NMS, respectively. In addition, as point out before, for Gaussian function based Soft-NMS, both the parameters θ and σ are used, after fine tuning the θ on linear function based Soft-NMS, we fix it as 0.2, and just take σ as the variable. In order to have a comprehensive comparison and understanding about different algorithms under various settings, some key measurements are listed, they are true positives, false positives, number of final detections, number of ground-truth bounding boxes, recall, precision and average precision with evaluation threshold 0.5 (tp, fp, dt, gt, rec, prec, AP@0.5) on the first row.

From observing the AP performance of different algorithms, it can be concluded that, both the

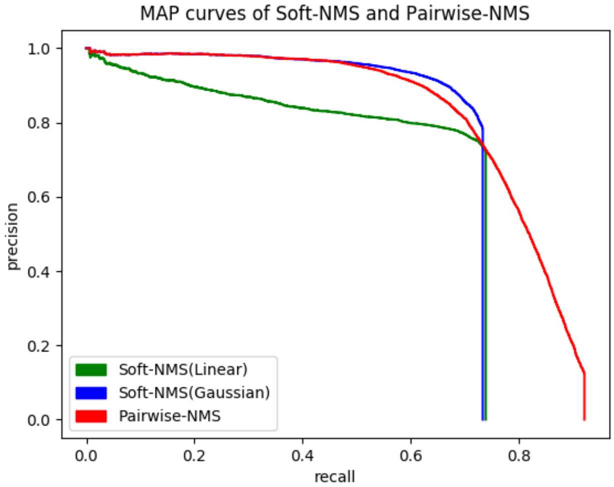


Figure 3.8: MAP curves of Soft-NMS and Pairwise-NMS. As can be been, both of the liner function based Soft-NMS and Gaussian function based Soft-NMS have lower performance than that of Pairwise-NMS on PETS dataset regarding to detection accuracy. Form surface, it is struggle for both type of Soft-NMS to reach a good balance between reall and precision on PETS dataset. From essential, we suspect that Pairwise-NMS has superior ablity than Soft-NMS to handle the very crowded cases.

linear function based Soft-NMS and the Gaussian function based Soft-NMS have lower performance than our method on PETS dataset. Specifically, Soft-NMS_L achieves highest AP of 63.27 percent, and Soft-NMS_G achieves highest AP of 70.24. Both of them are far behind from the performance of our proposed Pairwise-NMS. And from watching the recall and precision, it is much clear that, for Soft-NMS, it almost fails in all settings to reach a good trade-off between the precision and recall, which can be viewed as the surface explanation of the unsatisfied APs. And from the essential, we suspect there are two reasons: firstly, from the perspective of dataset, both of the VOC2007 and COCO dataset used by Soft-NMS is relative sparse and contains very few crowed cases, on the contrary, PETS is very crowded and is much challenging. For example, a lot of frames contain above 30 pedestrians, and they are quite small in resolutions and heavily occluded with each other. Secondly, from the perspective of detector, in order to keep the same settings with Hosang et al. [2016, 2017]; Tang et al. [2013], detection proposals generated by DPM (before NMS) are used here. The detection proposals for each image easily reach several thousands. Which may cause more difficulties for Soft-NMS compared to that at most 300 detection proposals are used in the original paper of Soft-NMSBodla et al. [2017]

In Figure 3.8, the MAP curves of best Soft-NMS_L, best Soft-NMS_G and Pairwise-NMS are given for the better illustration and easier comparison in visual.

In conclusion, under the extremely crowded scenes, our proposed method can achieve a better balance between the recall and precision and is much robust than Soft-NMS.

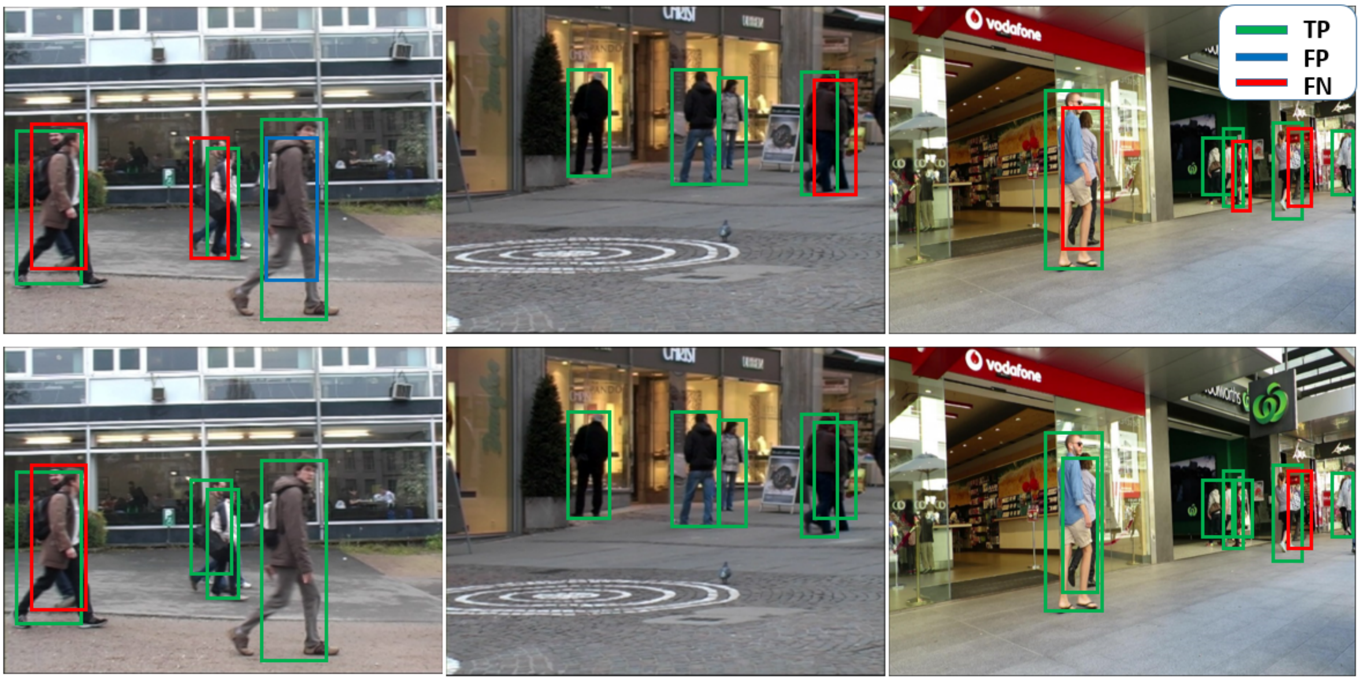


Figure 3.9: Visualized detections of GreedyNMS (the first row) and Pairwise-NMS (the second row). With green box, blue box, and red box to denote true positive, false positive and false negative, respectively. As can be been, compared with the detection results acquired from GreedyNMS, the detection results acquired from Pairwise-NMS dramatically reduce the false negatives (which leads to the higher recall), and meanwhile reduce the false positives slightly (which contributes to the precision).

3.4.6 Qualitative Results

In Figure 3.9, several visualized examples from MOT2015 dataset are given, the base detector is Faster-RCNN. The first row are the detection results we obtained using GreedyNMS, followed by the second row with results acquired by Pairwise-NMS. As can be seen from the legend at the top right, detections with blue-edge boxes are true positives (TP), detections with green-edge boxes are false positives (FP), and the detections with red-edge boxes are false negatives (FN).

In all of the three examples, due to the heavily overlapping between two or more person, GreedyNMS and Pairwise-NMS all failed in some cases and missing detect some targets (as denoted with the red boxes), however, compared with the results acquired from GreedyNMS, the detection obtained from Pairwise-NMS has less missing detections, which strongly prove the effectiveness of the proposed pairwise-relationship network for handling heavily occluded case. In addition, in the first image, as denoted by the blue box, Pairwise-NMS performs better in reducing false positives. To sum up, comparing to GreedyNMS, Pairwise-NMS effectively preventing the case of missing detections and discarding more false positives.

3.4.7 Ablation Study

Sampling Policy

Due to the imbalance between similar pairs and dissimilar pairs, it is important to make a wise policy to achieve good accuracy and to ease the learning process of pairwise-relationship network.

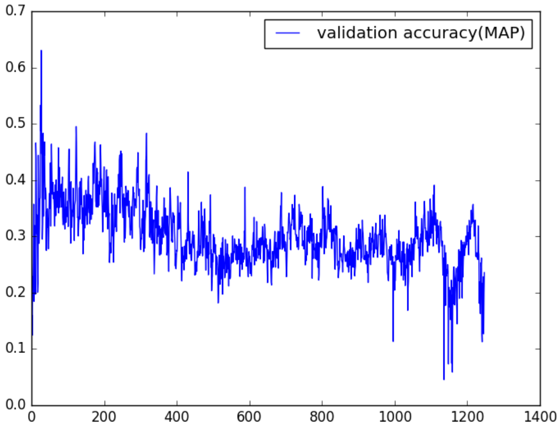


Figure 3.10: Validation accuracy of pairwise-relationship network using FC layer.

After a thorough investigation of datasets which contain heavy occlusions including TUD-Crossing and MOT15, we find the number of heavily occluded objects is relatively small compared with the total number of ground-truth bounding boxes. According to the analysis in Section 3.3.4, the proposal pairs that contain two different objects are defined as dissimilar pairs, which should not be merged. All other cases are defined as similar pairs and should be merged. Based on our statistics on TUD-Crossing as can been seen in Table 3.9, the ratio between dissimilar and similar pairs are more than 10. For the sake that the network can learn to recognize two objects, we randomly reduce the number of similar samples during training. Specifically, the ratio between dissimilar and similar pairs is kept with 1:3.

Meanwhile, among the similar pairs, considering how many objects (zero or one) the two proposals contain, a natural way of sampling pairs is to form the similar pairs based on their ratios of different cases. Because correctly predicting the similar pairs will dramatically reduce the false positives and improve the precision, the sampling ratios will also affect network's ability to differentiate the dissimilar and the similar pairs.

Different Network Structures

In the experiments, we discover that, for the last layer, pairwise-relationship network with GAP layer has better performance than the one with FC layer. It may due to the reason that FC layer usually carries the position information, which in turn causes confusion to the network to distinguish the case5 from case6 in Table 3.1. Specifically, it is difficult for the pairwise-relationship network with FC layer to differentiate the case that two proposals contain one common object and the case that two proposals contain two different objects.

In order to test the affects between using the global average pooling (GAP) layer and fully connected (FC) layer in the pairwise-relationship network. We use 80% images of TUD-Crossing dataset to train the pairwise-relationship network with GAP layer and the pairwise-relationship network with FC layer

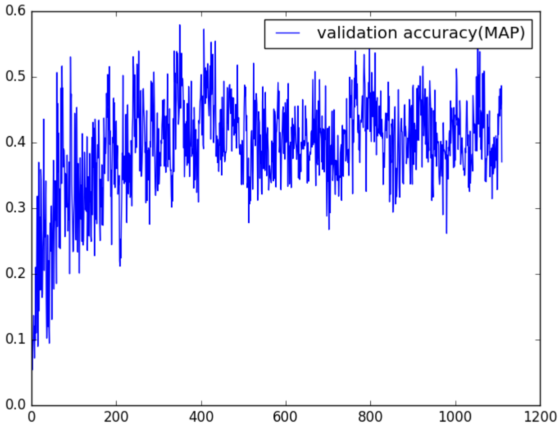


Figure 3.11: Validation accuracy of pairwise-relationship network using GAP layer

Table 3.9: Statistics of pairs on TUD-Crossing

proposals	IoU threshold	gt pairs	0 object pairs	1 object pairs	2 object pairs
300	0	537	980596	161320	7596
300	0.1	398	699854	133364	3061
300	0.2	254	533469	74523	1582
300	0.3	150	391318	16309	930

from scratch, and use another 20% images of TUD-Crossing as the validation set. The validation loss curves of the networks with GAP layer and FC layer are as shown in Fig 3.11 and Fig 3.10, respectively. In the two figures, X axis represents the number of training iteration, and Y axis is MAP (AP@0.5). As can be seen, the pairwise-relationship network with GAP layer has better performance and continue learning as the training goes. Which confirm our conjecture in the section of ablation Study, and for all of the experiments in this paper, the network with GAP layer is employed.

3.5 Conclusions

We propose a Pairwise-NMS as the post-processing step for object detection. Feeding two overlapping proposals to a pairwise-relationship network can smartly tell the Pairwise-NMS if there are two objects or one/zero object contained. It can cure the drawback of GreedyNMS for “suppressing all” under heavy occlusions, and is more robust than Soft-NMS in crowded scenes. Moreover, Pairwise-NMS can consistently improve the performance and neatly couple with GreedyNMS. And can be as an ingredient to be integrated into learning-based detectors including Faster-RNN, DPM without losing the efficiency. We believe that our work is beneficial to instance segmentation and multi-object tracking in crowded scenes. In the future, there will be two research problems of our interests. One of the problems is to integrate Pairwise-NMS into learning based detectors such as Faster-RCNN, YOLO, SSD to perform joint training and inference, and the other is to explore a more general rule to handle a cluster of multiple objects at the same time.

Bibliography

M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

A. Arnab and P. H. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017.

O. Barinova, V. Lempitsky, and P. Kholi. On detection of multiple object instances using hough transforms. *TPAMI*, 34(9):1773–1784, 2012.

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS– improving object detection with one line of code. In *ICCV*, 2017.

A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserv- ing crowd monitoring: Counting people without people mod- els or tracking. In *CVPR*, 2008.

A. Ellis and J. M. Ferryman. PETS2010 and PETS2009 evaluation of results using individual ground truthed single views. In *AVSS*, 2010.

M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

P. Felzenszwal, R. Girshick, and D. McAllester. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

R. Girshick. Fast R-CNN. In *ICCV*, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2012.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

J. Hosang, R. Benenson, and B. Schiele. A convnet for non-maximum suppression. In *GCPR*, 2016.

J. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. In *CVPR*, 2017.

S. Ioffe and C. Szegedy. Batch Normalization: accelerating deep network training by reducing internal covariate shift. In *arXiv preprint arXiv:1502.03167*, 2015.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

L. Leal-Taixe, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: towards a benchmark for multi-target tracking. In *arXiv preprint arXiv:1504.01942*, 2015.

B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

M. Lin, Q. Chen, and S. Yan. Network in network. In *arXiv preprint arXiv:1312.4400*, 2013.

T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar. Microsoft COCO: common objects in context. In *ECCV*, 2014.

T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. In *CVPR*, 2013.

T. T. Pham, S. H. Rezatofighi, I. Reid, and T.-J. Chin. Efficient point process inference for large-scale object detection. In *CVPR*, 2016.

H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

S. H. Rezatofighi, V. Kuma, A. Milan, E. Abbasnejad, A. Dick, and I. Reid. DeepSetNet: predicting sets with deep neural networks. In *ICCV*, 2017.

M. Rodriguez, I. Laptev, and J. Sivic. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

R. Rothe, M. Guillaumin, and L. V. Gool. Non-maximum suppression for object detection by passing messages between windows. In *ACCV*, 2014.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *ICCV*, 2013.

M. B. R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: global multi-object tracking using generalized minimum clique graphs. *ECCV*, 2012.

Chapter 4

Exemplar based Domain Adaptation for Semantic Segmentation via Convolutional Adaptors

This chapter targets at solving the domain adaption for semantic segmentation. Given the synthetic data (i.e. GTA5) with annotations for each frame, the goal is to achieve better segmentation performance on real data (i.e. Cityscapes), which has no annotations provided. To this end, we deliver three components. (1) Single and multi-exemplars are employed for each class in order to cluster the per-pixel features in the embedding space. For which, exemplars belong to the same class are expected to represent the distribution of the samples of that class, meanwhile, exemplars belong to different classes are expected to be separable, that is has a relative larger distance compared to that of exemplars belong to the same class, in the embedding space. Besides exemplar-based cross-entropy loss, clustering loss is delivered for further distinguishing the features which belong to different classes. (2) Class-balanced self-training strategy is utilized for generating pseudo labels in the target domain, and the confident pseudo labels will be used in the training loop in order to adapt the appearance information in target domain based on the trained model in the source domain. Which will iteratively update the segmentation model as well as the generated pseudo labels. (3) Moreover, in order to make a better alignment between the source domain and the target domain, convolutional adaptors are adopt in our network, which enforce the low-level and mid-level features are shared from the source and the target domains, and can further enforce the features in the source domain and the target domain are closed with each other. Extensive experiments has demonstrated the effectiveness of the proposed method.

4.1 Introduction

Semantic segmentation plays an important role for image understanding tasks, such as auto-driving Siam et al. [2018]; Treml et al. [2016], robot navigation Mousavian et al. [2019], 3D scene analysis Leibe et al. [2007]; Li et al. [2019a] and medical image analysis Ronneberger et al. [2015]. Fully-supervised segmentation has reach the point of matching human performance which mainly due to the heavy-burden network design and the tremendous annotated data, which bring human-being the promising expectation for auto-driving, meanwhile these models are usually data-driving and have strong bias towards the datasets they are trained on.

Since the real-world scenarios various a lot including the illumination, road structure, pose etc, it therefore require an unified network structure which can quickly adapt to the new scenario when the driverless car arrives. It means that the network developed is supposed to have the strong ability to quickly adapt and get well segmentation for the unseen objects and scenarios. However, annotating dense labels for new images are time-consuming and labor-consuming, and it is not practical to annotate new objects each time when come to a new scenario. Meanwhile, the existing fully supervised models which fine-tuning on benchmarks such COCO Lin et al. [2014], PASCAL VOC Everingham et al. [2010] can be reused on the new tasks, because although the appearance information varies from scene to scene, the low-level and the mid-level features can be shared among different domains.

One bunch of the methods which has boosted up the performance of this task employ the adversarial learning Busto et al. [2018]; Chen et al. [2017b]; Grandvalet and Bengio [2005]; Li et al. [2019b]; Treml et al. [2016]; Zou et al. [2018], but they have some limitations. Firstly, they assume that for both of the source domain and target domain, there is a data distribution center for each class, which may not be Busto et al. [2018], Chen et al. [2017b], French et al. [2017], Grandvalet and Bengio [2005] true. Based on this observation, we propose to use the exemplars, a clustering way, to replace the commonly used fully-connected (FC) classifier, we expecting that the multiple exemplars together can simulate the data distribution of the class in the embedding space, and proves that multi-exemplar based classifier achieves better performance that that of the single-exemplar based classifier through experiments. Secondly, although the source domain and the target domain share some structural information because they contain the same set of classes, but the appearance various a lot which lead to the performance drop dramatically when conducting inference on the target set with the pretrained model on the source data only. To this end, we employ the class-balanced self-training strategy introduced by CBST Zou et al. [2018]. Specifically, self-training conduct the joint optimization about the segmentation model and generated pseudo labels in the target domain. CBST conduct a further step which is inter-class balanced to avoid the generated labels are overwhelmed by the easy-to-classified classes.

Considering that even after using the self-training strategy, the amount of the generated pseudo labels are still very small compared to that of annotations in the source domain. And the generated pseudo labels are not totally reliable based on their predicted segmentation confidence. Inspired by the multi-domain learning work of Rebuffi et al. [2017], we design a similar structure also named "convolutional adapter" for UDA task based on VGG Simonyan and Zisserman [2014] backbone. Which drives from the original convolution layers via parallel and serial connections, and can make a better alignment between the features in the source domain and the target domain.

To sum up, our contributions are as follows:

- Using the multi-exemplars to replace the commonly used FC classifier for better representing the data distribution of each class, and combined with the proposed clustering loss to make the exemplars which belong to different classes are separable.
- Taking use of the class-balanced self-training strategy, which will generate pseudo labels in the target domain, and incorporate the appearance information into the training loop. Iteratively optimize the segmentation model and the generated pseudo labels.

- With the help of the designed convolutional adaptors, features from different domains can be better aligned. Therefore, achieve better segmentation performance in real data.

4.2 Related work

Semantic Segmentation and UDA The performance of fully supervised semantic segmentation has long been boosted up since the invention of the work by long et al Long et al. [2015], which include some well-known works such as RefineNet Lin et al. [2017], DeepLab Chen et al. [2017a] and its inherits. The overall trend is using more labeled data and much heavy networks to push the extreme performance to a limit. Unsupervised domain adaptation (UDA) is somehow a new problem which attract more and more attentions recently for the developing of driverless car and mobile robots. Since in real scenarios, the ubiquitous environment changes due to the illumination, pose, sensor various. It is impossible to develop an unified network which is capable of handling all of the new scenarios, therefore, how to take use of the avaliable annotated labels to quickly adapt to the new environment become a hot topic in the computer vision community.

Style Transfer Based Methods Style transfer learning methods assume the semantic/structural context/meaning of two domains are similar/same, and try to transfer the feature of appearance from the source to the target. The representatives are CycleGAN Zhu et al. [2017] and UNIT Liu et al. [2017]. Although it can incorporate the appearance information of the target domain, one limitation that style transfer based methods suffers is the “unreal” effects of tranfered style, which make these kind of methods still have gaps from the real data.

Adversarial Learning Based Methods Adversarial learning via using discriminator and generator, to enforce the features from two domains are closed with each other. Which can be done in pixel/feature/prediction Busto et al. [2018]; Chen et al. [2017b]; French et al. [2017]; Grandvalet and Bengio [2005] levels. By taking use of circle consistency loss, CycleGAN Zhu et al. [2017] tries to find the good matches between similar features. BDL Li et al. [2019b] employs adversarial loss and reconstruction loss to conduct domain-transfer based semantic segmentation, since it achieves the state-of-the-art performance in UDA task, we choose it as our baseline. However, BDL Li et al. [2019b] suffers the problem that the network structure and loss term are too complex and is not elegant at all.

Semi-supervised Learning Based Methods Self-training strategy seek to generate pseudo labels in the target domain based on the fully-supervised model trained in the source domain, which can effectively boost the performance of the UDA task. One limitation of the previous self-training based methods is that it is prone be overwhelmed by the easy classified pixels which has a relatively high confidence score during the process of generating pseudo labels, while ignore the hard-to-classify classes. CBST Zou et al. [2018] proposes to use a class-balanced self-training strategy to generate pseudo labels in the target domain based on the self-training strategy, which can effectively boost the performance of the UDA task, since it can generate some confident labels in the target domain, which is helpful for the UDA task to capture the appearance information from the target domain. However, self-training from essential can be viewed as a joint learning problem to optimize the model as well as the pseudo labels in the target domain, then it can not ensure the generated pseudo labels are accurate as expected. That means, as the training process undergoing, more and more generated pseudo labels will be added into the training annotations, the remaining labels will have much less confidence regarding to the prediction accuracy.

Multi-domain Learning Based Methods Inspired by the work of Rebuffi et al. [2017], we apply the multi-domain learning strategy in the UDA task. Being different from the transfer learning and multi-task learning, multi-domain learning aims to achieve feature sharing among domains, with the goal to quickly adapt to the new task while still retain the good performance in the old task. Specifically, assume the learning function from input X, can be represented by $F(X) = \alpha * F * x$, where F store most of the mid-level and low-level features learned from different domains, which are class-agnostic. α is the class-determined weight, which only take a small ratio of the overall parameters, while can quickly adapt to the target task. Here, we follow the philosophy of residual adaptor which is delivered by Rebuffi et al. [2017], that are serial adaptor and parallel adapter, we adapt these two kind of convolutional adapters based on VGG Simonyan and Zisserman [2014] backbone in our task to better align the features from source domain to the target domain.

4.3 Methodology

4.3.1 Overview

In this paper, targeting at handling unsupervised domain adaptation of semantic segmentation from the synthetic data to the real data, we deliver a novel network which is composed of three components. That are multi-exemplar based classifier, class-balanced self-training strategy and the convolutional adapters (both of the serial adapter and the parallel adapter). In particular, we first employ a multi-exemplar based classifier to replace the commonly used FC classifier. The assumption is based on that via using multi-exemplars, the distribution center of one class can be better represented compared to the previous methods, which usually assume that there is one distribution center for each class. Based on the proposed exemplars-based classifier, a clustering loss is also delivered to enforce the exemplars belong to different classes are separable from each other. Secondly, class-balanced self-training strategy is utilized to generate the pseudo labels in the target domain, which can boost the segmentation performance because it incorporates the appearance information from the target domain and interatively optimize the segmentation in training loop via using the new generated pseudo labels. Thirdly, both of the serial adapter and parallel adapter are proposed to better align the features from the source domain to the target domain. The overall workflow of the proposed method is as shown in Algorithm 5, and more details will be explained in the later sections.

Algorithm 2: Workflow of the Algorithm

```
1. Iterate all of the batches from the source domain;
2. Train with $L_{CE} + L_{Cluster}$ on the network $f$ ;
3. Iterate all of the batches from the target domain ;
4. Evaluate current confidence probability $P_c$ in the target domain, generate pseudo labels if
   $P_c > Conf\_Thre$;
5. Train with $L_{CE} + L_{Cluster}$ on network with the adapter $f + a$, on both of the pixels in the
   source domain and the pixels with pseudo label in the target domain;
```

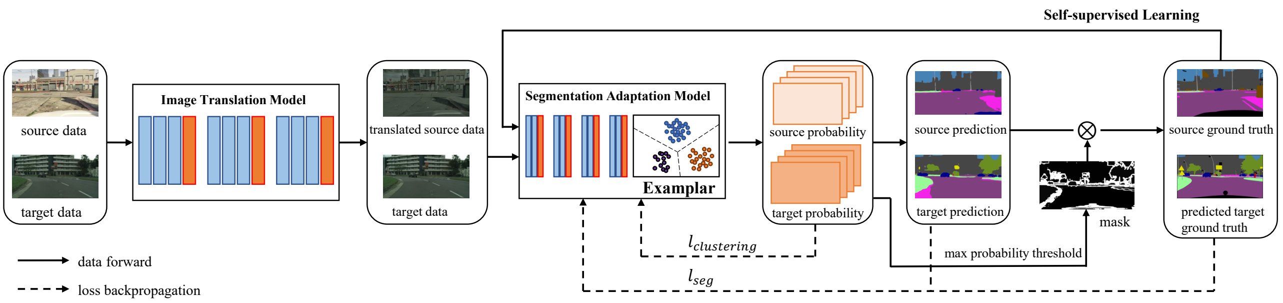


Figure 4.1: Work flow and loss function of the proposed method.

4.3.2 Network Structure

In Figure 4.1 is the workflow of the proposed method. From the left to the right, two images from the source domain and the target domain are first sent to a style transfer network (CicycleGAN Zhu et al. [2017] here), among them, target image keeps the same, while the source image maintain the same content but with transferred appearance like the target image. Then the transferred images are fed into a VGG network, after getting the features of the last convolutional layer, a multi-exemplar based classifier are designed, which is composed of the cross-entropy loss L_{CE} and the clustering loss $L_{Cluster}$. Since in the target dataset, there is no annotated labels avaliable at the very beginning, therefore class-balanced self-training is employed to generate pseudo labels in the target domain. After that, serial adapter and parallel adapter are applied to better align the features from the source domain to the target domain.

4.3.3 Exemplar-based Classifier

In this section, we mainly introduce our first component, that is the exemplar-based classifier. In particular, the mathematical expression of the cross entropy loss and the clustering loss are introduced first, and which is followed by the specific explanation for the single-examplar and the multi-exemplar based classifiers.

- We define network as f, and network with adaptor as $f + a$.
- Each class has k exemplars, the probability of assigning one sample z to one exemplar z_k^c is:

$$P_{ck} = \frac{-\exp \|z - z_k^c\|}{\sum_{k'} \sum_{c'} \left(-\exp \|z - z_{k'}^{c'}\|\right)} \tag{4.1}$$

The probability of one sample belonging to class c is:

$$P_c = \sum_{k=1}^{k} P_{ck} \tag{4.2}$$

- loss term

– Classify Loss (Cross Entropy Loss)
  Cross Entropy (CE) loss is defined as follows, and can be applied in two cases:
  (1) Apply to labeled source domain images.
  (2) Apply to target domain samples with pseudo labels.

$$\mathcal{L}_{CE} = CE(P_c, y)$$

– Clustering Loss
  Encouraging sample embeded z close to at lease one examplar z_k^c

$$\mathcal{L}_{cluster} = -\sum_{ck} P_{ck} \cdot \log P_{ck} + \lambda \sum_{ck} P_{ck} \cdot ||z - z_k^c||_2^2$$

The clustering loss include two items, and they are defined as above and have the meaning as follows:

(1) The left term is entropy, encourage z is closer to one exampar than the other, and can be understand as the relative distance.

(2) The right term is the absolute distance: encourage the distance between the sample z and the chosen exemplar z_k^c is small.

(3) Apply to all of samples in source and target domain.

Single exemplar vs Multiple exemplar In our design and experiments, because the number of exemplars k within each class is flexible, we deliver both of the single-examplar and multi-examplar based classifiers for the UDA task. According to our understanding, single-examplar based classifier is much more like the adversarial learning methods, which assume that for both of the source domain and the target domain, there is one distribution center, and the loss as well as the network structure are trying to enforce the two centers are close to each other. Being different, the multi-exemplar based classifier is much more like some other clustering methods such as K-means and gaussian mixture model(GMM), which try to use several centers in the embedding space to simulate the sample distribution of a specific class. Our experiments prove that using multi-examplars achieves better performance compared to that of using single-examplar for each class.

Normalization Due to the initialization for the exemplars, at the very beginning of training process, the value magnitude have a big difference between the pixel-wise feature and the exemplars, in order to better construct the exemplar-based classifier, we conduct a normalization for the embedding features and the exemplars.

$$P_{ck} = \frac{-\exp\left(\|z - z_k^c\|_2^2 * \beta\right)}{\sum_{k'} \sum_{c'} -\exp\left(\left\|z - z_{k'}^{c'}\right\|_2^2 * \beta\right)} \tag{4.3}$$

This formula is implemented by the torch.norm with norm magnitude equal to 1.0. In order to prevent the numerical problem, since the norm item as the denominator, a very small number is added 1e-16 when conducting the normalization. After the normalization results are acquired for the embedding features and the exemplars, L2_norm pairwise distance is calculated to construct the exemplar-based classifier in our task. We experiment with $\beta = 5$ and $\beta = 10$, and it proves no big difference according to the preliminary experimental results, for all of our experiments, we use $\beta = 5$ in our experiments. There is also comparison between using L1_norm distance and L2_norm distance in the ablation study.

4.3.4 Class Balanced Self-training

Self-training Zou et al. [2018] is widely used in UDA task for generating the pseudo labels in the target domain. Through this way, the appearance information from the target domain can be captured. And it can actually be viewed as a joint optimization problem for the segmentation model and the generated pseudo labels in the target domain. It will gradually generate pseudo labels from easy-to-learn samples to the hard-to-learn samples, with much less confidence. The following formulas about self-training defined same as in Zou et al. [2018]

Self-Training with self-paced learning

$$\begin{aligned}\min_{\mathbf{w},\hat{\mathbf{y}}} \mathcal{L}_{ST}\left(\mathbf{w},\hat{\mathbf{y}}\right) = & -\sum_{s=1}^{S}\sum_{n=1}^{N}\mathbf{y}_{s,n}^{\top}\log\left(\mathbf{p}_n\left(\mathbf{w},\mathbf{I}_s\right)\right) \\ & -\sum_{t=1}^{T}\sum_{n=1}^{N}\left[\hat{\mathbf{y}}_{t,n}^{\top}\log\left(\mathbf{p}_n\left(\mathbf{w},\mathbf{I}_s\right)\right) + k|\hat{\mathbf{y}}_{t,n}|_1\right] \\ & s.t. \quad \hat{\mathbf{y}}_{t,n} \in \left\{\left\{\mathbf{e}^{(i)}|\mathbf{e}^{(i)} \in \mathbb{R}^C\right\} \cup \mathbf{0}\right\}, \forall t,n \\ & \quad\quad k > 0\end{aligned} \tag{4.4}$$

Class-balanced Self-Training

As mentioned before, self-training has the risk that the easy-to-learn samples will dominant the generated samples with pseudo labels. Which will lead to that there is no annotations for the hard-to-learn samples. Based on this observation, Zou et al. [2018] propose a class-balanced self-training strategy to generate the pseudo labels, in order to make sure for each class, the generated labels only compete with other candidates based on their confidence within the same class. And the expression is as follows.

$$\begin{aligned}& \min_{\hat{\mathbf{y}}} -\sum_{t=1}^{T}\sum_{n=1}^{N}\left[\sum_{c=1}^{C}\hat{\mathbf{y}}_{t,n}^{(c)}\log\left(\mathbf{p}_n\left(c|\mathbf{w},\mathbf{I}_s\right)\right) + k|\hat{\mathbf{y}}_{t,n}|_1\right] \\ & s.t. \quad \hat{\mathbf{y}}_{t,n} = \left[\hat{\mathbf{y}}_{t,n}^{(1)},\cdots,\hat{\mathbf{y}}_{t,n}^{(C)}\right] \in \left\{\left\{\mathbf{e}^{(i)}|\mathbf{e}^{(i)} \in \mathbb{R}^C\right\} \cup \mathbf{0}\right\}, \forall t,n \\ & \quad\quad k > 0\end{aligned} \tag{4.5}$$

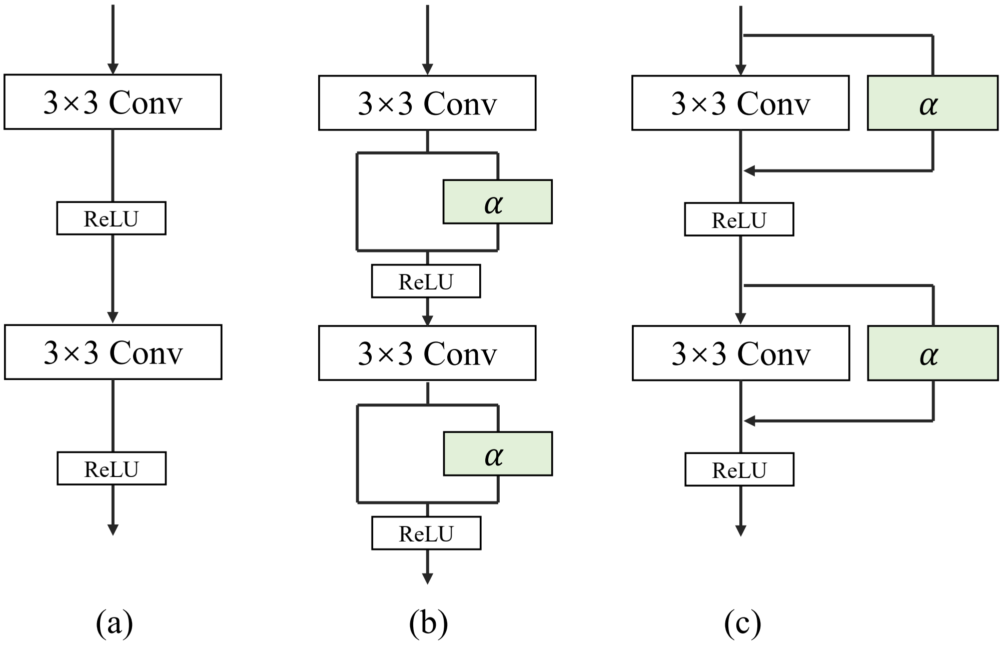


Figure 4.2: Convolutional adapters: (a) is an original block with two convolutions. (b) and (c) are modified versions with serial and parallel adapter, respectively

4.3.5 Domain Adaptation with Convolutional Adapter

As mentioned before, current methods for UDA may have a limitation that even though the network structure and loss terms are designed to enforce the embedding features are closed with each other, they may still have a gap which block the performance of UDA. In order to overcome this limitation, inspired by the work of Rebuffi et al. [2017], we deliver the convolutional adapters to conduct a better domain adaptation from the source domain to the target domain. That is, achieving a better alignment for the features from the source domain and the target domain. Specifically, based on the backbone VGG Simonyan and Zisserman [2014] network structure. Serial adapter and parallel adapter are added to achieve the better feature alignment.

Original Convolution

Figure. 4.2 (a) shows the plain convolution module used in the VGG structure. Each convolution with kernel size 3×3 is followed by an active function ReLU layer. We denote this operation by $\mathcal{F}(\cdot)$. This process can be formulated as follows:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) \tag{4.6}$$

Where $\mathbf{x}$ represents the input feature map and $\mathbf{y}$ is the obtained output feature map. In each stage of VGG, two or three convolution modules are stacked.

Serial Adapter

After the convolution, we add a bypass convolution to learn the adaptation. In Figure. 4.2 (b), the bypass is denoted as α. The result is added element-wisely to the output of the original convolution to achieve the purpose of adapting to different domain. This serial adapter with single convolution can be

Meth	road	side	buil	wall	fenc	pole	ligh	sign	vege	terr	sky	pers	ride	car	truc	bus	trai	moto	bicy	m13	m16	m19
ST	83.8	17.4	72.1	14.3	2.9	16.5	16.0	6.8	81.4	24.2	47.2	40.7	7.6	71.7	10.2	7.6	0.5	11.1	0.9	28.1	-	-
BDL-Adv	78.27	27.08	71.12	13.38	8.24	16.47	16.18	10.73	78.13	30.25	70.41	37.04	1.28	67.92	6.42	3.68	0.0	4.79	0.43	28.52	31.57	35.93
Single-Exam	79.73	26.8	71.54	12.54	9.88	17.17	15.91	9.42	78.47	30.86	72.79	38.6	0.82	67.85	5.71	3.44	0.0	4.91	0.78	28.85	31.92	36.24
Multi-Exam	78.14	26.22	71.98	12.73	10.23	16.5	18.56	11.63	78.15	31.16	68.07	41.42	0.96	68.17	6.77	3.94	0.0	5.43	0.35	28.97	32.03	36.39
Multi-SSL	78.0	26.42	71.65	14.0	9.79	17.96	17.79	10.92	78.57	31.61	70.77	41.15	1.33	67.65	6.53	3.8	0.0	5.99	0.86	29.2	32.29	36.53
Exam-SSL-SAdap	77.25	27.87	71.56	13.92	9.46	18.04	18.74	12.79	78.77	32.42	70.83	40.97	1.36	67.8	7.12	4.43	0.0	6.14	1.42	29.52	32.58	36.92
Exam-SSL-PAdap	72.54	24.87	72.23	13.59	10.69	18.76	16.45	12.95	78.09	32.28	73.52	41.25	1.78	70.34	7.2	6.46	0.08	7.27	1.72	29.58	32.66	36.88

Table 4.1: GTA5 to Cityscapes with VGG backbone.

formulated as follows:

$$\mathbf{y} = \mathcal{F}\left(\mathbf{x}\right) + \mathcal{F}_{\alpha}\left(\mathcal{F}\left(\mathbf{x}\right)\right) \tag{4.7}$$

Where $\mathcal{F}_{\alpha}\left(\cdot\right)$ represents the bypass convolution in the serial adapter.

Parallel Adaptor

Parallel adaptor operates the bypass convolution in parallel with the main stream. The bypass convolution uses the same input as main stream convolution to learn the adaptation parameters. This parallel adapter with single convolution can be formulated as follows:

$$\mathbf{y} = \mathcal{F}\left(\mathbf{x}\right) + \mathcal{F}_{\alpha}\left(\mathbf{x}\right) \tag{4.8}$$

When this bypass convolution performs identity mapping, i.e. $\mathcal{F}_{\alpha}\left(\mathbf{x}\right) = \mathbf{x}$, the parallel adapter constitutes a residual module.

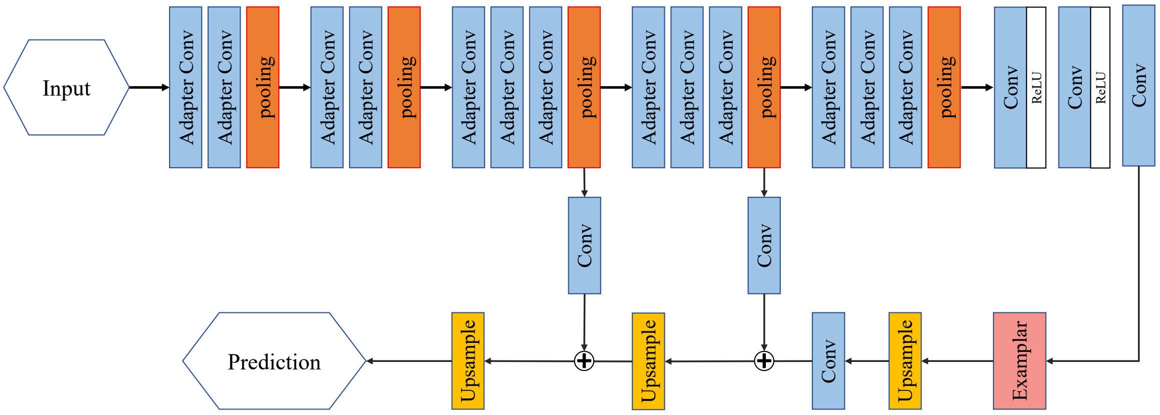


Figure 4.3: Network structure of the proposed method.

4.4 Experiments

In this section, extensive experiments are provided. Following the same dataset settings of baseline methods such as BDL Li et al. [2019b] and CBST Zou et al. [2018]. The synthetic data we employed is GTA5, and the real dataset is Cityscapes Cordts et al. [2016]. Both of the synthetic data and real data contain the same 19 classes. As mentioned before, for the synthetic data, the per-pixel dense annotation is provided for each image, and for the real data, which has no annotation at all, during training, the color image of Cityscapes training set are accessible. All the performance metric are acquired from Cityscapes validation data, which contain 500 images.

4.4.1 Training details

For a fair comparison, all other settings are keep consistent with the baseline network BDL Li et al. [2019b] except the *examplar* and the *class-balanced self-training* parts. That is, for VGG backbone, the learning rate is 1e-5, the training iterations are 80k, with the momentum 0.9 and weight_decay is 5e-4.

4.4.2 Qualitive Results

As shown in Table 4.1, the qualitative results (J Mean) of CBST Zou et al. [2018], BDL Li et al. [2019b], and the proposed method with different components are listed out. Specifically, which include single-exemplar based classifier, multi-exemplar based classifier, multi-examplar based classifier with self-training, multi-examplar based classifier with self-training and the convolutional adapter. From overall, we can conclude that our method outperforms the baseline methods, meanwhile, each component of the proposed method are effective to boost the segmentation performance.

4.5 Ablation Study

In this section, in order to prove the effectiveness of the designed components proposed in this method, we conduct different ablation studies on three adds-on, that are exemplars, self-training and the convolutional adapters.

4.5.1 Abalation Study of Different Examplar Initialization

Init Way	mIoU19	mIoU16	mIoU13
Xavier	28.33	31.16	35.40
Zero	28.89	31.88	36.25
Uniform	28.21	31.08	35.37
Random	28.72	31.7	35.99

Table 4.2: Different Initializations for 256-dim examplar-classifier.

Since our method is somehow a clustering method, which means different initializations can make a big difference for the final results. Specifically, we take use of different ways to do the initializations, include *Xavier*, *Zero*, *Uniform*, and *Random.* Through the experiments, it shows that the result acquired from Zero init is a bit better than other ways. Therefore, we employ Zero Init in all of our experiments if not specifically specified.

4.5.2 Ablation Study of Examplar Dimensions

Dim	mIoU19	mIoU16	mIoU13	Speed	Mem
256	28.89	31.88	36.25	0.16s	2844M
512	28.99	32.01	36.33	0.17s	3480M
1024	28.59	31.61	35.89	0.18	3906M
2048	28.6	31.55	35.77	0.20	5473M
4096	29.18	32.24	36.5	0.22	9766M

Table 4.3: Results of examplar-classifier with different dimentions

We also conduct experiments with different dimensions of exemplars. Which include the exemplar dimension of 256, 512, 1024, 2048 and 4096. We list out the feature dimension as well as their accuracy of mIoU19, mIoU16, mIoU13, speed and memory costing per frame. And as can be seen, the much higher dimension the exemplar is, the more accurate performance we acquired regrading to the segmentation accuracy. However, it takes much more memory and has much more slow speed. And since we use multi-exemplars per class, therefore, in our experiments, 256 dimensional features are employed.

4.5.3 Ablation Study of Single-Exemplar vs Multi-Examplar

We also tested on single-exemplar as well as multi-exemplars. Specifically, which include the settings of 1,3,5,10 exemplars per class, and with the initialization of Zero and Xavier. Based on the testing

Init way	Num-Exam	Exam-Dim	mIoU19	mIoU16	M13
Zero	1	256	28.79	31.78	36.05
Zero	3	256	28.33	31.16	35.4
Zero	5	256	29.08	32.13	36.41
Zero	10	256	27.96	30.74	35.08
Xavier	1	4096	28.89	31.88	36.25
Xavier	3	4096	28.6	31.63	35.9
Xavier	5	4096	28.37	31.29	35.6
Xavier	10	4096	28.73	31.71	36.0

Table 4.4: Results of exemplar-classifier with different number of exemplars per class

results, we found that using 5 exemplars per class with zero initialization achieve a better balance between the memory-costing and the accuracy. Therefore, we take use of 5 exemplars per class with zero initialization in our experiments.

4.5.4 Ablation Study of w/o normalization

Since the pixel-wise feature and the exemplars has very different scales, then it is better to normalize them to the same magnitude. We conducted three group of experiments to do a comparison. That are multi-exemplar based classifier, multi-exemplar classifier with L1_Norm for the embedding features and the exemplars and multi-exemplar classifier with L2_Norm for the embedding features and the exemplars. And it can be seen from Table 4.5, compared to the other two, using L2_norm achieve much better segmentation performance. By default, in all of our experiments, we using the classifier which has a L2_Norm normalization for the embedding features and exemplars.

Meth	road	side	buil	wall	fenc	pole	ligh	sign	vege	terr	sky	pers	ride	car	truc	bus	trai	moto	bicy	m19	m16	m13
Multi-Exam	74.99	25.1	70.76	13.19	10.09	16.68	16.65	9.44	79.22	31.56	68.92	38.96	1.23	67.54	6.68	4.09	0.0	5.65	0.76	28.5	31.45	35.64
Multi-Exam-L1Norm	74.81	25.6	70.64	13.1	8.78	16.31	16.17	11.37	78.65	30.73	69.41	38.75	0.77	67.07	6.52	4.0	0.0	5.21	0.76	28.35	31.34	35.63
Multi-Exam-L2Norm	75.59	25.67	71.69	13.61	9.31	17.18	17.28	10.73	78.95	31.66	71.59	38.83	1.14	67.99	6.36	5.03	0.0	4.77	0.4	28.83	31.86	36.13

Table 4.5: GTA5 to cityscapes, network with VGG backbone w/o normalization.

4.5.5 Ablation Study of Self-training

Meth	road	side	buil	wall	fenc	pole	ligh	sign	vege	terr	sky	pers	ride	car	truc	bus	trai	moto	bicy	m19	m16	m13
BDL-VGG	78.27	27.08	71.12	13.38	8.24	16.47	16.18	10.73	78.13	30.25	70.41	37.04	1.28	67.92	6.42	3.68	0.0	4.79	0.43	28.52	31.57	35.93
BDL-VGG-s1	75.13	24.5	71.09	13.79	8.82	16.15	16.54	10.11	78.99	31.4	70.73	39.79	0.64	67.81	6.61	4.7	0.0	5.32	0.46	28.56	31.54	35.83
BDL-VGG-s2	76.7	26.25	71.34	13.42	9.23	16.9	16.58	11.13	78.75	31.6	70.97	39.97	0.88	67.69	6.56	4.57	0.0	5.42	0.78	28.88	31.91	36.23
BDL-DeepLab	86.37	34.69	81.22	25.5	26.83	29.62	31.08	29.54	81.28	27.47	77.85	57.75	25.63	78.34	33.73	31.85	4.41	28.6	41.74	43.87	47.99	52.77
BDL-DeepLab-s1	89.14	38.03	81.35	23.95	17.95	28.43	30.15	30.94	82.5	31.4	79.94	58.15	28.97	81.15	21.22	40.04	13.53	28.02	41.27	44.53	48.75	54.59
BDL-DeepLab-s2	87.89	38.06	81.91	26.73	23.58	29.87	31.53	27.35	83.23	33.21	79.87	58.16	26.65	81.88	38.51	36.77	3.63	27.63	42.37	45.2	48.97	54.1

Table 4.6: Self-training Experiments of GTA5 to Cityscapes with DeepLab backbone.

As an important step in UDA, the purpose of class-balanced self-training is to generate pseudo labels in the target domain, with the relatively higher confidence, these samples will be incorporated in the training loop. For each round, the weights of the network and the generated pseudo labels will be iteratively optimized. As can be seen in Table 4.6, both of self-training with two rounds on VGG backbone and DeepLab backbone are given out. Specifically, there are two trends can be seen from the table. The first one is that both of VGG backbone and DeepLab backbone has continuous improvements as the self-training process going. Secondly, compared to the self-training experiments on VGG, experiments on DeepLab backbone achieves much gains, since the much more stronger backbone most likely predict much more confident and accurate labels.

4.5.6 Ablation Study of Convolutional Adaptor

Recent work in UDA task find it is hard to directly align the feature distribution from the source domain to the target domain. Multi-domain try cure this problem through share the majority of the network parameters among different domains, while only store a very small ratio of parameters which can be quickly adapted to the target domain. Specifically, based on the convolution of VGG, serial adapter and parallel adapter is delivered, and both of those two convolutional adapters achieve segmentation performance improvements compared to the convolutions.

Meth	road	side	buil	wall	fenc	pole	ligh	sign	vege	terr	sky	pers	ride	car	truc	bus	trai	moto	bicy	m19	m16	m13
VGG-Multi-Exam	74.75	25.61	71.27	14.01	11.43	17.77	15.96	11.85	79.54	32.42	70.45	42.27	0.97	67.36	6.26	4.23	0.0	4.99	0.56	29.04	32.06	36.14
VGG-SAdapter	76.37	27.5	72.11	14.63	10.58	17.87	17.0	10.01	78.53	30.89	72.87	42.48	1.11	68.53	6.79	4.66	0.0	5.13	0.89	29.37	32.52	36.71
VGG-PAdapter	78.21	27.76	72.19	13.7	10.69	17.21	17.41	9.15	78.55	32.05	72.47	41.79	1.79	69.0	6.91	3.3	0.0	5.53	0.66	29.39	32.46	36.75

Table 4.7: GTA5 to cityscapes, network with Deeplab backbone w/o the convolutional adaptor.

4.6 Conclusion

Targeting at handling unsupervised domain adaptation (UDA) problem, in this paper, a novel method, which is composed of three components, is proposed. Firstly, in contrast to the previous methods, which assume there is a distribution center in the embedding space for both of the source domain and the target domain, we instead use the clustering method, that are k exemplars work together to simulate the distribution center of each class. Secondly, class-balanced self-training strategy is utilized for generating pseudo labels in the target domain, which will be incorporated in the training loop for capturing much more appearance information from the target domain. Thirdly, in order to better align the features from the source domain and target domain, serial adapter and parallel adapter are employed to achieve this purpose. At the moment, we fix the number of exemplars within each class as five, which given much better performance than using single exemplar per class. In the future, smartly choose the number of exemplars within each class would be one of our interests.

Bibliography

P. P. Busto, A. Iqbal, and J. Gall. Open set domain adaptation for image and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017b.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

J. Li, Y. Liu, D. Gong, Q. Shi, X. Yuan, C. Zhao, and I. Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019a.

Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019b.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019.

S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 587–597, 2018.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, A. Mayr, M. Heusel, M. Hofmarcher, M. Widrich, et al. Speeding up semantic segmentation for autonomous driving. In *MLITS, NIPS Workshop*, volume 2, page 7, 2016.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Y. Zou, Z. Yu, B. Kumar, and J. Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

Chapter 5

Meta Learning with Differentiable Closed-form Solver for Fast Video Object Segmentation

Video object segmentation plays a vital role to many robotic tasks, beyond the satisfied accuracy, quickly adapt to the new scenario with very limited annotations and conduct a quick inference are also important. In this chapter, we are specifically concerned with the task of fast segmenting all pixels of a target object in all frames, given the annotation mask in the first frame. Even when such annotation is available, this remains a challenging problem because of the changing appearance and shape of the object over time. In this chapter, we tackle this task by formulating it as a meta-learning problem, where the base learner grasping the semantic scene understanding for a general type of objects, and the meta learner quickly adapting to the appearance of the target object with a few examples, which can be reflected from the training and inference procedure. Our proposed meta-learning method uses a closed form optimizer, the so-called "ridge regression", which has been shown to be conducive for fast and better training convergence. Moreover, we propose a mechanism, named "block splitting", to further speed up the training process as well as to reduce the number of learning parameters. In comparison with the state-of-the art methods, our proposed framework achieves significant boost up in processing speed, while having highly comparable performance compared to the best performing methods on the widely used datasets. Video demo can be found here [1].

5.1 Introduction

The goal of video object segmentation is to distinguish an object of interest over video frames from its background at the pixel level. Fast and accurate video object segmentation plays an important role in robotics research and has various applications, including, but not limited to, robotic vision Lenz et al. [2015], film making Gee [2009], public surveillance Xu et al. [2016].

In contrast to many vision tasks such as image classification Krizhevsky et al. [2012],face recognition Parkhi et al. [2015] and object detection Girshick [2015]; Redmon et al. [2016] for which the performance

[1] Video demo can be found in the following link:
https://www.youtube.com/watch?v=btJlqLj0-nc

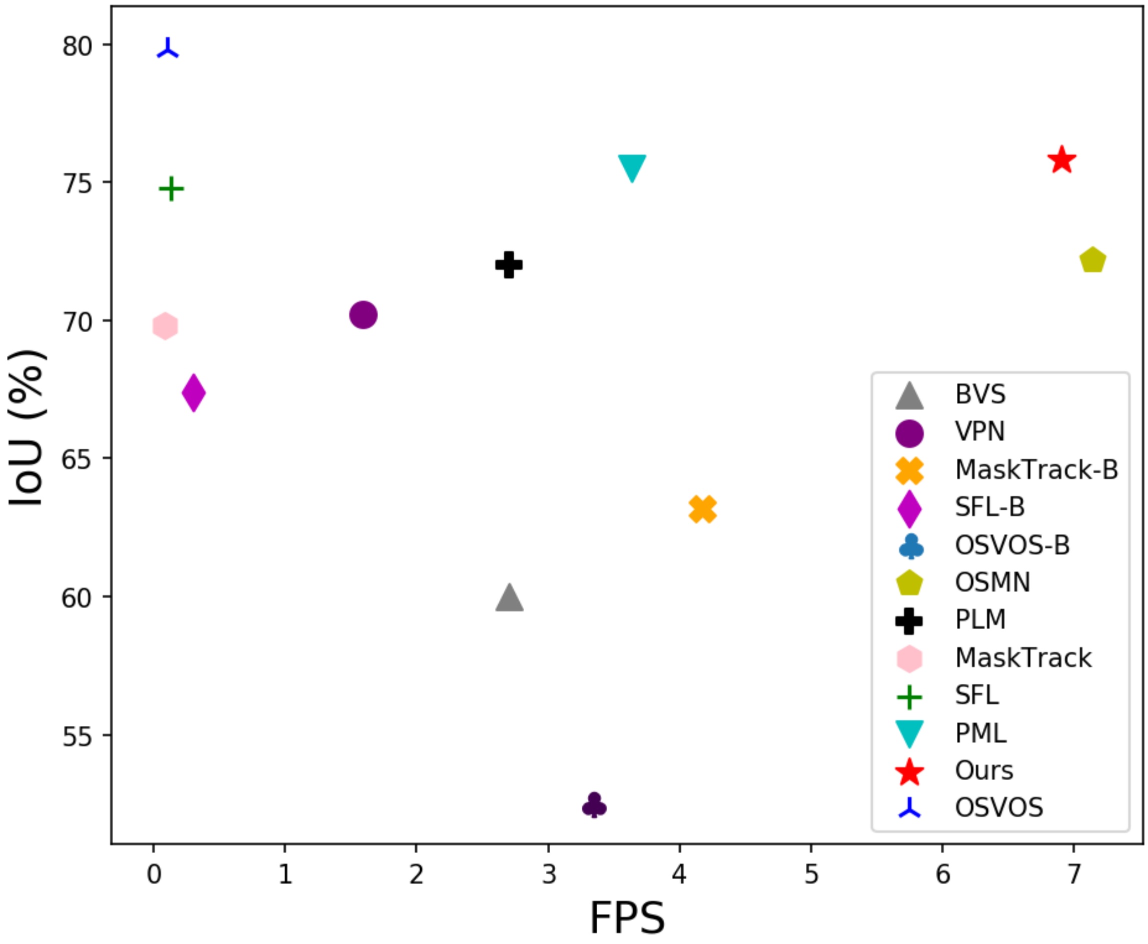


Figure 5.1: A comparison of the quality and the speed of previous video object segmentation methods on DAVIS2016 benchmark. We visualize the intersection-over-union (IoU) with respect to the frames-per-second (FPS).

of the algorithms reach to the point of being suitable for real-world applications, the performance of video object segmentation algorithms are still far beyond the human performance Perazzi et al. [2016]. This is mainly because to acquire the dense pixel-wise labeling is super expensive, thus this problem does not benefit from availability of a massive corpus of training data, unlike the other aforementioned tasks.

Recently, deep learning-based approaches have shown promising progresses on video object segmentation task Caelles et al. [2017]; Maninis et al. [2018]; Voigtlaender and Leibe [2017]. However, they still struggle to satisfy both good accuracy and fast processing inference. In this paper, we aim to bridge this gap.

Inspired by the meta-learning method which is proposed for image classification task Bertinetto et al. [2018], we propose an intuitive yet powerful algorithm for video object segmentation, in which the reference frame is available with its annotated mask. In addition, we also propose block splitting to speed up the matrix computation, significantly improving the efficiency of the whole framework. Our objective is to train a system that can “adapt” this annotation information to subsequent frames in a fast yet flexible way at inference time. Specifically, at inference time the reference frame (i.e. one

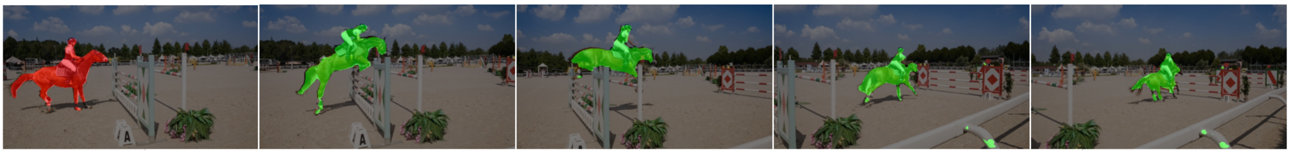


Figure 5.2: Example result of our technique: The segmentation of the first frame (red) is used to learn the model of the specific object to track, which is segmented in the rest of the frames independently (green). One every 10 frames shown of 50 in total.

with ground-truth annotation) is mapped to vector in a high dimensional embedding space $X = \phi(I)$ using a CNN ϕ.

We then determine using *ridge regression* Myers and Myers [1990], the coefficients of a matrix W that best maps X to the ground truth, $Y = WX$. W is then the video-specific "adaptor", and it maps the feature vectors for every query image (*i.e.* every other image in the video sequence) to their predicted segmentation masks. Training comprises the process of learning the mapping $\phi(.)$ by presenting the network with pairs of images (from a variety of videos but with each pair coming from the same video), each with ground-truth annotation, and back-propagating the loss through ϕ. This is illustrated in Figure. 5.3 and described in more detail later in the paper.

We observe that a limitation of the proposed approach is that the ridge regression scales poorly with the dimension of the feature feature produced by $\phi(.)$ because the optimization requires an huge matrix inversion. We address this through the use of a "block splitting" method that approximates the matrix in block diagonal form, meaning the inversion can be done much more efficiently.

Our main contributions are three-fold:

- A meta-learning based method for video object segmentation is developed, using a closed form solver (ridge regression) as the internal optimizer. This is capable of performing fast gradient back-propagation and can adapt to previously unseen objects quickly with very few samples. Inference (i.e. segmentation of the video) is a single forward pass per frame with no need for fine-tuning or post-processing.

- Ridge regression in high-dimensional feature spaces can be very slow, because of the need to invert a large matrix. We address this by using a novel block splitting mechanism, which greatly accelerates the training process without damaging the performance.

- We demonstrate the state-of-the-art video segmentation accuracy relative to all others methods of comparable processing time, and even better accuracy than many slower ones (see Figure. 5.1).

5.2 Related Works

5.2.1 Semi-supervised Video Object Segmentation

The goal of video object segmentation is to 'cutout' the target object(s) from the entire input video sequence. For semi-supervised video object segmentation, the annotated mask of the first frame is given, and the algorithm is designed to predict the masks of the rest frames in the video. There are

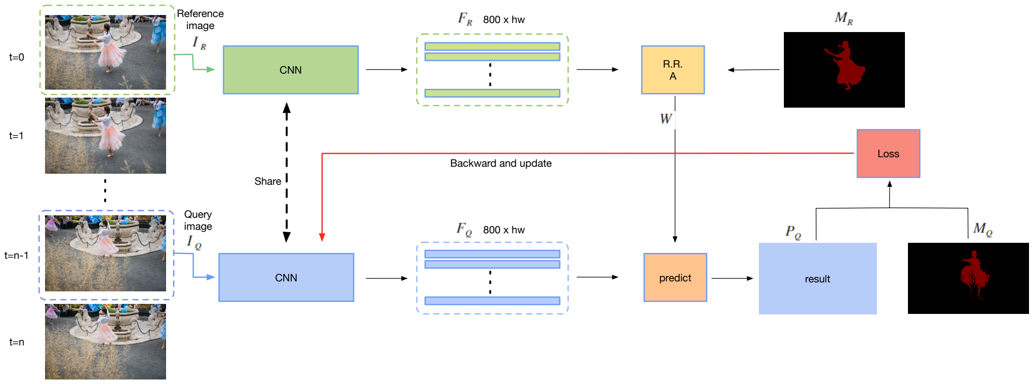


Figure 5.3: Workflow of the proposed method. An image pair sampled from the same video as the input to the network. The first image I_R and its annotation M_R as the reference frame, and the second image I_Q and its annotation M_Q (or prediction P_Q during inference) as the query frame. The image pair first passes through the *feature extractor* (DeepLabv2 Chen et al. [2017] with ResNet101 He et al. [2016]) to compute a 800D embedding tensor F_R, F_Q. Then a *mapping matrix* W between F_R and M_R is calculated in the reference frame (Eq. 1) using ridge regression. After that, the prediction result P_Q in the query frame is acquired by multiplying F_Q and W (Eq. 2). During training, the loss error between P_Q and M_Q is back-propagated to enhance the network' adaptation ability between the reference frame and the query frame. During inference, the reference frame (I_R and M_R) is always the first frame, and the query image I_Q is the rest sequence from the same video. Through iterative meta-learned, our network is capable of quickly adapting to unseen target object(s) with a few examples.

three categories in this spectrum. The first one, which include MSK Perazzi et al. [2017], MPNVOS Sun et al. [2018] etc, is to use optical flow to track the mask from the previous frame to the current frame. Similarly, the second category formulates the optical flow and segmentation in two parallel branches, and utilizes the predicted mask from the previous frame as a guidance, some representatives are Segflow Cheng et al. [2017], OSNM Yang et al. [2018] etc. The final class which keeps the state-of-the-art performance on DAVIS benchmark Perazzi et al. [2016] is to try to over-fit the appearance of the target object(s), and expect the method can generalize in the subsequent frames. Specifically, OSVOS Caelles et al. [2017] uses one-shot learning mechanism to conduct fine-tuning on the first frame of test video to capture the appearance of the target object(s), and conduct inference on the rest frames. The limitations of OSVOS are: (1) it can not adapt to the unseen parts (2) when dramatic changes of appearance happen in subsequent frames, the method's performance significantly degrade. Inspired by the overall design principle of OSVOS, there are some following methods which employ various additional ingredients to improve the segmentation accuracy. Such as OSVOS-S Maninis et al. [2018], OnVOS Voigtlaender and Leibe [2017], but they all suffer the limitation of super-slow for inference.

In this paper, we mainly target to fast video object segmentation, since no optical flow and fine-tuning processes are used, the proposed method is appropriate for real-world applications.

5.2.2 Meta Learning

Meta learning is also named learning to learn Naik and Mammone [1992]; Schmidhuber [1987], it is an alternative to the de-facto solution that has emerged in deep learning of pre-training a network using a large, generic dataset (eg ImageNet Deng et al. [2009]) followed by fine-tuning with a problem-specific dataset. Meta-learning aims to replace the fine-tuning stage (which can still be very expensive) by training a network that has a degree of plasticity so that it can adapt rapidly to new tasks. For this reason it has become a very active area recently, especially with regard to one-shot and few-shot learning problems Fei-Fei et al. [2006]; Lake et al. [2015].

Recent approaches for meta-learning can be roughly put into three categories: (i) metric learning for acquiring similarities; (ii) learning optimizers for gaining update rules; and (iii) recurrent networks for reserving the memory. In this work, we adopt the meta-learning algorithm that belongs to the category of learning optimizers. Specifically, inspired by Bertinetto et al. [2018] which was originally designed for image classification, we adopt ridge regression, which is a closed-form solution to the optimization problem. The reason for using it is because, compared with the widely-used SGD LeCun et al. [1998] in CNNs, ridge regression can propagate gradient efficiently, which is matched with the goal of *fast mapping*. Through extensive experiments, we demonstrate that the proposed method is in the first echelon regarding to speed for fast video object segmentation, while obtaining more accurate results without any post-processing.

5.2.3 Fast Video Object Segmentation

A few previous methods proposed to tackle fast video object segmentation. In particular, FAVOS Cheng et al. [2018] first tracks the part-based detection. Then, based on the tracked box, it generates the part-based segments and merges those parts according to a similarity score to form the final segmentation results. The limitation of FAVOS is that it can not be learned in an end-to-end manner, and heavily relies on the part-based detection performance. OSNM Yang et al. [2018] proposes a model which is composed of a modulator and a segmentation network. Through encoding the mask prior, the modular can help the segmentation network quickly adapt to the target object. RGMP Wug Oh et al. [2018] shares the same spirit with OSNM. Specifically, it employs a Siamese encoder-decoder structure to utilize the mask propagation, and further boosts the performance with synthetic data. The most similar work to ours is PML Chen et al. [2018], which formulates the problem as a pixel-wise metric learning problem. Through the FCN Long et al. [2015], it maps the pixels to high-dimensional space, and utilizes a revised triplet loss to encourage pixels belonging to the same object much closer than those belonging to different objects. Nearest neighbor (NN) is required for retrieval during inference. In contrast our meta-learning approach acquires a mapping matrix between the high-dimensional feature and annotated mask in reference image using ridge regression, and then can be adapted rapidly to generate the prediction mask. Compared to baseline method PML Chen et al. [2018], our method is twice faster and achieves 3.8 percent gains regarding to segmentation accuracy. And with the same efficiency, the J *mean* of our method is 3.4 percent better than OSNM Yang et al. [2018] on the DAVIS2016 Perazzi et al. [2016] validation set.

5.3 Methodology

5.3.1 Overview

We formulate the video object segmentation as a meta-learning problem. For each image pair which comes from a same video, ridge regression is used as the optimizer to learn the base learner. Meta learner is naturally built through the training process. Once the meta learner is learned, it possesses the ability of *fast mapping* between the image features and object masks, and can be adapted to unseen objects quickly with the help of the reference image.

According to the phase that user input involved in the training loop, the current existing methods can be classified into three categories.

User input outside the network training loop This category utilizes the user input to fine-tune the network to over-fit the appearance cues of target object(s) during inference. The representatives are OSVOS Caelles et al. [2017] and its following works Bao et al. [2018]; Maninis et al. [2018]; Voigtlaender and Leibe [2017]. Since online fine-tuning is required during inference, the limitation of these algorithms is time-consuming, which usually take seconds per image, thus is not practical for the real-world applications.

User input within the network training loop This category of work injects the user input as the additional input for training the network. Through this way, no online fine-tuning is needed. These algorithms incorporate the user input either by using a parallel network or concatenating the image with the user input Wug Oh et al. [2018]; Yang et al. [2018]. One limiation of this kind of methods is that the model needs to be recalculated once the user input changes, thus it is not practical for adaptation especially for long videos.

User input is detached from the network training loop In contrast to the previous methods, our algorithm shares the same spirit with PML Chen et al. [2018] in design. The network and user input are detached, and the user input can be much more flexible. Moreover, once the user input is given (for example, the annotation in the reference image), the network can quickly adapt to the target objects without any extra operations.

5.3.2 Segmentation as Meta-Learning

For simplicity, we assume single-object segmentation case, and the annotation of first frame is given as the user input. Note that our method can also be applied for multi-objects and easily extended to other types of user input, e.g., scribble, clicks etc.

We adopt the following notation:

- C: the number of feature channels (in our case 800);
- w, h: the spatial resolution of the extracted features (in our case 1/8th of the orginal image size);
- F_R, F_Q: the feature tensors of size $C \times h \times w$ produced by ϕ;
- X: a flattened tensor of F_R or F_Q, with shape $h \cdot w \times C$;
- Y: the flattened tensor of annotation mask M_R or M_Q, with shape $h \cdot w \times 1$;
- W: the *mapping matrix* of size $C \times 1$ between the feature space and annotation mask.

As noted above, there are two components to the learner: (i) $\phi(.)$ an embedding model that maps images to a high-dimensional feature space, $C \times h \times w$; and (ii) an adaptor W of size $C \times 1$, found using ridge regression, that maps the embedded features to a (flattened) segmentation mask (of size $h \cdot w \times 1$).

Embedding Model We adopt DeeplabV2 Chen et al. [2017] built on the ResNet-101 He et al. [2016] backbone structure as our feature extractor ϕ. This choice allows a direct comparison of our method with the baseline, PML Chen et al. [2018]. First, we use the pretrained model on COCO Lin et al. [2014] dataset as the initialization for semantic segmentation. Then the ASPP Chen et al. [2017] layer for classification is removed and replaced by our video-specific mapping W.

Ridge Regression Ridge regression is a closed form solver and widely-used in machine learning community Nouretdinov et al. [2001]; Saunders et al. [1998]. The learner seeks W that minimizes Λ as follows:

$$\begin{aligned}\Lambda(X, Y) &= \underset{W}{\arg\min}\, ||XW - Y||^2 + \lambda ||W||^2 \\ &= (X^T X + \lambda I)^{-1} X^T Y\end{aligned} \tag{5.1}$$

where, X, Y and W are as defined above, and λ is a regularization parameter, and set to 5.0 in all of our experiments. As can be seen in Figure. 5.3, during training, an image pair as well as their annotations are sampled from the same video sequence. The feature F_R extracted from the reference image I_R (in the Figure 2.3 this is the first image) and its annotation M_R will be used to calculate the mapping matrix W.

$$P_Q = F_Q \times W \tag{5.2}$$

(where we abuse notation and use the unflattened feature tensors for clarity).

For the query image I_Q, likewise we compute the feature F_Q, map these to the predicted segmentation mask P_Q using Equation 5.2 in which W is the matrix computed from the reference image and its ground truth. The loss between the prediction mask P_Q and the annotation M_Q for the query provides the back-propagation signal to improve ϕ's ability to produce adaptable features.

During inference in our case, the reference frame I_R will be always the first frame, for which the annotation mask is provided, and the query frames I_Q will be the rest of frames in the same video.

5.3.3 Block Splitting

Thanks to ridge regression, the computation of the *mapping matrix* and gradient back-propagation are already very fast compared with other algorithms, which also focus on video object segmentation.

$$F(X) = (X^T X + \lambda I)^{-1} \tag{5.3}$$

During the experiments, we found the higher dimension of the feature used as the input for meta-learning module, the more accurate segmentation results likely be achieved. However, we also observed that the higher dimension of the feature being utilized, the slower of the training process. Specifically, during the computation of *mapping matrix W*, it involves a matrix inverse calculation. as denoted by Equation 5.3, which will become the bottleneck of fast propagation when the very high dimensional feature is used.

In order to further speed up the training process of the proposed network, we deliver a *block splitting*

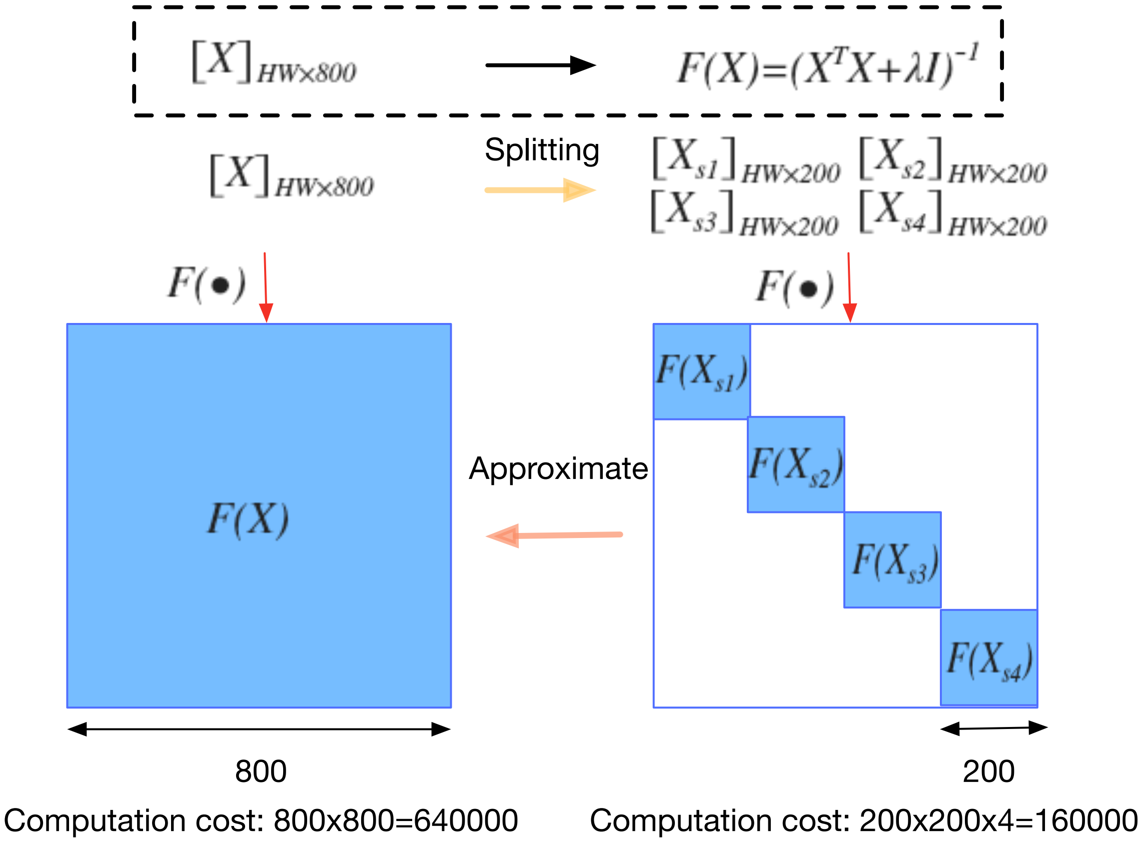

$[X]_{HW\times800}$
$F(X)=(X^TX+\lambda I)^{-1}$
$[X]_{HW\times800}$
Splitting
$[X_{s1}]_{HW\times200}$ $[X_{s2}]_{HW\times200}$
$[X_{s3}]_{HW\times200}$ $[X_{s4}]_{HW\times200}$
$F(\bullet)$
$F(\bullet)$
$F(X)$
Approximate
$F(X_{s1})$
$F(X_{s2})$
$F(X_{s3})$
$F(X_{s4})$
800
200
Computation cost: 800x800=640000
Computation cost: 200x200x4=160000

Figure 5.4: Illustration of the proposed *block splitting*: during matrix inverse calculation of ridge regression, the computation of the higher dimensional feature is approximated by the sum of computation of that lower dimensional features. Which can effectively speed up the training process as well as reducing the parameters and memory.

mechanism, and its work principle as shown in Figure. 5.4. In particular, our motivation is that the matrix inverse computation $F(X)$ for much high-dimensional feature (eg. 800D) can be approximated by the sum of the computations of that relative low-dimensional features (eg. 200D $\times$ 4). From the work principle, it can be viewed that a $n \times n$ matrix can be approximated by four $n/4 \times n/4$ irrelevant diagonal matrix.

The advantages of using the proposed block splitting mechanism are: Firstly, it can largely speed up the matrix inverse process involved in ridge regression, thus it saves the training time to some extent. Secondly, through the matrix approximation step as aforementioned, the network parameters involved in the ridge regression as well as memory utilized in our network are reduced. The experimental evidence can be found in Ablation Study (Section 5.5).

5.3.4 Training

Training Strategy For training, optimizer is SGD with momentum 0.9, with weight decay 5e-4. We use the DeepLabV2 Chen et al. [2017] with backbone network ResNet-101 He et al. [2016] as the *feature extractor*, and the constant learning rate, i.e. 1.0e-5, is used during the whole training process. The dimension of extracted feature is 800 outputed by the *feature extractor*, which is used as the input for

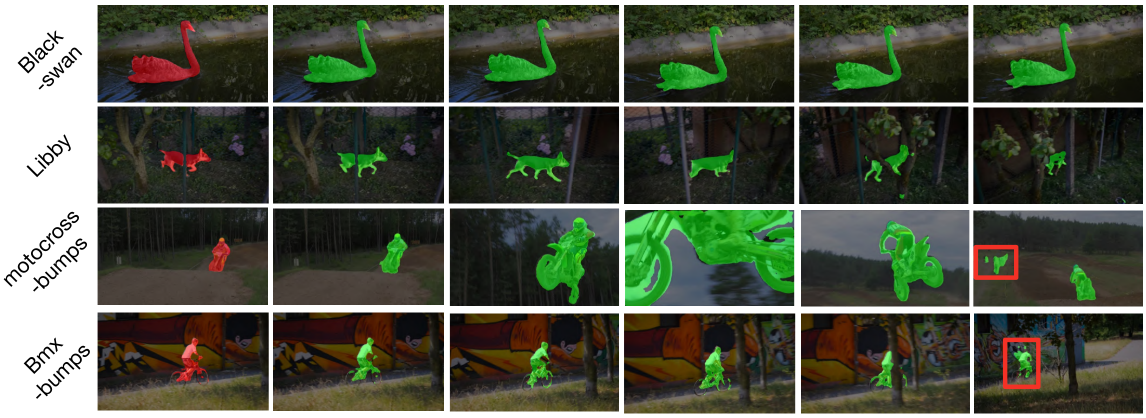


Figure 5.5: Qualitative results: Homogeneous sample of DAVIS sequences with our result overlaid.

Method	DAVIS	Online Tuning	OptFlw	CRF	BS	Speed(s)
OFL	68.0	-	✗	✓	✗	42.2
BVS	60.0	-	✗	✗	✗	0.37
ConvGRU	70.1	✗	✓	✗	✗	20
VPN	70.2	✗	✗	✗	✗	0.63
MaskTrack-B	63.2	-	✗	✗	✗	0.24
SFL-B	67.4	✗	✓	✗	✗	0.30
OSVOS-B	52.5	✗	✗	✗	✗	**0.14**
OSNM	72.2	✗	✗	✗	✗	**0.14**
PML*	72.0	✗	✗	✗	✗	0.28
Ours	**75.8**	✗	✗	✗	✗	0.145
PLM	70.0	✓	✗	✗	✗	0.50
SFL	74.8	✓	✗	✗	✗	7.9
MaskTrack	69.8	✓	✗	✗	✗	12
OSVOS	**79.8**	✓	✓	✗	✓	10

Table 5.1: Performance comparison of our approach with recent approaches on DAVIS 2016 validation set Performance measured in mean IoU. PML* denotes PML without spatial-temporal and online adaptation which is the same case with our method.

the meta-learning module.

Loss BCEWithLogitsLoss[2] is employed for training the proposed network, it essentially is a combination of the Sigmoid layer and binary cross entropy (BCE) loss, it benefits from the *log-sum-exp* trick for numerical stability. And compared to BCE loss, it is more robust and less likely to cause numerical problem when computing the inverse matrix in the ridge regression step.

$$\begin{aligned} \ell(x,y) &= L = \{l_1, ..., l_N\}^T \\ l_n &= -w_n[y_n \cdot \log \delta(x_n) + (1 - y_n) \cdot \log(1 - \delta(x_n))] \end{aligned} \tag{5.4}$$

where N is the batch size. x_n is the input of the loss calculation, and y_n ($y_n \in [0,1]$) is the ground truth label. w_n is a rescaling weight given to the loss of each batch element.

[2] https://pytorch.org/docs/stable/nn.html

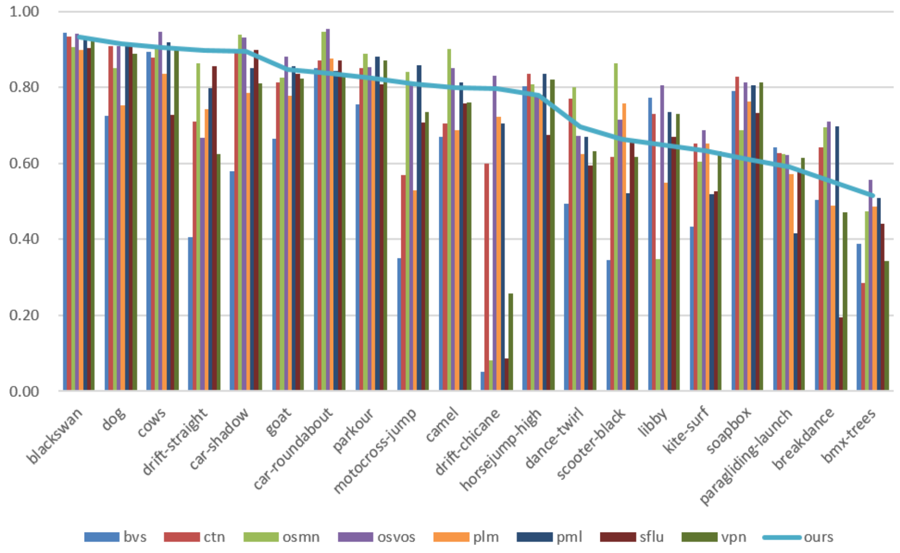


Figure 5.6: Per-sequence results of mean region similarity (J) . Sequences are sorted by the performance of our algorithm.

5.4 Experiments

5.4.1 Dataset

We verify the proposed method both on DAVIS2016 Perazzi et al. [2016] and SegTrack v2 Li et al. [2013] datasets.

On DAVIS2016, which contains 50 pixel-level annotated video sequences, and each video only contains one target object for segmenting. Among these 50 video sequences, 30 video sequences as the training set with which the annotated mask is provided for every frame. And another 20 video sequences as the validation set, and only the annotation of the first frame is allowed to access.

SegTrack v2 Li et al. [2013] is extended from SegTrack Tsai et al. [2012] dataset. Both of them contain the dense pixel-level annotation for each frame within each video. For segtrack v2 dataset, we test our algorithm on all the sequences which contain one target object.

5.4.2 Results on DAVIS2016

Quantitative Results Table 5.1 shows the experimental results on DAVIS2016 Perazzi et al. [2016] on different methods. Apart from the performance (measured by *J mean*), switches for online-fining, using optical-flow, dense CRF (CRF) and boundary snapping (BS) are also described. Meanwhile, the inference time is also shown. In particular, compared with most of the competitors, our algorithm shares the same or much faster processing time with superior performance regarding the segmentation accuracy. Compared with OSVOS Caelles et al. [2017], for which the online fine-tuning is necessary, our method just takes a smaller fraction of time to do inference. Compared to the baseline method PML Chen et al. [2018] which use the same *feature extractor*, our method is twice faster and achieve 3.8 percent gains with the same settings. Compared OSNM Yang et al. [2018], with the same efficiency, our method achieve 3.4 percent improvements regarding to the segmentation accuracy.

Qualitative Results Figure. 5.5 demonstrates some visualized results of our method. As shown in Figure. 5.5, our method is not only good at recovering object details (e.g., the results on the sequence of *blackswan*), but also robust against heavy occlusions (eg. the results on the sequences *bmx-bumps* and *libby*, dramatic movement as well as abrupt rotation (eg. the results on the sequence *motocross-bumps*). However, there are very few scenarios which may lead to failure cases (denoted by the red box), and mainly caused by the (noisy) objects which have not appeared at the first frame of the video, and can be easily cured by some post-processing steps, including tracking Cheng et al. [2018], online adaptation Chen et al. [2018]; Voigtlaender and Leibe [2017].

In Figure. 5.7, we show some visualized results compared with OSVOS Caelles et al. [2017] and PML Chen et al. [2018]. For the *breakdance*, *scooter-black* and *dance-jump* sequences, which contain fast moving and abrupt rotation, OSVOS Caelles et al. [2017] performs worse than PML Chen et al. [2018]. And for the *dog* sequence, PML Chen et al. [2018] can not achieve a satisfied result due to the dramatic change of the light conditions. However, on both of these two scenarios, the proposed method performs better than both of OSVOS and PML, which is benefit from robust adaptation ability of our network.

5.4.3 Results on SegTrack Dataset

In Figure. 5.8, some visualized results in the segTrack Tsai et al. [2012] dataset are shown. Which are acquired by direcly utlized the model trained on DAVIS2016 dataset. As can be seen, in most cases, our model maintain a good segmentation accuracy, and with a few case fails (as denoted by the red box), which mainly due to the dramatically changes of the light conditions and exact same appearance between the background and the target object. These results prove our method has a better generalization ability and can be quickly adapted to other unseen objects with very few examples (here, only the annotation in the first frame is provided).

5.5 Ablation Study

5.5.1 Comparison with PML with different adds-on

Method	Spat.-Temp.	Online Adapt.	MaskIoU
PML-Abal1	✗	✗	72.0
PML-Abal2	✗	✓	73.2
PML-Abal3	✓	✗	74.3
PML	✓	✓	75.5
Ours	✗	✗	**75.8**

Table 5.2: Comparison with basedline method PML Chen et al. [2018] under different settings.

Compared to baseline method PML Chen et al. [2018] which with same backbone network (DeepLabV2) under the same settings, our method achieved 3.8 percent improvement regarding to MaskIoU accuracy as shown in Table 5.2. And compared with baseline method adding spatial temporal attention and online adaptation, our method is still slightly better and twice faster.

Split No	Feature	Speed(s)	Memory	Computation Cost
1	800	1.50	11590	640k
2	400	1.23	11720	320k
4	200	0.75	11580	160k
8	100	0.86	11584	80k

Table 5.3: Ablation study on *block splitting*: feature dimension, running speed, memory and computation cost with different settings are listed out.

5.5.2 Feature Dimension and Block Splitting

As mention in Section 5.3.3, since our meta learning module (ridge regression) requires the computation of matrix inverse, the training speed will varies significantly regrading the features with various dimensions utilized for this step. And based on the fact that low dimensional features usually have the faster speed but lose some details of image information. On the contrary, high dimensional features are time-consuming but carry much rich information. We propose a *block splitting* mechanism to train the meta learner. In Table 5.3, the splitting number (of feature), feature dimension, running speed (per iteration), memory cost (of the whole network), as well as computation cost (of the computation of matrix inverse) with different settings are listed out. As can be seen, with the feature dimension decreasing, the overall trend are running speed increasing, computation cost decreasing, dramatically. However the memory cost reduce slightly, which mainly because of the backbone *feature extractor* take up most of the memory usage. All the numbers are tested on the single GPU card (with type of GTX 1080). Please note, the performance of using different splits change slightly during the preliminary experiments. The reason for using feature with 800D is based on the observation that: The higher dimension of the feature, the stronger representation ability and the slower training speed. 800D feature is somehow a compromise between the good performance and fast training speed.

5.5.3 Per Sequence Performance Analysis

In Figure. 5.6, *J mean* of per sequence of different methods are outlined. It is sorted according our algorithm's performance in each sub-sequence, which provides a more intuitive understanding for the proposed algorithm. Firstly, the proposed method achieve a better video segmentation accuracy when compared to many other methods. Secondly, our algorithm works quite well on most of sequences, even on the most challenging sequences, e.g., *breakdance* and *bmx-tree*, the *J mean* is above 0.5. Thirdly, benefit from the quick adaption ability of meta-learning, around half of sequence achieve *J mean* over 0.8. Moreover, our method can well recover the object details as well as robust against fast movement and heavy occlusion, which are aligned with our conclusion in Section 5.4.2

5.6 Conclusion

In this paper, we explore applying meta-learning into video object segmentation system. A closed form optimizer, i.e., ridge regression, is utilized to update the meta learner, which achieves fast speed while maintains the superior accuracy. Through iteratively meta-learned, the network is capable of conducting *fast mapping* on unseen objects with a few examples available. Compared to the fine-tuning methods, our algorithm with similar performance but just a smaller fraction time is required, which is appeal to the real-world applications. In addition, a block splitting mechanism is delivered to speed up the training process, which also has the benefits of reducing parameters and saving memory. In future work, we would like to use other basic optimizers, such as, Newton's methods and logistic regression. Meanwhile, based on the flexible design of our meta-learner, instead of inferring the rest frames from the given whole annotation of the first frame. Inferring whole object from only part of annotation or user feedback is also worth to investigate.

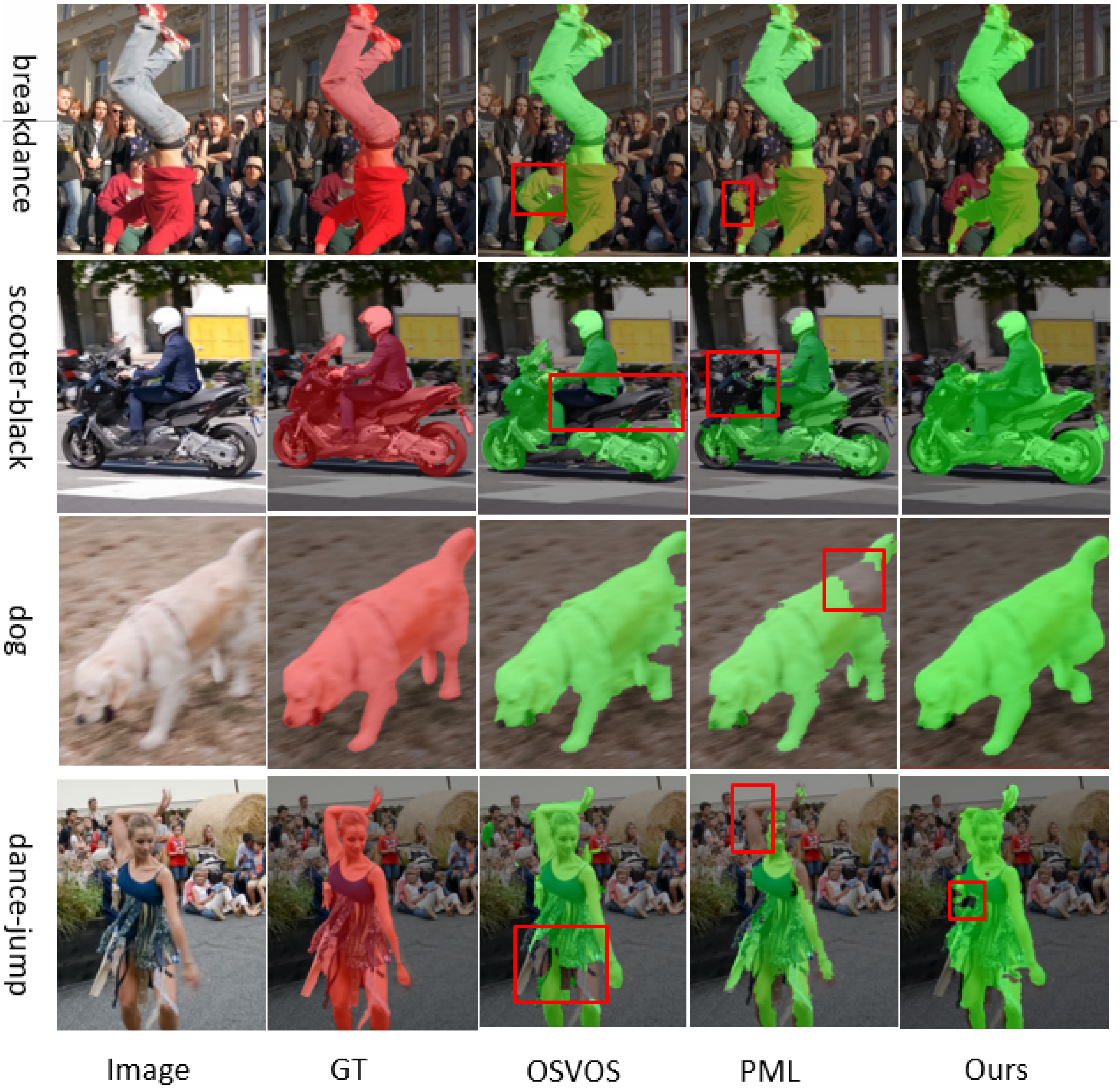
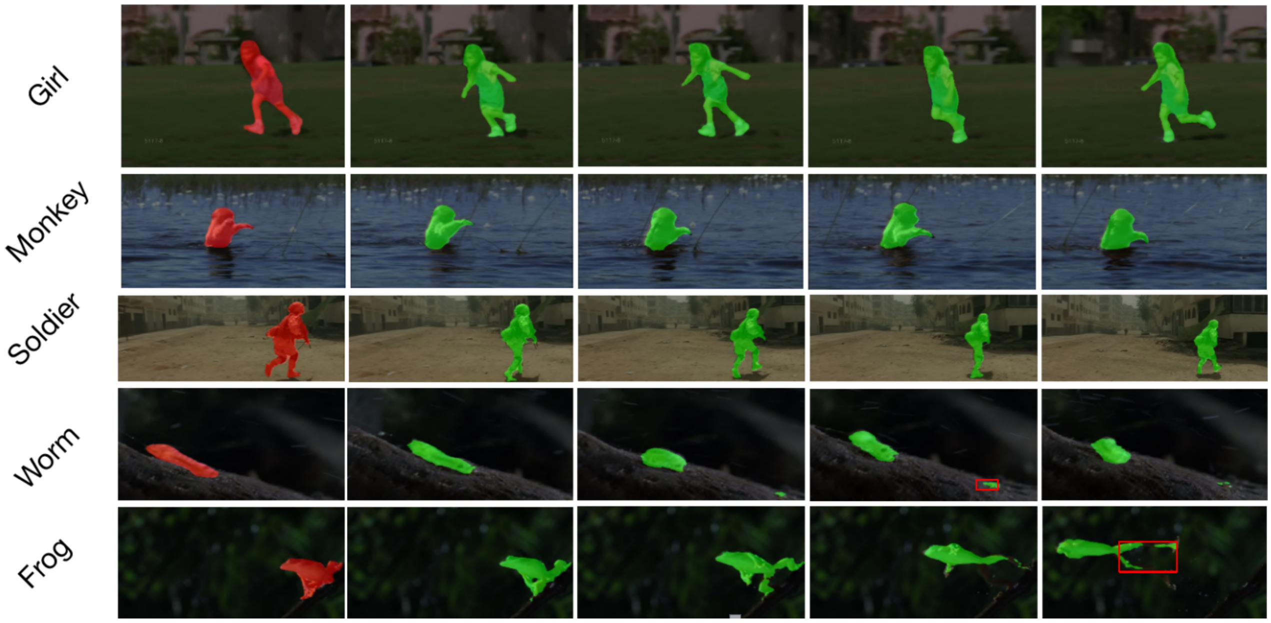


Figure 5.7: Visualized comparison between the proposed method and other methods. With the red box to denote the error region.

Figure 5.8: Qualitative results: Homogeneous sample of SegTrack sequences with our result overlaid.



Bibliography

L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018.

J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7415–7424, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

J. P. Gee. Deep learning properties of good digital games: How far can they go? In *Serious games*, pages 89–104. Routledge, 2009.

R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1515–1530, 2018.

R. H. Myers and R. H. Myers. *Classical and modern regression with applications*, volume 2. Duxbury press Belmont, CA, 1990.

D. K. Naik and R. J. Mammone. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE, 1992.

I. Nouretdinov, T. Melluish, and V. Vovk. Ridge regression confidence machine. In *ICML*, pages 385–392, 2001.

O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. 2015.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. 1998.

J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

J. Sun, D. Yu, Y. Li, and C. Wang. Mask propagation network for video object segmentation. *arXiv preprint arXiv:1810.10289*, 2018.

D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *International journal of computer vision*, 100(2):190–202, 2012.

P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018.

Z. Xu, C. Hu, and L. Mei. Video structured description technology based intelligence analysis of surveillance videos for public security applications. *Multimedia Tools and Applications*, 75(19): 12155–12172, 2016.

L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

Chapter 6

In Defense of OSVOS

As a milestone for video object segmentation, one-shot video object segmentation (OSVOS) has achieved a large margin compared to the conventional optical-flow based methods regarding to the segmentation accuracy. Its excellent performance mainly benefit from the three-step training mechanism, that are: (1) acquiring object features on the base dataset (i.e. ImageNet), (2) training the network on the training set of the target dataset (i.e. DAVIS-2016) to be capable of differentiating the object of interest from the background, that form the parent network. (3) online fine-tuning the interested object on the first frame of the target test set to overfit its appearance, then the model can be utilized to segment the same object in the rest frames of that video. In this paper, we argue that for the step (2), OSVOS has the limitation to 'overemphasize' the generic semantic object information while 'dilute' the instance cues of the object(s), which largely block the whole training process. Through adding a common module, video loss, which we formulate with various forms of constraints (including weighted BCE loss, high-dimensional triplet loss, as well as a novel mixed instance-aware video loss), to train the parent network in the step (2), the network is then better prepared for the step (3), i.e. online fine-tuning on the target instance. Through extensive experiments using different network structures as the backbone, we show that the proposed video loss module can improve the segmentation performance significantly, compared to that of OSVOS. Meanwhile, since video loss is a common module, it can be generalized to other fine-tuning based methods and similar vision tasks such as depth estimation and saliency detection.

6.1 Introduction

With the popularity of all kinds of mobile device and sensors, countless video clips are uploaded and shared through the social media platforms and video websites every day. Smartly analysing these video clips are very useful yet quite challenging. The revival of deep learning boosts the performance of many recognition tasks on static images to a level that can be matched with human beings, including object classification Girshick [2015]; Liu et al. [2019]; Redmon and Farhadi [2018], semantic segmentation Chen et al. [2018a]; Liu et al. [2018]; Long et al. [2015] and object tracking Cao et al. [2018]; Wang et al. [2018]. Compared to static images, video clips contain much more rich information, and the temporal correlations among inter-frame, if being used appropriately, it can significantly improve the performance of the corresponding tasks on static images. As one of the most active fields in computer vision community, video object segmentation aims to distinguish the foreground objects(s)

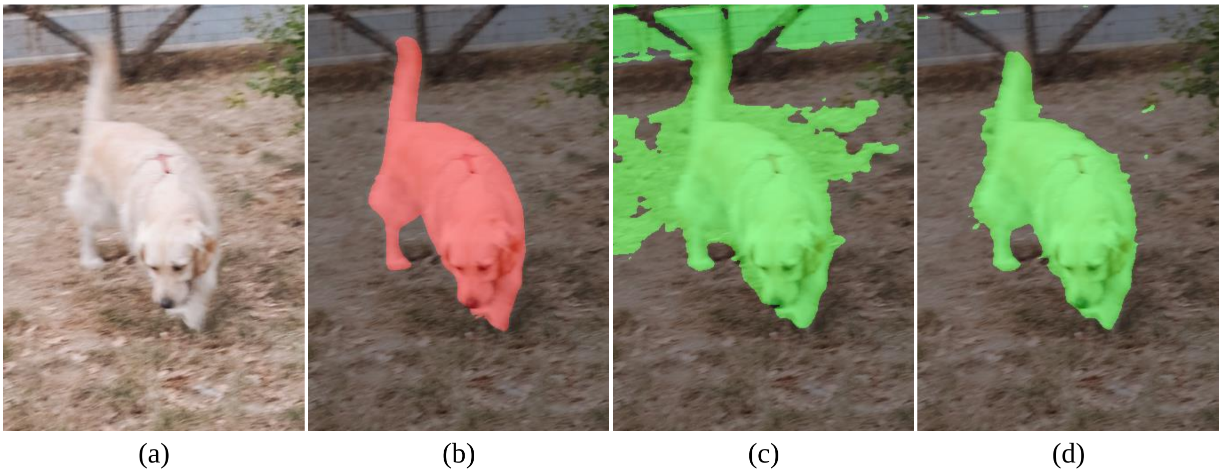


Figure 6.1: A visualized example of OSVOS and OSVOS-VL. (a) Image (b) Ground truth (c) Segmentation with OSVOS (d) Segmentation with OSVOS-VL (the proposed method with video loss).

from the background in pixel level. In 2017, one-shot video object segmentation (OSVOS) is proposed by Caelles et al Caelles et al. [2017], as a milestone in this research field, which achieves over 10 % improvements compared to the previous conventional methods regarding the segmentation accuracy.

Motivation and principle of OSVOS The design of OSVOS is inspired by the perception process of human beings. Specifically, when we recognize an object, the first thing come into our view are the image features such as corners and textures of the scene, then we can distinguish the object(s) from background through shape and edge cues, which also named objectness. Finally, based on the rough localization from the above two steps, we will pay attention on the details of the target instance.

In particular, OSVOS utilizes a fully-convolutional neural network (FCN) Long et al. [2015] to conduct video object segmentation, and the three phases are:

- **Acquire object features**: to acquire the generic semantic features from *ImageNet*.
- **Train parent network**: to train a network on DAVIS-2016 training set, which is capable of distinguishing the foreground object(s) from the background.
- **Online fine-tuning**: based on the *parent network*, to train the network which is overfitting the appearance of the target instance on the first frame.

The Pros and Cons of OSVOS

- **Pros:** The online fine-tuning process of OSVOS wishes to fully acquire the appearance of the target object in the first frame. Hence, it is capable of handling the fast moving, abrupt rotation as well as heavy occlusion, which are the limitations of the conventional optical flow based methods.

- **Cons:** i) when similar (noisy) objects appear in the subsequent frames of the video sequence, they will be wrongly segmented as foreground objects. ii) when the appearance of the target object changes dramatically in the later phase of the video sequence, the algorithm fails to segment the new appearance parts.

The motivation of video loss We propose the video loss in defense of OSVOS based on two observations:

- For CNN, the low-level layers have relatively large spatial resolutions, and carry more details about the object instance, while the high-level layers have more stronger abstract and generalization ability, leading to carry more category information. Especially, in the second phase of OSVOS, i.e. training the *parent network*, it actually tries to fine-tune the network to acquire the ability of distinguishing the objects from the background. However, it dilutes the 'instance' information. And quickly adapts to the specific (target) instance, which is exactly the need of third phase (i.e. *online finetuning*). Video loss can effectively 'rectify' the training process of *parent network*, and make it be better prepared for the *online fine-tuning*.
- Each video is supposed to maintain an *average object*, and through mapping, we expect that the objects from a same video are close to each other in the embedding space, while the objects from different videos are far away from each other. By this way, video loss can help the network to maintain an *average object* for each video squence.

6.2 Related Works

For the task of semi-supervised video object segmentation, the annotations of first frame is given, and the algorithm is expected to infer the object(s) of interest in the rest frames of the video. According to design principle, the existing algorithms which achieve the state-of-the-art performance on DAVIS benchmark Perazzi et al. [2016] for semi-supervised video object segmentation can be roughly classified into three categories:

6.2.1 Tracking based Methods

In this category, one stream of methods employ the optical flow to track the mask from the previous frame to the current frame, including MSK Perazzi et al. [2017], MPNVOS Sun et al. [2018] etc, one limitation of those methods is that they can not handle heavy occlusion and fast moving. Most recently, there are an emergence of methods which use the ReID technique to conduct the video object segmentation, including PReMVOS Luiten et al. [2018] and FAVOS Cheng et al. [2018]. Specifically, FAVOS using ReID to tackle the part-based detection box first, and through merging the (box) region based segments to form the final segmentation. PReMVOS firstly generate instance segmentation proposals in each frame, and then take use of the ReID technique to do data association to pick the correct segments in temporal domain, which can largely reduce the background noises brought by other nearby or overlapped object(s).

6.2.2 Adaptation based Methods

For this category of methods, the core idea is utilizing the mask priors acquired from the previous frame(s) to be the guidance, to supervise the prediction in the current frame. Specifically, Segflow Cheng et al. [2017] takes use of a parallel two-branch network to predict the segmentation as well as optical flow, through the bidirectional propagation between two frames, calculating optical flow and segmentation together and expecting them to benefit from each other. RGMP Wug Oh et al. [2018] takes both annotations of the first frame and predicted mask of the previous frame as guidance, employs a Siamese encoder-decoder to conduct the mask propagation as well as detection, and with synthetic data to further boost the segmentation performance. OSMN Yang et al. [2018] shares the similar design principle with RGMP, while the difference is that it uses an modulator to quickly adapt the first annotation to the previous frame, which can then be used by the segmentation network as the spatial prior.

6.2.3 Fine-tuning based Methods

Besides the aforementioned two categories of methods, there are some fine-tuning based methods which achieve the top performance in video object segmentation benchmark are OSVOS-S Maninis et al. [2017], OnVOS Voigtlaender and Leibe [2017], CINM Bao et al. [2018] etc, and all of them are derived from OSVOS Caelles et al. [2017]. Specifically, OSVOS-S Maninis et al. [2017] aims to solve the problem of removing noisy object(s) with the help of instance segmentation. While OnAVOS Voigtlaender and Leibe [2017] tries to enhance the network's ability for recognizing the new appearance of the target object(s) as well as suppressing the similar appearance carried by the noisy object(s). CINM Bao et al. [2018] is also initialized with the fine-tuning model, and employ a CNN to infer the markov random field (MRF) in spatial domain, and with optical flow to track the segmented object(s) in temporal domain.

6.2.4 Video Loss

In this paper, targeting to improve the fine-tuning methods, with OSVOS as an entry, we deliver a tiny head *video loss*. As aforementioned, our observation is based on the 'delayed' learning process for target instance(s) between the *parent network* and *online fine-tuning*. Through incorporating a basic component, i.e. video loss, we achieve better performance regarding to segmentation accuracy compared to OSVOS using exactly same backbone network structure(s). Furthermore, considering that it may not always be easy to distinguish the object(s) from background in 2D image coordinate, we further utilize metric learning and the proposed mixed instance-aware video loss to enforce the pixels, after mapping through a FCN in high-dimensional space, which belong to target object(s) or background are supposed to be closed with each other, while any two pixels with one belong to the target object(s) and the other belong to background are supposed to has a relatively far distance with each other. Through employing the proposed *video losses*, the performance has been significantly improved regarding to the segmentation accuracy, and some noisy objects have been effectively removed. Moreover, since *video loss* is a common building block, it can be generalized to all kinds of fine-tuning based methods including, but not limit to OnVOS Voigtlaender and Leibe [2017], OSVOS-S Maninis

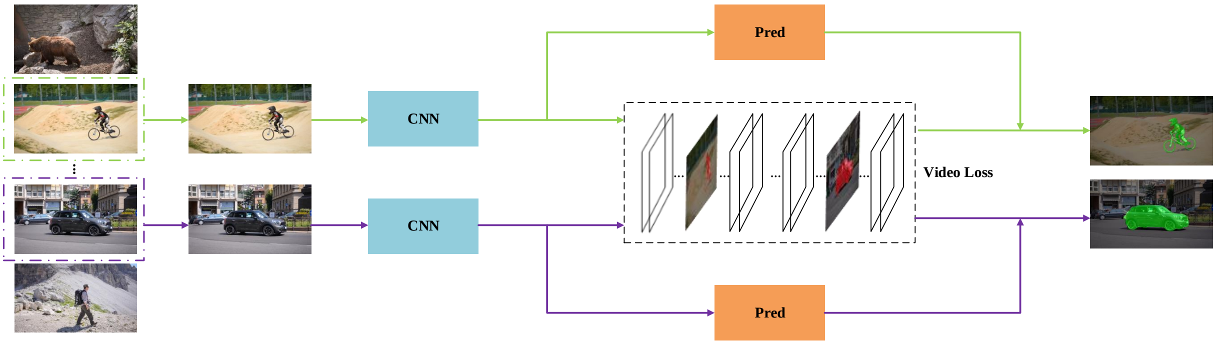


Figure 6.2: The workflow of OSVOS-VL. The first column denote images from differ video. Compared to OSVOS (without the middle block of video loss), only a tiny head, video loss block, is added.

et al. [2017], CINM Bao et al. [2018] etc.

6.3 Methodology

The motivation, design and key implementation of *video loss* will be illustrated in detail.

6.3.1 Overview

The assumption for *video loss* is that, different objects are linear separable in high-dimentional space, i.e. the feature space. Meanwhile, the euclidean distances of the features of the same object are supposed to be smaller than that of different objects. The workflow of OSVOS-VL is shown in Figure 6.2. As can be seen, *video loss* just like a light-weight head being parallel with the prediction part, thus the extra time cost is insignificant. Once the better features are obtained after training of *parent network*, it would ease the learning processing of *online fine-tuning* stage and is much prone to achieve accurate segmentation results compared to that of OSVOS. This design can be viewed as maintaining an *average target object* for each video, and expecting the objects from different videos are much more far away than that of from the same video, which effectively prevents the background noise from other objects (of no interest). We deliver three types of video loss (VL) in this paper. The first one is the two dimensional video loss (2D-VL), which make the *parent network* to push away different objects in image coordinates. The second and third are the high-dimensional video loss (HD-VL). Established on 2D-VL, the HD-VL further maps 2D features to high dimensional space, and clusters pixels which belong to the same instance together, and utilize object centers in HD-space as constraints.

6.3.2 Two Dimensional Video Loss

In OSVOS Caelles et al. [2017], considering the sample imbalance between the target object(s) and the background, weight cross entropy loss is employed to conduct the pixel-wise segmentation task. The expression of weighted cross entropy loss as follows:

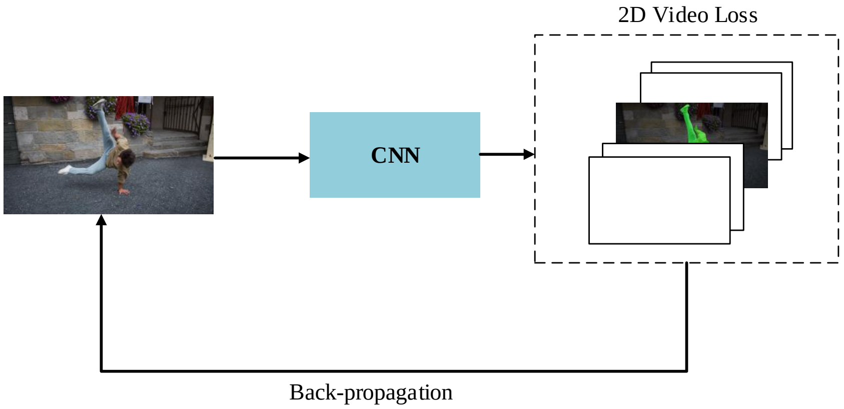


Figure 6.3: The illustration of two-dimensional video loss.

$$L_{2d} = -\frac{Y_-}{Y} \sum_{j \in Y_+} \log P\left(y_j = 1|X\right) - \left(1 - \frac{Y_-}{Y}\right) \sum_{j \in Y_-} \log P\left(y_j = 0|X\right) \tag{6.1}$$

Where X is the input image, $y_i \in 0, 1, j = 1, \ldots, |X|$ is the pixelwise binary label of X, and Y and Y_- are the positive and negative labeled pixels. $P(\cdot)$ is obtained by applying a sigmoid to the activation of the final layer. $|Y_-|/|Y|$ is employed for the purpose of training the imbalanced binary task as in Xie and Tu [2015].

OSVOS Caelles et al. [2017] only rely on weighted cross entropy loss to fine-tune *parent network*, but we argue that it will mix up all of the objectness features in DAVIS dataset, without classifying which kind of objects the foreground belongs to. It may make the online fine-tuning process more harder to recognize which instance the object is. Therefore, we propose 2D-VL to force the network to learn features of different instances during the training of *parent network*.

For this purpose, we add the identity of each video (v_{id}) into the training process as input. After recognizing the v_{id} of the training (image) data, the network can update each specific (video) category through back propagation. Please note, 2D-VL share the same expression with Equation 6.1, but different from the prediction branch, our 2D-VL only updates corresponding (video) category directly, as illustrated in Figure 6.3.

6.3.3 High Dimensional Video Loss

With the observation that objects in 2D dimension usually have similar appearances or shapes, which brings too much confusion to the network to distinguish an object accurately, we propose to map the prediction to a high-dimensional space firstly, and expecting that after mapping, the distance among different objects is enlarged. The mapping process in high dimensional space is shown in Figure. 6.4. In HD space, we look forward to clustering embeddings from different objects into different groups.

In PML Chen et al. [2018b], a modified triplet loss is utlized to pull samples with same identity close to each other, and only constrain the smallest negative points and smallest positive points. Inspried but different from that, we randomly sample 256 points in both foreground parts and background

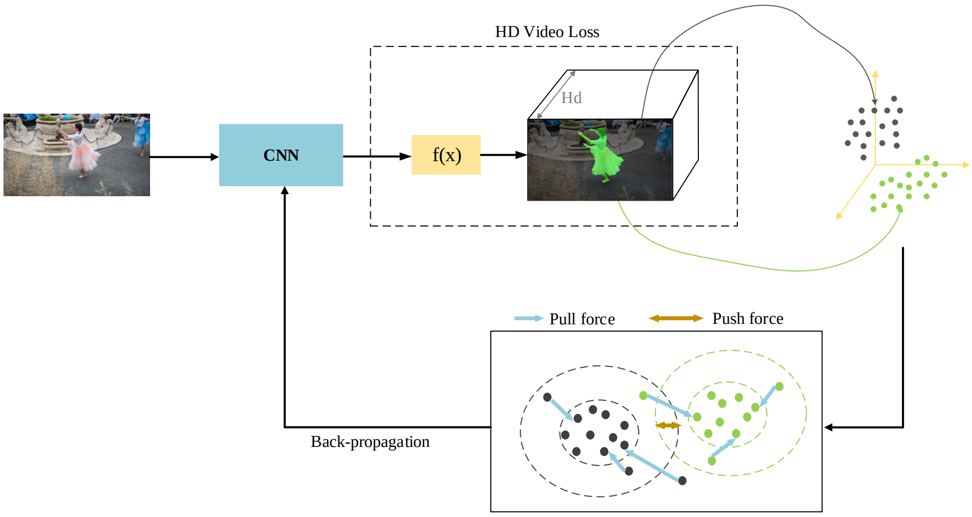


Figure 6.4: The illustration of high-dimensional video loss.

parts, pulling points from the same part together and push points from different parts away. The triplet loss is defined as

$$L_{hd\_tl} = y \left\| f\left(x_1\right) - f\left(x_2\right) \right\| + (1-y)\, max\left(0, \quad \lambda - \left\| f\left(x_1\right) - f\left(x_2\right) \right\|\right) \tag{6.2}$$

where $y = 1$ when point x_1, x_2 belong to a same cluster (foreground or background) and $y = 0$ when point x_1, x_2 belong to different clusters.

6.3.4 Mixed Instance-aware Video Loss

Contrastive Center Loss Inspired by the work De Brabandere et al. [2017], we define a contrastive center loss for the purpose of pulling embeddings with the same label close to each other while pushing embeddings with different labels far away from each other. To restrict the entire foreground area and background area, we first calculate the center point of each part. Then we use the contrastive center loss function to penalize the distance between these two center points. The motivation behind this is to restrict distribution of foreground embeddings and reduce the amount of computation.

$$L_{hd\_cl} = max\left(0, \quad \lambda - \left\| \mu_{+} - \mu_{-} \right\|\right) \tag{6.3}$$

where μ_+ represents the center point of foreground cluster, and μ_- represents the center point of background cluster, both in high-dimensional space.

Mixed Loss Contrastive center loss is a loss restricting the overall distribution of examples, while triplet loss considers the constraints in pixel level. In order to combine two types of constraints, here we define a mixed loss as

$$L_{hd\_mix} = \beta_1 L_{hd\_cl} + \beta_2 L_{hd\_tl} \tag{6.4}$$

Where β_1 and β_2 are the coefficients for balancing two loss terms.

Method	Parent Network	Finetuning	Backbone
OSVOS	**52.5**	75.0	VGG16
OSVOS-V2d	50.8	**76.2**	VGG16
OSVOS	53.1	65.7	MobileNet
OSVOS-V2d	**54.1**	**66.2**	MobileNet

Table 6.1: J Mean of OSVOS and OSVOS-V2d with different backbone network structures.

Method	Parent Network	Finetuning
OSVOS	53.1	65.7
OSVOS-V2d	54.1	66.2
OSVOS-Vhd	53.7	66.9
OSVOS-Vmixed	**58.6**	**67.5**

Table 6.2: J Mean of OSVOS, OSVOS-V2d, OSVOS-Vhd, OSVOS-Vmixed. With MobileNet as backbone, and 20 is dimensioins for the embedding of OSVOS-Vhd and OSVOS-Vmixed.

6.3.5 Training

In order to form a fair comparison with OSVOS Caelles et al. [2017], we adopt the same settings for the training of *parent network* and *online fine-tuning* except the training epochs. Specifically, SGD solver with momentum 0.9 is used, learning rate is 1e-8, the weight decay is 5e-4. Batch size is 1 for VGG16 based experiments and is 2 for MobileNet Howard et al. [2017] based experiments, respectively. For training the *parent network*, fine-tuning of 240 epochs is conducted based on the initialization of ImageNet Deng et al. [2009] features. For online fine-tuning, 10k iterations of fine-tuning is applied for all of the experiments for the fair comparison.

6.4 Experimental Results

6.4.1 Dataset

DAVIS-2016 Perazzi et al. [2016] is the most widely used dataset for video object segmentation, which is composed of 50 videos with pixel-wise annotations for single-object. Among them, 30 video sequences are chosen as training set, and the other 20 video sequences are utilized as test set.

6.4.2 Quantitative Results

In Table 6.1, J Mean for both *parent network* and *online fine-tuning* with different structures as backbones are listed out. As can be seen, for both of the experiments which based on VGG16 and MobileNet, OSVOS+ video loss achieve the better performance during *onlie fine-tuning* phase, while with comparable performance with OSVOS during *parent network* training phase, which proves our assumption that video loss, as a common module, is effective in helping the (FCN) network to recognize the target instance. In Table 6.2, compared to OSVOS with video loss utilized in 2D (OSVOS-V2d), OSVOS with high-dimensional loss (OSVOS-Vhd, OSVOS-Vmixed) performs better, which is matched with our observation that sometimes it is much more easier for similar features to

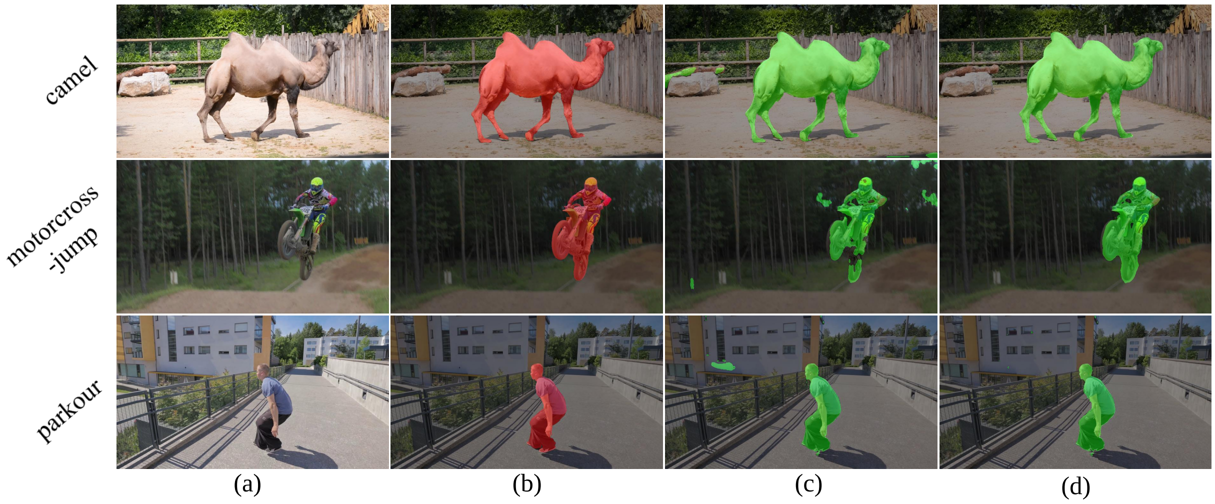


Figure 6.5: Visualization results of OSVOS and OSVOS-VL. (a) Input (b) Ground Truth (c) Segmentation from OSVOS (d) Segmentation from OSVOS-V2d.

be distinguished in high-dimensional space than that of in two-dimensional space. Please note, all of our experiments trained with 10k iterations and without any post-processing for the purpose of fair comparison and saving training time, which is slightly different from Caelles et al. [2017], and the preliminary experiments we tested show that as the training iterations increasing (around 20k iterations), which can replicate the numbers that the paper Caelles et al. [2017] report.

6.4.3 Qualitative Results

We also provide some visualized comparisons in Figure 6.5 between OSVOS and OSVOS-V2d, as can be seen, among the results acquired by OSVOS, wrong segments are accompanied in the sourrondings of the target object, we suspect that is because only rely on prediction loss in OSVOS can not distinguish instance information between the target object and background (noisy) objects. In contrast, OSVOS-VL effectively remove the noisy parts compard to that of OSVOS.

6.4.4 Performance on per sequence

In order to have a better understanding of the work principle of the proposed video loss block, we illustrate performance comparsion of OSVOS and OSVOS-V2d on per sequence, as shown in Table 6.3, for both VGG16 and MobileNet based experiments, in 12 out of 20 sequences, OSVOS-V2d achieve better performance than OSVOS, in some sequences such as *bmx-trees*, *camel*, *dance-twirl*, *dog*, *drift-chicane*, *drift-straight*, *motocross-jump*, *paragliding-launch*, OSVOS-V2d achieves consistent improvements on both backbones, and these sequences usually contain abrupt motions or noisy objects which share the similar appearance with the target object.

Sequence	VGG+OSVOS	VGG+OSVOS-V2d	MN+OSVOS	MN+OSVOS-V2d
Blackswan	**94.1**	93.6	93.3	**94.0**
bmx-trees	52.8	**58.5**	42.0	**42.8**
breakdance	67.6	**67.7**	**75.1**	70.0
camel	83.7	**85.8**	70.2	**75.1**
car-roundabout	**88.3**	75.6	**83.0**	74.2
car-shadow	**88.6**	83.5	**74.3**	74.2
cows	**95.2**	94.9	**90.7**	87.9
dance-twirl	60.7	**64.9**	63.4	**66.2**
dog	72.6	**88.3**	88.7	**90.8**
drift-chicane	61.3	**73.9**	26.9	**36.0**
drift-straight	56.4	**61.8**	35.5	**35.7**
goat	**88.1**	87.9	82.0	**85.5**
horsejump-high	**84.5**	81.5	**69.8**	69.1
kite-surf	**75.3**	73.9	54.9	**55.3**
libby	75.4	**77.2**	**69.4**	68.3
motocross-jump	60.2	**67.3**	49.6	**52.7**
paragliding-launch	63.9	**64.0**	56.3	**58.5**
parkour	89.0	**89.2**	**81.5**	73.6
scooter-black	**58.2**	35.4	57.6	**62.3**
soapbox	84.3	**86.1**	**49.5**	46.3

Table 6.3: J Mean of OSVOS and OSVOS-V2d on per sequence. VGG denotes VGG16 and MN denotes MobileNet as the feature extractor.

6.5 Conclusion

In this paper, we deliver a common module, video loss, for video object segmentation, which is tailored to overcome the limitation of fine-tuning methods, during the phase of training *parent network*, dilute the instance information, hence delay the overall training process. Considering in CNN, the shallow layers usually contain much rich details of object(s) which are the key cues to specify different instances, while the deeper layers have more stronger generalization ability to recognize generic objects. Various video losses are proposed as the constraints to supervise the training process of *parent network*, which is effective in removing the noisy objects. Once the training process is finished, the *parent network* is well prepared to adapt to the instance quickly during *online fine-tuning*. One of our future interests will be extending the video loss into other fine-tuning methods such OSVOS-S, OnVOS. Another one will be with the help of the network search technique to automatically decide the training epochs and learning rate.

Bibliography

L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018a.

Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018b.

J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7415–7424, 2018.

B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

Y. Liu, L. Liu, H. Rezatofighi, T.-T. Do, Q. Shi, and I. Reid. Learning pairwise relationship for multi-object detection in crowded scenes. *arXiv preprint arXiv:1901.03796*, 2019.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. *arXiv preprint arXiv:1807.09190*, 2018.

K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *arXiv preprint arXiv:1709.06031*, 2017.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

J. Sun, D. Yu, Y. Li, and C. Wang. Mask propagation network for video object segmentation. *arXiv preprint arXiv:1810.10289*, 2018.

P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. *arXiv preprint arXiv:1812.05050*, 2018.

S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018.

S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

Chapter 7

RGBD Based Dimensional Decomposition Residual Network for 3D Semantic Scene Completion

RGB images differentiate from depth as they carry more details about the color and texture information, which can be utilized as a vital complement to depth for boosting the performance of 3D semantic scene completion (SSC). SSC is composed of 3D shape completion (SC) and semantic scene labeling while most of the existing approaches use depth as the sole input which causes the performance bottleneck. Moreover, the state-of-the-art methods employ 3D CNNs which have cumbersome networks and tremendous parameters. We introduce a light-weight Dimensional Decomposition Residual network (DDR) for 3D dense prediction tasks. The novel factorized convolution layer is effective for reducing the network parameters, and the proposed multi-scale fusion mechanism for depth and color image can improve the completion and segmentation accuracy simultaneously. Our method demonstrates excellent performance on two public datasets. Compared with the latest method SSCNet, we achieve 5.9% gains in SC-IoU and 5.7% gains in SSC-IOU, albeit with only 21% network parameters and 16.6% FLOPs employed compared with that of SSCNet.

7.1 Introduction

We live in a 3D world where everything occupies part of the physical space under the view of the human perception system. Similarly, 3D scene understanding is of importance since it is a reflection about the real-world scenario. As one of the most vital fields in 3D scene understanding, Semantic Scene Completion (SSC) has verity of applications, including robotic navigation Gupta et al. [2013], scene reconstruction Hays and Efros [2007], auto-driving Laugier et al. [2011] *etc.* However, due to the dimensional curse brought by 3D representation Wang and Yang [2010] and the limited annotation datasets, the research field of SSC still step slowly in the past decades.

With the renaissance of deep learning Gong et al. [2017]; Krizhevsky et al. [2012]; Yan et al. [2019] and a few large-scale datasets being made available Deng et al. [2009]; Lin et al. [2014]; Song et al. [2017] in recent years. The research activities of 3D shape processing thrive again in the computer vision community, injecting new possibilities and objectives for SSC as well introducing some unprecedented challenges.

Conventional methods usually utilize the hand-crafted features, such as voxel Kim et al. [2013] and TSDF Izadi et al. [2011] to represent the 3D object shape, and make use of the graph model to infer the scene occupations and semantic labeling individually Gupta et al. [2015]; Kim et al. [2013]. The current state-of-the-art technique SSCNet Song et al. [2017], instead uses an end-to-end 3D network to conduct the scene completion and category labeling simultaneously. Through combining the semantic and geometrical information implicitly via the network learning process, the two individual tasks can benefit from each other.

Though remarkable gains in terms of scene completion and labeling accuracy have been achieved, the massive amount of parameters brought by the 3D representation make it computing-intensive. Moreover, another problem suffered in the existing SSC is the low-resolution representation Guo and Tong [2018]; Song et al. [2017]. In particular, due to the limitation of computation resources, both of the conventional and deep learning based methods sacrifice high-resolution to compromise an acceptable speed.

On the other hand, most of existing methods solely use depth as input, which is struggled to differentiate objects from various categories. For example, a paper and a tablecloth on the same table can be easily distinguished by color or texture information. To sum up, depth and color image are different modalities captured by the sensor, they all provide us with what the scene looks like. The former gives us more sense about the object shape and distance, while the later transfers more information about the object texture and saliency. It is proved that both of the two modalities are helpful to boost the performance of SSC task Garbade et al. [2018], although how to fuse them is still an unsolved problem.

To overcome the problems as mentioned above, we propose a light-weight semantic scene completion network, which utilizes both of the depth and RGB information. It formulates the 3D scene completion and labeling as a joint task and learns in an end-to-end way. The main contributions of this paper are three-fold:

- Firstly, we propose the dimensional decomposition residual (DDR) blocks for 3D convolution, which dramatically reduces the model parameters without performance degradation.
- Secondly, 3D feature maps of RGB and depth are fused in multi-scale seamlessly, which enhances the network representation ability and boost the performance of SC and SSC tasks.
- Thirdly, the proposed end-to-end training network achieves state-of-the-art performance on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets.

7.2 Related Works

7.2.1 3D Scene Completion and Semantic Labeling

As an important branch in 3D scene understanding, semantic scene completion (SSC) has many real-world applications and has received increasing attention recently with the support of deep learning Krizhevsky et al. [2012] and the large-scale annotated dataset Song et al. [2017].

SSCNet Song et al. [2017] is the first one which formulates the shape completion and semantic labeling as a joint task and learns the task in an end-to-end way. TS3D Garbade et al. [2018] is based on SSCNet, and utilizes an additional network to incorporate the color information into the learning loop. Both of SSCNet and TS3D adopt truncated signed distance function (TSDF Izadi et al. [2011]) to encode the 3D volume, where every voxel stores the distance value d to its closest surface, and the sign of the value indicates whether the voxel is in free space or occluded. However, TSDF is computationally intensive since it requires the calculation of the distance between the points on the surface and each point belong to the objects. Although with remarkable performance achieved, the 3D convolution representation results in a network that is computationally expensive with highly redundant parameters.

7.2.2 Computation-efficient Networks

As a milestone in deep learning architectures, ResNet He et al. [2016] uses a residual block to prevent the performance degradation that occurs when network layers become deep. The extreme deep network leads to the state-of-the-art performance in many tasks including image classification Krizhevsky et al. [2012], object detection Liu et al. [2019]; Redmon et al. [2016]; Ren et al. [2017] and segmentation Chen et al. [2018a]; He et al. [2017]. However, this is very expensive concerning computation resource and heavy-burden He et al. [2016]; Krizhevsky et al. [2012]. To cater to the appeal for real-world applications, there is a trend to tailor the heavy-burden networks to the light-weight network in recent years.

Feature Representation Considering the redundant information contained in the 3D scene completion, the first spectrum of work try to model the scene with sparse feature representation. Specifically, OctNet Riegler et al. [2017] and O-CNN Wang et al. [2017] utilize the Octree-based CNN to represent the 3D object shape. PointNet Charles et al. [2017] and Kd-Networks Klokov and Lempitsky [2017] employ point clouds to indicate the occupation of the scene. Although saving the memory and computation, the neighbor pixels are usually mapped to the same voxel, which inevitably causes the detail missing for semantic labeling and scene understanding.

Group Convolution In recent two years, there are several popular light-weight networks have been proposed, include MobileNet Howard et al. [2017]; Sandler et al. [2018] and ShuffleNet Zhang et al. [2018b]. In MobileNet, depth-wise convolutions and point-wise convolutions are utilized to separate the channels as well as reduce the parameters and the calculations. In ShuffleNet, besides the group point-wise convolution and depth-wise convolution are adopted, shuffle layer is developed for information exchange between different shuffle units. However, the above models heavily rely on depth-wise convolution and group convolution, and mainly target at 2D networks thus can not directly be applied for 3D tasks.

Spatial Group Convolution To improve the computing efficiency of the 3D network. EsscNet Zhang et al. [2018a] is introduced, rather than to conduct the group convolution on feature channel dimension, which adopts the group convolution on the spatial aspect. The drawback of spatial group convolution is that it splits the features manually into separate parts, which cause the performance drops. Meanwhile, the splitting process involves hash table maintaining and coordinate with other blocks, and is cumbersome for transplantation. On the contrary, the proposed DDR block is much

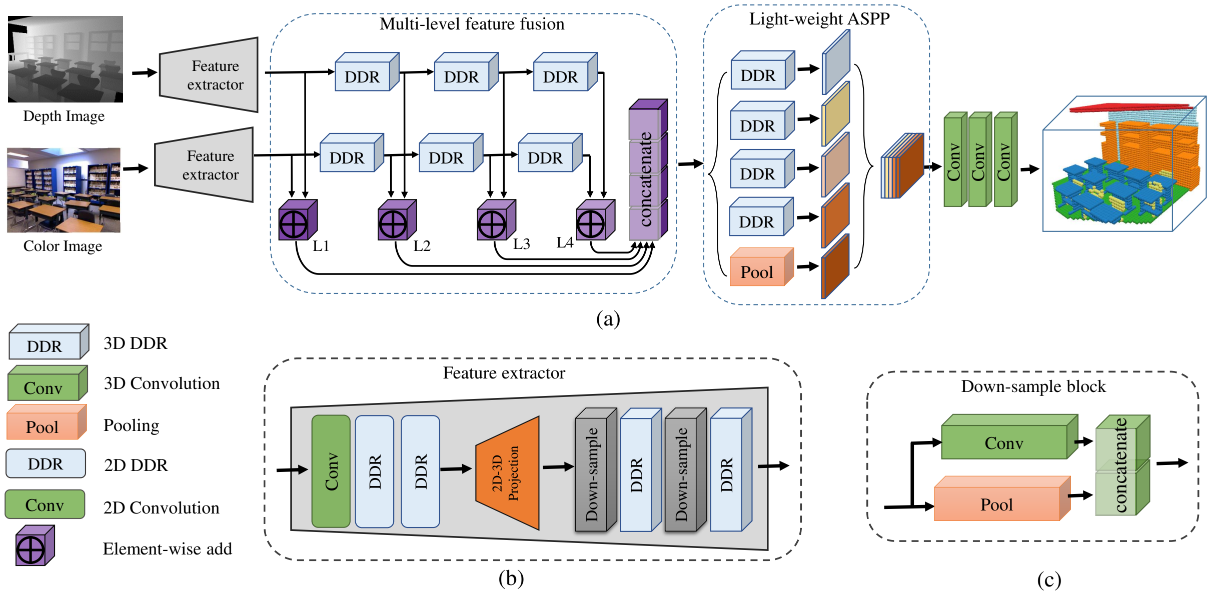


Figure 7.1: (a) Network architecture for semantic scene completion. Taking RGBD image as input, the network predicts occupancies and object labels simultaneously. (b) Detailed structure of the feature extractor. (c) Structure of the down-sample block.

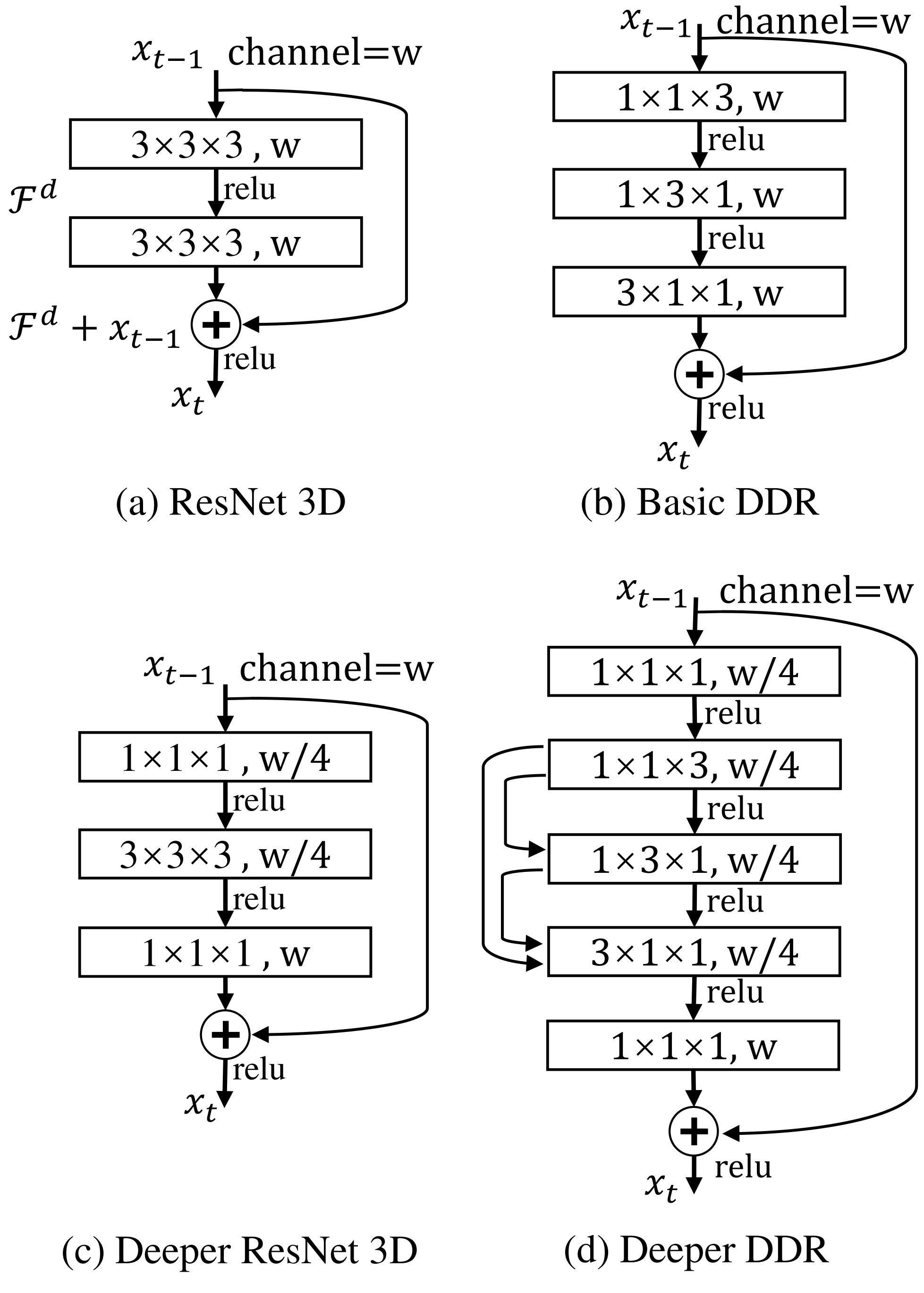
flexible, and it can be planted to any network which contains the 3D modules.

7.2.3 Modality Fusion in SSC

There are many works focused on RGBD fusion in 2D applications Chang et al. [2017]; Gupta et al. [2015]; Park et al. [2017]; Qi et al. [2017]; Wang et al. [2016]. RGBD sensor can capture the depth and color images simultaneously, although depth can be used to infer the geometry of the scene, which is too sparse to reconstruct the occluded parts of the scene. Compared with depth, color image carries more cues about texture, color, and reflections, which can be viewed as an essential complement to the depth for SSC task. Following the design philosophy of SSCNet, TS3D Garbade et al. [2018] adds the color image into the work-flow. However, the scene labeling needs to be estimated twice, and the depth flow and color flow are still apart from each other from the essential.

In Guedes et al. [2018], two fusion strategies were proposed, one is early-stage fusion which concatenates the feature at the first layer, and another is mid-level fusion which concatenates the features before the output layer. Although follow the overall design and reuse the features of SSCNet, the performance of adopting both fusion strategies are unexpectedly worse than that of SSCNet.

The most related work for feature fusion is RDFNet Park et al. [2017], which utilizes multi-scale fused features from color images, and aims to build a 2D segmentation framework. However, fusing the features in the 3D network is much more challenging as mentioned before. In this paper, we propose a novel fusion strategy which effectively fuses the 3D depth and color features on multi-scales without bringing in extra parameters.



(a) ResNet 3D

(b) Basic DDR

(c) Deeper ResNet 3D

(d) Deeper DDR

Figure 7.2: Residual blocks and the proposed DDR blocks.

7.3 Methodology

7.3.1 Overview

This section presents the proposed light-weight network for SSC. The computation-efficient Dimensional Decomposition Residual (DDR) block, as well as a novel modality fusion module, are emphasized. On the one hand, through dimensional splitting on 3D convolutions and dense connection, using DDR blocks can dramatically reduce the network parameters. On the other hand, through fusing the 3D features of depth and color image seamlessly, the proposed network can efficiently make use of the information captured by the RGBD sensors, and various modulates of inputs complement with each other thus boost the performance of shape completion and scene labeling simultaneously. The framework of the proposed network is shown in Figure 7.1. The network has two feature extractors, which take a full resolution depth and the corresponding color image as inputs, respectively. The network first uses 2D DDR blocks to learn the local textures and the geometry representation. Then, the 2D feature maps are projected to 3D space by a projection layer. A multi-level fusion strategy is then applied to fuse the texture and geometry information. After that, the network responses are then concatenated and fed into the subsequent light-weight Atrous Spatial Pyramid Pooling (ASPP) module to aggregate information in multiple scales. In the end, another three pointwise convolutional layers are used to predict the final voxel labels. The following parts will explain the design details of each module.

7.3.2 Dimensional Decomposition Residual Blocks

Basic DDR

Residual layers He et al. [2016] have the property of allowing convolutional layers to approximate residual functions,

$$x_t = \mathcal{F}^d\left(x_{t-1}, \{W_i\}\right) + x_{t-1} \tag{7.1}$$

where x_{t-1} and x_t are the input and output.. The function $\mathcal{F}^d\left(x_{t-1}, \{W_i\}\right)$ represents the residual mapping to be learned and d is the dilation rate within the block. This residual formulation facilitates learning and alleviates the degradation problem present in architectures that stack a large number of layers Romera et al. [2018].

Directly applying the original (2D) ResNet block into the 3D dense prediction task, the two corresponding 3D residual layers will be: the non-bottleneck design with two-layer $3 \times 3 \times 3$ convolutions as described in Figure 7.2(a), and the three-layer bottleneck version as depicted in Figure 7.2(c).

However, both of the two structures will suffer the problem of high computational costs as the network parameters grow in cubic. We propose to redesign the residual through decomposing the 3D convolution into three consecutive layers along each dimension. The proposed basic DDR block is shown in Figure 7.2(b) and its deeper bottleneck version is shown in Figure 7.2(d). In this way, the network can reduce parameters and capable of capturing 3D geometric information according to the theory in Szegedy et al. [2016].

Here we provide an episode to illustrate the effectiveness of DDR block for reducing network parameters: Considering a 3D CNN with input channels c^{in}, output channels c^{out}, and kernel size of

$k^x \times k^y \times k^z$. Without losing the generality, we can assume $k^x = k^y = k^z = k$. The original block in 3D CNN is then be decomposed into three consecutive layers with filter size $1 \times 1 \times k$, $1 \times k \times 1$ and $k \times 1 \times 1$, accordingly. The computational costs of the original block and DDR block are proportional to $c^{in} \times c^{out} \times k \times k \times k$ and $c^{in} \times c^{out} \times (k+k+k)$, respectively. The advantage of DDR for reducing network parameters will be enlarged when k become large, since $3k \ll k^3$. As an example, the parameters of a typical 3D convolutional layer with a $3 \times 3 \times 3$ kernel will drops to 1/3 after adopting DDR block.

Deeper DDR

Inspired by the bottleneck design He et al. [2016], we further deliver a deeper DDR block. In specific, for each residual function, a $1 \times 1 \times 1$ layer is added at both the top and bottom of the Dimensional Decomposition convolutions. The $1 \times 1 \times 1$ layers are responsible for reducing and restoring the dimensions, which make the three Dimensional Decomposition convolutional layers form a bottleneck with smaller input/output dimensions.

Moreover, parameter-free identity shortcuts are added within each dimensional decomposition convolution. The dense identity connections are not only helpful for robust feature representation but also have the merits of alleviating the vanishing-gradient problem and strengthen the feature propagation He et al. [2016]. In the remainder of the paper, DDR refers to the deeper DDR block unless specifically noted.

7.3.3 Two Modality Multi-level Feature Fusion

Feature Extractor Module

In our network, there are two parallel branches for feature extraction corresponding to the depth and color image. As shown in Figure 7.1(b), the feature extractor module is composed of three components: a 2D feature extractor, a 3D feature extractor and a projection layer which mapping the 2D feature to the 3D feature. The network first utilizes 2D feature extractor to learn local color and texture representation. After feature mapping to the 3D space by a projection layer, 3D feature extractor is employed to acquire the geometry and context information.

2D Feature Extractor To extract features from a 2D depth and color image, a 2D point-wise convolution is firstly used to increase the channels of feature maps. Then two 2D DDR blocks are stacked for residual learning. Through the process, the resolution of the output feature map keeps the same as the input image. Please note, in our network, the number of parameters for 2D DDR blocks is 192, which is insignificant when compared with the 195k parameters for 3D DDR blocks. Therefore, we mainly focus on the light-weight operations of 3D DDR blocks.

Projection Layer Since each pixel in the depth map corresponding to a tensor in the 2D feature map, every feature tensor can be projected into the 3D volume at the location with the same depth value. This step yields an incomplete 3D volume that assigns to every surface voxel its corresponding feature tensor. For the voxels that are not occupied by any depth values, their feature vectors are set to zeros. The mapping index $\mathcal{T}_{u,v}$ at (u, v) can be computed using the depth value $I_{u,v}$ and camera pose C which are provided along with each image. Because the feature volume resolution is lower than the feature map resolution, several neighboring features will be projected into the same voxel,

	scene completion			semantic scene completion											
Methods	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
Lin *et al.* Lin et al. [2013]	58.5	49.9	36.4	0.0	11.7	13.3	**14.1**	9.4	29.0	24.0	6.0	7.0	16.2	1.1	12.0
Geiger *et al.* Geiger and Wang [2015]	65.7	58.0	44.4	10.2	62.5	19.1	5.8	8.5	40.6	27.7	7.0	6.0	22.6	5.9	19.6
SSCNet Song et al. [2017]	57.0	**94.5**	55.1	15.1	**94.7**	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
EsscNet Zhang et al. [2018a]	**71.9**	71.9	56.2	17.5	75.4	25.8	6.7	**15.3**	**53.8**	**42.4**	11.2	0	33.4	11.8	26.7
ours	71.5	80.8	**61.0**	**21.1**	92.2	**33.5**	6.8	14.8	48.3	42.3	**13.2**	**13.9**	**35.3**	**13.2**	**30.4**

Table 7.1: Results on the NYU dataset. Bold numbers represent the best scores.

	scene completion			semantic scene completion											
Methods	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
Zheng *et al.* Zheng et al. [2013]	60.1	46.7	34.6	-	-	-	-	-	-	-	-	-	-	-	-
Firman *et al.* Firman et al. [2016]	66.5	69.7	50.8	-	-	-	-	-	-	-	-	-	-	-	-
SSCNet Song et al. [2017]	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	33.9	57.0	**59.5**	28.3	8.1	**44.8**	25.1	40.0
TS3D Garbade et al. [2018]	80.2	91.0	74.2	33.8	**92.9**	46.8	**27.0**	27.9	**61.6**	51.6	27.6	**26.9**	44.5	22.0	42.1
ours	**88.7**	88.5	**79.4**	**54.1**	91.5	**56.4**	14.9	**37.0**	55.7	51.0	**28.8**	9.2	44.1	**27.8**	**42.8**

Table 7.2: Results on the NYUCAD dataset. Bold numbers represent the best scores.

and we use max-pooling to simulate this step. With the feature projection layer, the 2D feature maps extracted by the 2D CNN are converted to a view-independent 3D feature volume. During training, the mapping indexes $\mathcal{T}$ between feature map tensors and voxels are recorded in a table for gradient back-propagation.

3D Feature Extractor After the feature projection layer, a view-independent 3D feature volume is acquired. In this step, we further extract features using two 3D DDR blocks. A down-sample block is added in front of each DDR blocks to reduce the size of the feature maps and increase the dimension of its channel. Figure 7.1(c) shows the structure of the down-sample block. A pooling layer and a pointwise convolution layer are concatenated to increase the channels of the output feature map of the down-sample block.

Multi-level Feature Fusion

One primary challenge of 3D RGBD based semantic segmentation is how to effectively extract the color features along with depth features and to utilize those features for the labeling. To fully use the multi-modal features, we propose a novel feature fusion strategy which is inspired by Lin et al. [2017]; Park et al. [2017]. We employ multi-modal CNN feature fusion while preserving the lower computational cost. In specific, different levels of features are extracted through multiple DDR modules, and then these features are merged together by element-wise add. The reason for using element-wise add rather than other operations is because it can fuse the features neatly with insignificant computation costs.

Through the cascaded DDR blocks, both low-level features and high-level features are captured, which enhance the representation ability of the network and is beneficial for the performance of semantic scene completion task.

Methods	Params/k	FLOPs/G	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
Ours-Depth	155.0	20.6	30.6	93.0	28.6	6.7	13.6	60.3	20.0	12.3	0.	30.9	12.0	28.9
Ours-RGB	155.0	20.6	19.3	91.8	30.5	3.7	13.1	44.4	37.1	10.6	5.5	31.0	11.9	27.2
Ours-RGBD	195.0	27.2	21.1	92.2	33.5	6.8	14.8	48.3	42.3	13.2	13.9	35.3	13.2	30.4

Table 7.3: Ablation experiments of RGB and Depth fusion.

7.3.4 Light-weight ASPP Module

Different object categories have various physical 3D sizes in indoor scenes. This requires the network to capture information at different scales in order to recognize the objects reliably. Atrous spatial pyramid pooling (ASPP) Chen et al. [2018a,b] exploits multi-scale features by employing multiple parallel filters with different dilatation rates and has been proved to be powerful to improve the CNN's ability to handle objects with various sizes. However, directly applying ASPP in 3D semantic scene completion would bring in tremendous parameters as well as large computations.

Based on this consideration, we introduce a light-weight ASPP (LW-ASPP) which is capable of handling scale variability while requiring fewer computations. In specific, LW-ASPP uses multiple parallel DDR blocks with different sampling (dilation) rates. The dilated DDR is implemented by setting a dilation rate in the three-dimensional decomposition convolutions within the DDR block. The dilated DDR explicitly adjusts the field-of-view of filters as well as controls the resolution of the feature responses. The features extracted from different sampling rates are further concatenated and fused to generate the final result with the output layer, which is constructed by the three 3D point-wise convolution layers as shown in Figure 7.1.

7.3.5 Training and Loss

Training Given the training dataset (*i.e.* the RGBD images and ground truth volumetric object labels of 3D scenes), our method can be trained end-to-end. SSCNet Song et al. [2017] sets a small value (0.05) as the weight of the voxels in free space for data balancing in the training process. We adopt the same strategy in our early training process. With each additional 50 training epochs, the weight of empty voxels is gradually doubled until it is set to be the same as the other occupied voxels. All the experiments are conducted using the pyTorch framework on GPU. Our model is trained using the SGD optimizer with a momentum of 0.9, weight decay of 10^{-4} and batch size is 2, the initial learning rate is 0.01 and divided by a factor of 10 when the training loss changes less than 1e-4 within 5 consecutive epochs.

Loss For training the network, we employ the softmax cross entropy loss on the unnormalized network outputs y:

$$\mathcal{L} = -\sum_{c=1}^{N} w_c \hat{y}_{i,c} \log \left(\frac{e^{y_{ic}}}{\sum_{c'}^{N} e^{y_{ic'}}} \right) \tag{7.2}$$

where $\hat{y}_{i,c}$ are the binary ground truth vectors, *i.e.* $\hat{y}_{i,c} = 1$ if voxel i is labeled by class c, N is the number of classes, and w_c is the loss weight. To compute the loss function, we remove all voxels outside the field of view and the room and include all non-empty voxels plus occluded voxels.

7.4 Experiments

In this section, we evaluate and compare the proposed method with the state-of-the-art approaches on two public datasets, *i.e.* NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016]. Both the quantitative and qualitative results demonstrate the superiority of our algorithm on SSC task.

7.4.1 Dataset and Metrics

Dataset We evaluate the proposed method on the NYUv2 dataset Silberman et al. [2012], which is in the following denoted as NYU. NYU consists of 1449 indoor scenes that are captured via a Kinect sensor. Following SSCNet Song et al. [2017], we use the 3D annotated labels provided by Rock et al. [2015] for semantic scene completion task. NYUCAD Firman et al. [2016] uses the depth maps generated from the projections of the 3D annotations to reduce the misalignment of depths and the annotations. We compare our method with the state-of-the-art methods on both NYU and NYUCAD datasets.

Metrics As the evaluation metric, the voxel-level intersection over union (IoU) between the predicted voxel label and ground truth label is used. For the task of semantic scene completion, we evaluate the IoU of each object classes on both the observed and occluded voxels. For the task of scene completion, we treat all non-empty object class as one category and evaluate IoU of the binary predictions on the occluded voxels.

7.4.2 Comparisons with the State-of-the-art Methods

Table 7.1 shows the results on NYU dataset acquired by our method and the state-of-the-art methods. We achieve state-of-the-art performance regarding different metrics. Specifically, we achieve the best performance for both the tasks of scene completion and semantic scene completion and also rank the second best of recall and precision for scene completion. We outperform the previous SSCNet by a significant margin in overall performance, that are 5.7% gains in semantic scene completion and 5.9% gains in scene completion. The proposed network demonstrate the superior performance in some categories such as *ceil.*, *table*, *tvs*, *furn. etc*. We inspect this improvement due to the novel architecture, that makes use of the robust features from multi-level and multi-modalities, and data fusion, which effectively complement the details from the color image to these textureless objects.

To validate the robustness and generalization of the proposed network, we also conduct experiments on NYUCAD dataset as shown in Table 7.2. The comparison results with the state-of-the-art methods present the same trend. Among all of the methods, we achieve the best performance for semantic scene completion and scene completion.

7.4.3 Quantitative Analysis

Since we target at a light-weight 3D network for semantic scene completion, in this section, we list the params and FLOPs of the proposed method as well as the baseline method. As shown in Table 7.4. In specific, compared with the state-of-the-art method SSCNet, the parameters in our method is 21.0% of that in SSCNet, and the FLOPs is 16.6% of that SSCNet. However, the performance of both scene completion and semantic scene completion is around 6% higher than that of SSCNet. Compared with

Methods	Params/k	FLOPs/G	SC-IoU	SSC-IoU
SSCNet Song et al. [2017]	930.0	163.8	55.1	24.7
EsscNet Zhang et al. [2018a]	-	22.0	56.2	26.7
Ours-Depth	155.0	20.6	59.0	28.9
Ours-RGBD	195.0	27.2	61.0	30.4

Table 7.4: Params, FLOPs and Performance of our approach compared with other methods.

the EsscNet Zhang et al. [2018a], depth solely is used as the input for a fair comparison, our method is computationally cheaper than EsscNet with 6% reduction in FLOPS and increased performance. For SC and SSC tasks, EsscNet reaches the accuracies of 56.2% (SC) and 26.7% (SSC), and we achieve 59.0% (SC) and 28.9% (SSC).

7.4.4 Qualitative Analysis

Figure 7.3 shows visualized results (in different scenarios) of the scene segmentation generated by the proposed method (c) and SSCNet (d), ground truth (b) are also provided as a reference. All the results are acquired on the NYUCAD validation set. As can be seen, compared with SSCNet, the scene completion results of our method is much more abundant in detail and less error-prone.

It can be easily seen that our method performs better for objects such as *furn*, *wall*, and *floor*. For example, in the second and third rows, SSC will cause some missing in the details of the wall, which is rarely happening in our algorithm. Part of the reason that our method is better for handling the texture-less and small objects we attribute which come from the incorporated color features. In row(1), our method effectively captures the detail information about the leg of the chair. In addition, compared with SSCNet, the proposed method maintains the segmentation consistency for objects with big sizes, such as the *wall* and *floor* in the row(2) and *ceiling* in the row(4). And row(3) shows a much challenging instance, *i.e.* the window, both SSCNet and our method cannot acquire satisfied results. However, our method can recognize part of the information. Row(5) and row(6) show the failure cases in our methods, specifically, in the row(5), the *fresco* on the wall has the similar texture with the stuff on the bookshelf, it thus wrongly classified into the category of the *bookshelf*. In row(6), the ground-truth *furniture* circled by the red dashed rectangle, SSCNet wrongly predicts it into the object category, and our network wrongly classifies it as a chair, which may due to the quite similar shape and color information between the furniture and the chair category. In supplementary materials, more visualized results are provided.

7.5 Ablation Study

RGB and Depth Fusion Both RGB and Depth information are important for 3D scene understanding. To verify the effectiveness of the proposed multi-level fusion strategy, we evaluate the performance of our method with only depth or RGB image as the input. As can be seen in Table 7.3, and the performance on SSC of our method for only using depth or color image as input are 28.9% and 27.2%, respectively. Since RGB images carry more details such as color and texture, which is beneficial for the semantic information, this can be seen from the results of category *tvs* and category *sofa*. However, the advantage of using depth lies on it carries more geometry information, for the objects which are

Method	Params/k	FLOPs/G	SC-IoU	SSC-IoU
Without ASPP	132.0	21.13	56.8	26.8
3D-ASPP	431.0	63.28	62.7	30.8
LW-ASPP	195.0	27.22	61.0	30.4

Table 7.5: Params, FLOPs and Performance with/without ASPP.

Method	Params (k)	FLOPs (G)	Speed (FPS)	Memory (M)	Network Depth	SSC-IoU (%)
SSCNet [34]	930.0	163.8	0.7	5305	14	24.7
Ours-3D-ResNet	1540.5	204.7	1.3	1841	28	**30.8**
Ours-DDR	**195.0**	**27.2**	**1.5**	**1829**	**44**	30.4

Table 7.6: The inference speed and GPU memory usage of our DDRNet and the 3D-ResNet based networks. All results are acquired on a GTX1080ti GPU and evaluated on the NYU[33] test set.

difficult to differentiate through color information, it is much easier to tell the difference according to their shapes. Such as *table* and *floor*. Moreover, depth is less sensitive regarding illumination changes and the dramatic color variation within the same category, which may explain for the indoor scene, the result of using depth is a bit better than that of using a color image as input.

Meanwhile, merging depth and color features in our method significantly improve the SSC performance, which proves the two-modality information can be an excellent complement to each other. And benefit from the light-weight DDR block applied in the network, the overall computations and parameters remain small.

Light-weight ASPP The effectiveness of ASPP has been verified in 2D semantic segmentation Chen et al. [2018a] task. However, the direct expansion of ASPP from 2D to 3D would bring in a massive amount of parameters as well as make the network cumbersome. Lightweight ASPP (LW-ASPP) using DDR block as the primary core, which not only effectively reduces the network parameters but also inherits the merits of ASPP for capturing multi-scale information, thus is beneficial to the 3D task.

In order to verify the validity of LW-ASPP, we design a group of experiments in which LW-ASPP was removed from the network or replaced with 3D ASSP directly extended from ASPP. As can be seen in Table 7.5, when compared with the network without ASPP module, adding LW-ASPP boosts the SC-IoU 3.2% and SSC-IoU 3.6%. When replacing LW-ASPP with 3D-ASPP, the performance can be further improved by a small margin but with the sacrifice of over two times params and around three times FLOPs.

Change of speed/memory and performance As shown in Table 7.6, DDRNet with quite a few parameters and FLOPs compared to SSCNet. DDRNet has a deeper structure, thus stronger non-linear representation ability than the 3D-ResNet version, albeit less memory cost required. Moreover, DDRNet achieves much faster speed with an insignificant performance loss.

7.6 Conclusion

This paper proposes a novel structure for handling the semantic scene completion problem. Specifically, an end-to-end light-weight Dimensional Decomposition Residual (DDR) network is delivered for scene completion and semantic scene labeling. The two contributions are the proposed factorized convolution layer and a novel two-modality fusion mechanism. The former is effective to reduce the parameters within the network, and the later can fuse the depth and color image seamlessly in multi-level, the state-of-the-art results are achieved for both SSC and SC task on two public datasets. In the future, considering to differentiate instances of the indoor scene as well as to incorporate the shuffle layer into the proposed light-weight network will be our research interests.

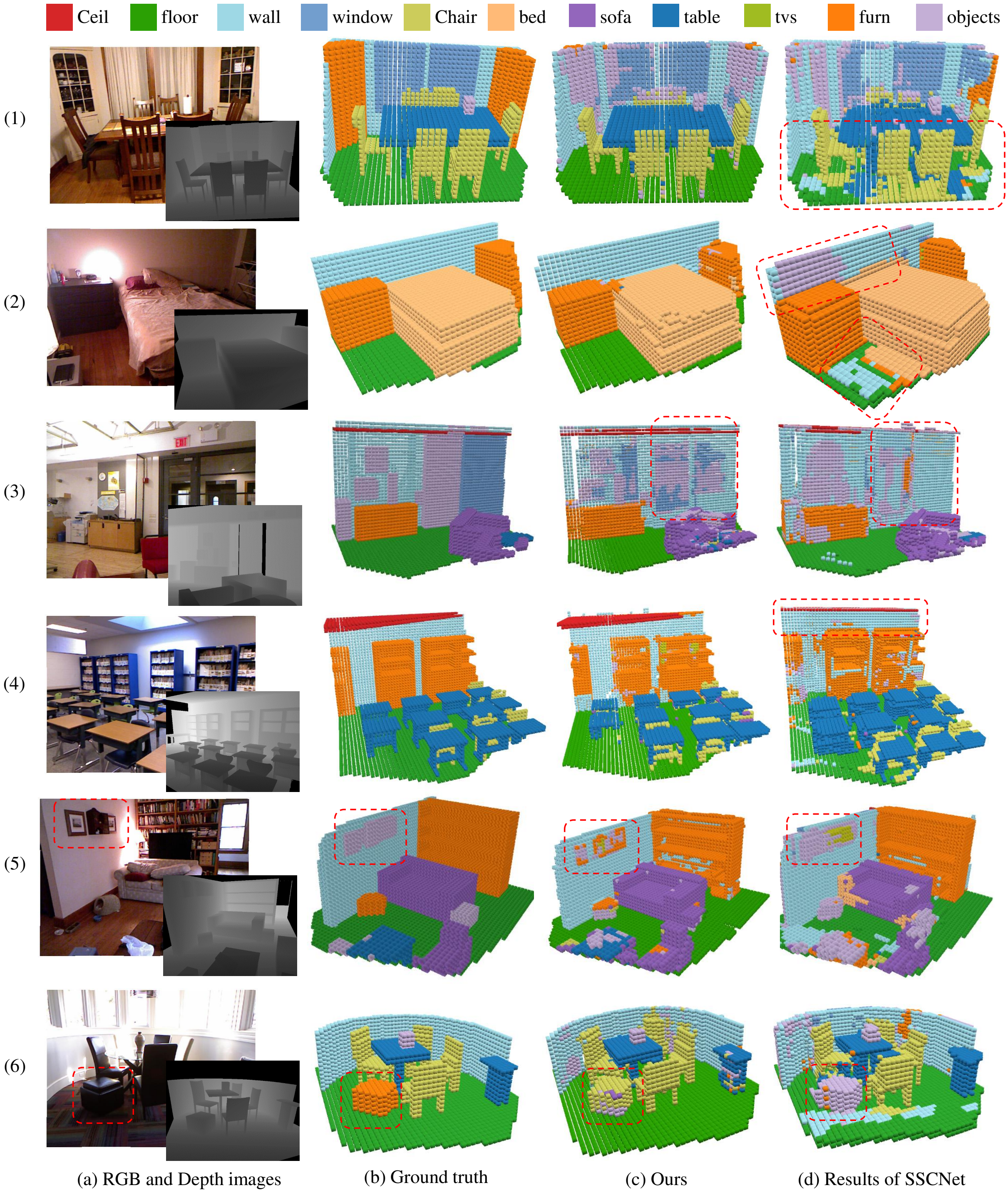


Figure 7.3: Qualitative results on NYUCAD. From left to right: Input RGB-D image, ground truth, results obtained by our approach, and results obtained by SSCNet Song et al. [2017]. Overall, our completed semantic 3D scenes are less cluttered and show a higher voxel class accuracy compared to SSCNet. Refer to Section 7.4.4 for the detailed analysis.

Bibliography

A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv:1709.06158*, 2017.

R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85, 2017.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018a.

L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018b.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, pages 5431–5440, 2016.

M. Garbade, J. Sawatzky, A. Richard, and J. Gall. Two stream 3d semantic scene completion. *arXiv:1804.03550*, 2018.

A. Geiger and C. Wang. Joint 3d object and layout inference from a single rgb-d image. In *German Conference on Pattern Recognition*, pages 183–195, 2015.

D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, pages 2319–2328, 2017.

A. B. S. Guedes, T. E. de Campos, and A. Hilton. Semantic scene completion combining colour and depth: preliminary experiments. *arXiv preprint arXiv:1802.04735*, 2018.

Y. X. Guo and X. Tong. View-volume network for semantic scene completion from a single depth image. *arXiv:1806.05361*, 2018.

S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, pages 564–571, 2013.

S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *IJCV*, 112(2):133–149, 2015.

J. Hays and A. A. Efros. Scene completion using millions of photographs. *TOG*, 26(3):4, 2007.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, pages 559–568. ACM, 2011.

B. S. Kim, P. Kohli, and S. Savarese. 3d scene understanding by voxel-crf. In *ICCV*, pages 1425–1432, 2013.

R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *ICCV*, pages 863–872, 2017.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *ITS Magazine*, 3(4):4–19, 2011.

D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, pages 1417–1424, 2013.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Y. Liu, L. Liu, H. Rezatofighi, T.-T. Do, Q. Shi, and I. Reid. Learning pairwise relationship for multi-object detection in crowded scenes. *arXiv preprint arXiv:1901.03796*, 2019.

S. J. Park, K. S. Hong, and S. Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017.

X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *CVPR*, pages 5199–5208, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, June 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2577031.

G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, volume 3, 2017.

J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *CVPR*, pages 2484–2493, 2015.

E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *T-ITS*, 19(1):263–272, 2018.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198, 2017.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

J. Wang, Z. Wang, D. Tao, S. See, and G. Wang. Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks. In *ECCV*, pages 664–679, 2016.

P. S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *TOG*, 36(4):72, 2017.

X. Wang and R. Yang. Learning 3d shape from a single facial image via non-linear manifold embedding and alignment. In *CVPR*, pages 414–421, 2010.

Q. Yan, D. Gong, and Y. Zhang. Two-stream convolutional networks for blind image quality assessment. *IEEE Transactions on Image Processing*, 28(5):2200–2211, 2019.

J. Zhang, H. Zhao, A. YaoE, Y. Chen, L. Zhang, and H. LiaoE. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, pages 733–749, 2018a.

X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018b.

B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, pages 3127–3134, 2013.

Chapter 8

Depth Based Semantic Scene Completion with Position Importance Aware Loss

Semantic Scene Completion (SSC) refers to the task of inferring the 3D semantic segmentation of a scene while simultaneously completing the 3D shapes. We propose PALNet, a novel hybrid network for SSC based on single depth. PALNet utilizes a two-stream network to extract both 2D and 3D features from multi-stages using fine-grained depth information to efficiently captures the context, as well as the geometric cues of the scene. Current methods for SSC treat all parts of the scene equally causing unnecessary attention to the interior of objects. To address this problem, we propose Position Aware Loss(PA-Loss) which is position importance aware while training the network. Specifically, PA-Loss considers Local Geometric Anisotropy to determine the importance of different positions within the scene. It is beneficial for recovering key details like the boundaries of objects and the corners of the scene. Comprehensive experiments on two benchmark datasets demonstrate the effectiveness of the proposed method and its superior performance. Code and demo[1] *are avaliable at: https://github.com/UniLauX/PALNet.*

8.1 INTRODUCTION

Semantic scene completion (SSC) Choi et al. [2015]; Gupta et al. [2015]; Song et al. [2017] is composed of scene completion and semantic labeling, which targets at inferring the completed 3D shape and the semantic categories of the occupied voxels (grid units) in the scene simultaneously. As illustrated in Figure. 8.1, given a single-view depth, SSC assigns a class label to every voxel including the surface of the scene and the occluded parts in the frustum. SSC is useful and widely applied in numerous real-world applications such as virtual reality Chen et al. [2017]; Van Krevelen and Poelman [2010], robot grasping Varley et al. [2017], automatic navigation Vineet et al. [2015], *etc.*

3D convolutional neural network (3D-CNN) is frequently used in the task of 3D scene prediction. Since 3D-CNN requires a regular grid as input, voxels are naturally chosen to represent the 3D scene

[1] Video demo can be found here: https://youtu.be/j-LAMcMh0yg

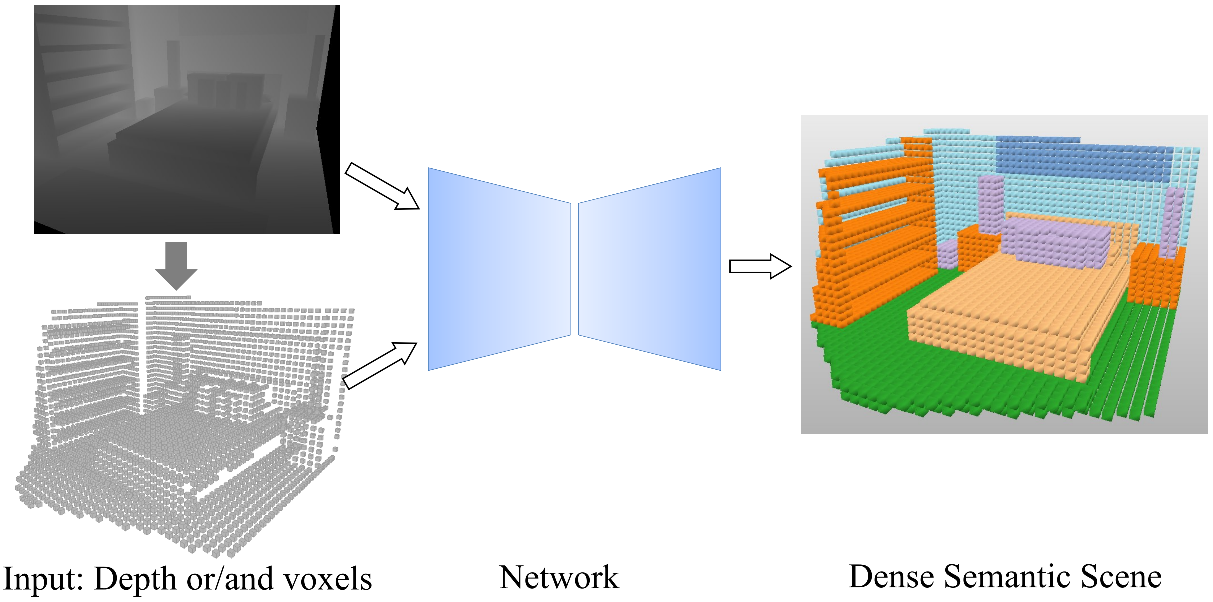


Figure 8.1: Semantic Scene Completion: Based on input depth or its corresponding 3D voxel volume, the goal of SSC is to simultaneously complete the partial 3D shapes and predict the dense semantic labels of both observed and unobserved parts in the view frustum.

in these 3D-CNN based methods, and different loss functions are employed for training them Guo and Tong [2018]; Song et al. [2017]; Tchapmi et al. [2017].

A limitation of previous methods is that they ignore the position importance of the voxels. Usually, all of the voxels are treated equally regardless of their positions, appearance, and shape. However, from the human perception system Desingh et al. [2013]; Lindsay and Norman [2013], when we recognize objects, the two most prominent features that make the target object distinguish from other surrounding objects are the corners and uneven surfaces. Furthermore, there are a large amount of voxels located within the objects which are likely to share the similar texture and are redundant for classification task. A relatively small portion of the voxels (at the edges of a scene or the corners of the objects) are the critical clues for identifying the classes. Based on these observations, we propose Local Geometric Anisotropy (LGA) to quantitatively determine the importance of different positions, and further introduce *Position Aware Loss (PA-Loss)* which gives different importance to voxels based on their LGA. With PA-Loss, the network is forced to pay more attention to the surfaces, edges, and corners which leads to slightly faster convergence and a boost in performance of the SSC task.

Another limitation of the previous methods is that they do not use the complete depth information. During voxelization, several pixels in depth can be projected into one voxel in the 3D grid volume and due to the memory limitation, the input resolution used in 3D-CNN is unlikely to be very large, which leads to a coarse voxel level representation. In particular, a common voxelized input for 3D-CNN models is 240 × 144 × 240 Garbade et al. [2018]; Song et al. [2017]; Zhang et al. [2018], while the original input depth has resolution of 640 × 480 or much higher. In this paper, we take both the advantages of the full resolution depth and the encoded voxels by designing a hybrid network structure with two input branches. In the other branch, we feed the network directly with the original depth to avoid the loss of detailed information. In another branch, we adopt the 3D voxels as the input to take advantage of the Truncated Signed Distance Field (TSDF) encoding Izadi et al. [2011]. Through the use of differentiable 2D-3D projection layer, the 2D CNN and 3D CNN blocks are effectively combined within the network.

The main contributions of this paper are two-fold.

- Firstly, we propose a novel PA-Loss for SSC task, which emphasizing the rare voxels that are located on the surface or corners of the scene, while diluting the voxels which carry redundant information within the objects. Our experiments indicate that PA-Loss has the benefit of slightly faster convergence for training and can achieve better performance than previous works.
- Secondly, based on single-view depth, we propose a hybrid network, which takes full advantage of both fine-grained depth and TSDF. The detail information extracted from the full resolution depth is beneficial for semantic labeling as well as scene completion, which also distinguish our approach from the existing mainstream approaches that only use the voxelized TSDF as the sole input.

8.2 Related work

8.2.1 Semantic Scene Completion

Recently several methods have been proposed for SSC using deep learning techniques Garbade et al. [2018]; Guo and Tong [2018]; Song et al. [2017]; Zhang et al. [2018]. Among them, the most representative work is the SSCNet Song et al. [2017] which conducts the semantic labeling and scene completion simultaneously and also proves that these two tasks can benefit from each other. SSCNet takes advantage of TSDF and uses voxels to represent the 3D space. Although better results have been achieved compared with the previous methods, SSCNet ignores the fine-grained information of depth. Zhang *et al.* Zhang et al. [2018] introduces spatial group convolution (SGC) to reduce the computation costs but with poor performance than SSCNet Song et al. [2017]. SEGCloud Tchapmi et al. [2017] employs fine-grained 3D point as input but the computing and memory costs are incredibly high. VVNet Guo and Tong [2018] combines 2D-CNN and 3D-CNN by replacing some 3D volume layers with the corresponding 2D view network layers which leads to a much accurate and efficient network compared with SSCNet. However, it discards TSDF, which can provide the prior knowledge about the space encoding and is vital to distinguish between the empty and occupied parts of the scene.

Two representives of generative models for SSC are 3D-GAN Wu et al. [2016] and ASSC Wang et al. [2018b]. 3D-GAN Wu et al. [2016] uses volumetric convolutional networks to generate 3D objects from a probabilistic space. ASSC Wang et al. [2018b] applies auto-encoder to encode the latent context of the single-view depth and uses a 3D generator to rebuild the 3D complete scene.

To enrich the input information and boost the accuracy of SSC, TS3D Garbade et al. [2018] and DDR-SSC Li et al. [2019] proposed to add a RGB branch in addition to the voxel branch, which introduce extra network or parameters, and are less accurate than our method.

Different from the previous methods, our proposed *Position Aware Loss Network (PALNet)* takes advantages of both the fine grained depth and the TSDF encoded 3D volume. Specifically, we formulate the depth detail as a vital ingredient in the proposed network and make full use of the high-resolution depth by a 2D CNN. In the 2D CNN, a differentiable projection layer is employed to accurately project features in 2D space to the corresponding locations in 3D volume. Moreover, the encoded TSDF provides the prior geometric knowledge of scene which contributes to a much accurate model for SSC.

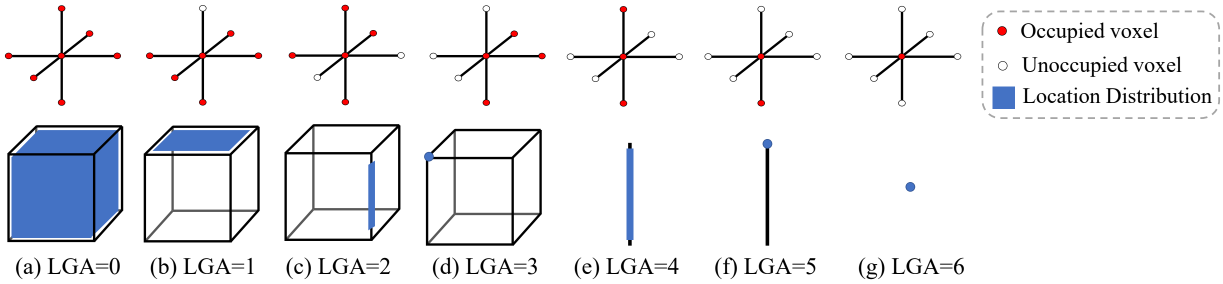


Figure 8.2: Local Geometric Anisotropy: voxels at different positions contain different local geometric information. (a)-(d) respectively indicate voxels inside the object, voxels on the surface, voxels on the edge, and a voxel on the vertex. For a strip-shaped object, (e),(f) indicate voxels on the edge and a voxel on the vertex. (g) indicates an isolated voxel.

8.2.2 Loss Function for 3D Dense Prediction

Compared to 2D image segmentation, 3D dense prediction has three characteristics. Firstly, the number of voxels in 3D dense prediction is much larger than the amount of pixels in the 2D image segmentation. Secondly, the number of voxels ranges a lot among objects with different sizes. Finally, the number of voxels outside the objects is far beyond that of inside the objects. And we further observed that in 3D semantic scene completion, voxels at different positions make different contributions as well as deliver various training difficulties to the scene understanding as mentioned before. Based on the above observations, it is thus vital to choose a suitable loss function to train the 3D network effectively with the consideration of voxel-wise data-balance. There are many classic loss functions available to train 3D networks.

Cross-entropy Loss

Extended from 2D vision tasks, cross-entropy loss can be used in 3D dense prediction. In essence, it treats all the predicted targets equally and is proved less efficient in 3D tasks Milletari et al. [2016]; Song et al. [2017].

Weighted Cross-entropy Loss

Weighting factor $w_c \in [0, 1]$ is introduced based on cross-entropy loss to handle the imbalance problem. Weighted cross-entropy loss can emphasize the importance of classes with rare samples, while it relies on manually set weight parameters. A compromise approach instead of manual selection of weights is to set w_c as the inverse frequency for the corresponding class. And it can only handle category-level data imbalance, but not voxel-wise imbalance.

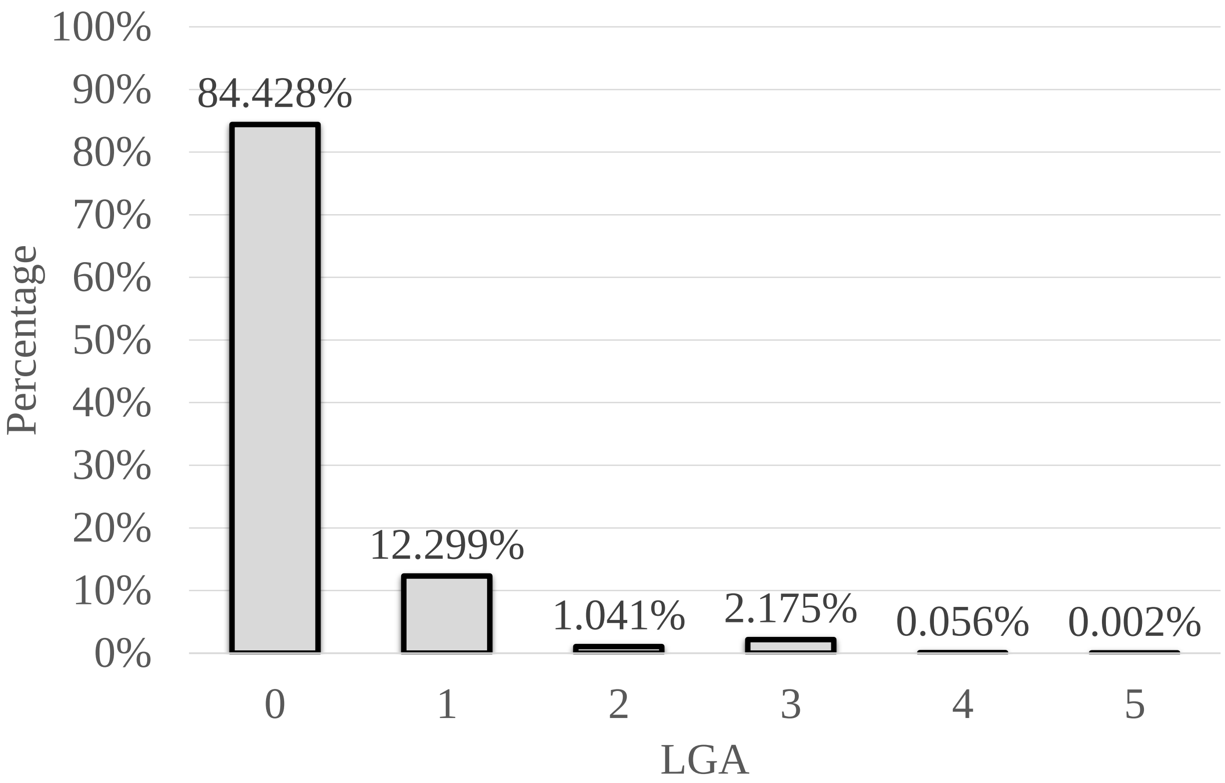


Figure 8.3: The percentage of voxels with different LGA values in dataset NYU Silberman et al. [2012]. More than 84% of the voxels have $LGA = 0$, indicating that most of the voxels are inside the objects, and less than 16% of the voxels are outside of the objects. Voxels in free space are not taken into account.

Focal Loss

Focal loss aims to address the data imbalance in object detection especially when the dataset contains too many easy negatives that contribute no meaningful learning signals. However, one limitation of using focal loss is that it underestimates the importance of well classified samples Nguyen et al. [2018]; Redmon and Farhadi [2018]. Also, the training process becomes sensitive to incorrectly labeled samples Guo et al. [2018].

Dice Loss

Dice loss Milletari et al. [2016] is proposed to address the data imbalance in volumetric medical image segmentation. It is a good choice to address the imbalance between the foreground and background in binary segmentation, but does not generalize well to multi-category segmentation. Besides, it is not as easy to optimize as cross-entropy loss, as its gradient may blow up to some enormous value when both the value of the target label and the prediction are small.

In summary, none of those loss functions can take into account the importance of different positions. In this paper, we propose Local Geometric Anisotropy (LGA) to determine the importance of geometric information contained in different voxels, and LGA is then used to calculate the weight factor for cross-entropy loss to form the proposed PA-Loss, which fully considers the impact of each element, leads to the better performance compared to other loss functions.

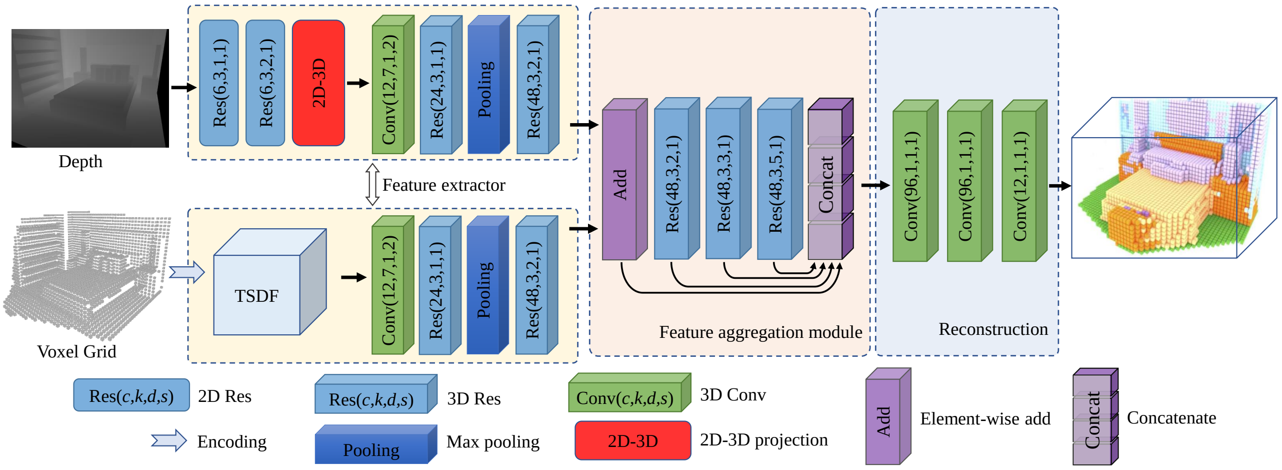


Figure 8.4: The hybrid network structure of PALNet for semantic scene completion. In depth-stream, 2D CNN is used to process the full resolution depth, the feature maps are projected to 3D space with the 2D-3D projection layer, and followed by 3D convolutions. In voxel-stream, TSDF encoded voxels are sent as input, all convolution operations employed are 3D convolutions. After the two streams of information are aggregated, they are sent to the subsequent network with large receptive field to capture 3D context to predict the complete semantic scene. We represent the 3D operation in the form of a cuboid in this figure. Res(c, k, d, s) is the dilated residual blocks with bottleneck. c is the output channel, k is the kernel size, d is the dilation rate, and s is stride in the convolution. Conv(c, k, d, s) is the 3D convolutional layer.

8.3 Methodology

We propose a framework for semantic scene completion in which: (1) Voxel-wise data imbalance is handled by the proposed PA-Loss. (2) Both depth details as well as geometry prior can be fully utilized by the novel PALNet as illustrated in Figure. 8.4. The details of the PA-Loss and PALNet are presented in the following subsections.

8.3.1 Position Importance Aware Constraint

Local Geometric Anisotropy (LGA)

The geometric information contained in voxels at different positions has high variability. In particular, the voxels inside the same object are more homogeneous and likely belong to the same semantic category as their surrounding voxels. Meanwhile, the voxels at the surfaces, edges, and vertices have richer geometric information (have different semantic labels with their surroundings) than those that are inside the objects. We refer to the semantic difference between the current voxel and its surrounding neighbors as Local Geometric Anisotropy.

To measure the LGA of a voxel with a specific semantic label, we focus on voxels that belong to the current semantic category and treat all other voxels with different semantic labels as another category. Specifically, the voxels which are sharing the same category with the current one are denoted 'occupied' voxels and the voxels that belong to other categories are denoted 'unoccupied' voxels, as shown in Figure 8.2. Given a voxel p, its LGA is calculated based on the 6-neighbor voxels, which is expressed as Eq 8.1:

$$M_{LGA} = \sum_{i=1}^{K} (c_p \oplus c_{q_i}) \tag{8.1}$$

where c_p is the semantic label of current voxel p, q_i is one of its K neighbours and $i \in \{1, 2, \cdots, K\}$, c_{q_i} is the semantic label of q_i. And $\oplus$ is the exclusive or (XOR) operation. If voxel p and its neighbour q_i have the same semantic label, then $c_p \oplus c_{q_i} = 0$, otherwise $c_p \oplus c_{q_i} = 1$.

As can be seen in Figure 8.2, voxels that differ significantly from surroundings get a higher M_{LGA} value than voxels consistent with surrounding voxels. For instance, in Figure 8.2(a), voxels which are inside an object are consistent with the surrounding voxels and have LGA value equal to 0. In Figure 8.2(b), voxels which are at the surface of an object are consistent with the interior voxels but different from other voxels regarding to the semantic categories and have LGA value equal to 1. We calculate the LGA value of all the voxels excluding those within the empty space.

The voxels inside the object account for a very high proportion of the overall voxels. However, the dominant homogeneous voxels with consistent semantic properties are easier to be trained and be classified correctly at a very early stage, and contribute little to the gradient update for the whole training process, which leads to slow convergence of the network. A histogram of LGA value is shown in Figure 8.3, where 84.4% of the voxels with LGA equal to 0 are in the interior of the object. All the remaining voxels with LGA values equal to 1 to 6 add up to only 15.6%.

PA-Loss

LGA is used to measure the spatial importance of different voxels and the position importance aware factor to construct the PA-Loss. Through this way, the proposed loss function has a strong response to the voxels with rich detail.

Based on LGA values, we set a base value λ and a constant α to control LGA importance factor I which is denoted as follows:

$$I = \lambda + \alpha M_{LGA} \tag{8.2}$$

PA-Loss is defined as follows:

$$L_{PA} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} I_n y_{nc} \log \hat{y}_{nc} \tag{8.3}$$

where N is the number of total voxels used to calculate the loss, and C is the number of classes. $\mathbf{y_{nc}}$, $\hat{\mathbf{y}}_{\mathbf{nc}}$ are the one-hot vector of the ground truth labels and the corresponding predictions in class c, I_n is the LGA importance factor of the voxel n.

The PA-Loss is differentiable, since the gradient of the PA-Loss can be easily computed from Eq. 8.1, 8.2 and Eq. 8.3 which are all differentiable.

8.3.2 2D and 3D Hybrid Network Architecture

The proposed PALNet is composed of four parts as illustrated in Figure 8.4. The depth stream (8.3.2) takes a full resolution depth as input with a 2D-3D projection layer (8.3.2) to transfer the feature tensors in 2D space into 3D space. The voxel stream (8.3.2) employs 3D CNN and takes the 3D grid as input. Then, both features extracted from the depth stream and voxel stream are fed into the multi-level feature aggregation module (8.3.2). After that, the reconstruction part (8.3.2) predicts the dense volume to perform the semantic scene completion.

Table 8.1: Results of various methods on NYU dataset Silberman et al. [2012]. Bold numbers represent the best score.

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	mIoU
Lin et al. Lin et al. [2013]	58.5	49.9	36.4	0.0	11.7	13.3	**14.1**	9.4	29.0	24.0	6.0	7.0	16.2	1.1	12.0
Geiger et al. Geiger and Wang [2015]	65.7	58.0	44.4	10.2	62.5	19.1	5.8	8.5	40.6	27.7	7.0	6.0	22.6	5.9	19.6
SSCnet Song et al. [2017]	59.3	**92.9**	56.6	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
TS3D Garbade et al. [2018]	64.9	88.8	60.2	8.2	94.1	26.4	19.2	17.2	55.5	48.4	16.4	22.0	34.0	**17.1**	32.6
EsscNet Zhang et al. [2018]	**71.9**	71.9	56.2	17.5	75.4	25.8	6.7	15.3	53.8	42.4	11.2	0	33.4	11.8	26.7
DDR-SSC Li et al. [2019]	71.5	80.8	61.0	21.1	92.2	**33.5**	6.8	14.8	48.3	42.3	13.2	13.9	35.3	13.2	30.4
VVNet Guo and Tong [2018]	69.8	83.1	61.1	19.3	**94.8**	28.0	12.2	19.6	**57.0**	**50.5**	**17.6**	11.9	35.6	15.3	32.9
PALNet(ours)	68.7	85.0	**61.3**	**23.5**	92.0	33.0	11.6	**20.1**	53.9	48.1	16.2	**24.2**	**37.8**	14.7	**34.1**

Table 8.2: Results of various methods on NYUCAD dataset Firman et al. [2016]. Bold numbers represent the best score.

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	mIoU
Zheng et al. Zheng et al. [2013]	60.1	46.7	34.6	-	-	-	-	-	-	-	-	-	-	-	-
Firman et al. Firman et al. [2016]	66.5	69.7	50.8	-	-	-	-	-	-	-	-	-	-	-	-
SSCnet Song et al. [2017]	75.4	**96.3**	73.2	32.5	92.6	49.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	25.1	40.0
TS3D Garbade et al. [2018]	80.2	94.4	76.5	34.4	**93.6**	47.7	**31.8**	32.2	**65.2**	54.2	30.7	**32.5**	**50.1**	30.7	45.7
DDR-SSC Li et al. [2019]	**88.7**	88.5	79.4	54.1	91.5	56.4	14.9	37.0	55.7	51.0	28.8	9.2	44.1	27.8	42.8
VVNet Guo and Tong [2018]	86.4	92.0	80.3	-	-	-	-	-	-	-	-	-	-	-	-
PALNet	87.2	91.7	**80.8**	**54.8**	92.8	**60.3**	15.3	**43.1**	60.7	**59.9**	**37.6**	8.1	48.6	**31.7**	**46.6**

Depth-Stream

As depicted in the upper left of Figure. 8.4, the depth-stream contains both 2D CNN and 3D CNN modules. The full resolution depth is first passed a 2D CNN module to extract the 2D features. The 2D CNN module consists two 2D dilated residual blocks with bottle-neck structure (Resblock in Figure. 8.4). The 3D CNN module consists of a 3D convolution layer, two 3D Resblocks and one 3D pooling layer. Through the 2D-3D projection layer, 2D features extracted from depth can be mapped into corresponding 3D space, which helps the network to preserve detailed information. Within the depth stream, each convolutional layer employs the Residual blocks He et al. [2016]. We adopt the dilated convolution Yu and Koltun [2016] to increase the receptive field of the network. We also take the advantage of its bottleneck version to increase the capacity of the network and reduce its parameters. Compared to the pure 3D CNN based approaches, a part of our hybrid network uses 2D CNN, which introduces a small number of parameters but can effectively improve the network depth and capacity.

2D-3D Projection

Each pixel in depth can be projected to a voxel in 3D space. The mapping index $\mathcal{T}_{u,v}$ of a pixel in depth at (u, v) can be computed using the depth value $I_{u,v}$ and the camera parameters which are provided along with each image. The 2D feature map can be mapped to the 3D space according to the correspondence between the depth and 3D voxels.

Voxel-Stream

In the voxel-stream, flipped-TSDF (f-TSDF) Garbade et al. [2018]; Song et al. [2017] is adopted as the network input. As depicted in the bottom left of Figure. 8.4, the 3D CNN module contains a 3D convolution, two 3D Resblocks, and one 3D pooling layer. Meanwhile, we voxelize the full 3D scene with object labels as the ground truth and the rest voxels in the scene are labeled as empty space resulting in a fully labeled voxel grid representation of the scene.

Multi-level Feature Aggregation Module

The multi-level feature aggregation module combines the two stream features through element-wise addition. Then, the combined features are handled by a series of 3D dilated residual blocks to increase the receptive field and different dilation rates are employed within each block to reduce the gridding problem Wang et al. [2018a].

Dense Grid Reconstruction

Finally, in the dense grid reconstruction, there are three standard 3D convolutional layers piled up to give the outputs. PALNet predicts the probability distribution of voxel occupancy and object categories for all voxels inside the camera view frustum.

8.4 Experimental results

8.4.1 Datasets

We evaluate the proposed method on the NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets. The NYU dataset includes 1449 depth maps (795 for training, 654 for testing) captured by the Kinect depth sensor. The ground truth for completion and semantic segmentation are obtained from Guo et al. [2015] by voxelizing the 3D mesh annotations. In some cases, the manually labeled volumes and their corresponding depth are not well aligned in the NYU dataset. To address the misalignment, Firman *et al*. Firman et al. [2016] provide the NYUCAD dataset, in which the depth is rendered from the labeled volume.

8.4.2 Implementation Details

We implement the PALNet with PyTorch and train it from scratch. The dimensions of the 3D space $(H \times D \times V)$ are $4.8 \times 2.88 \times 4.8\,\mathrm{m}$. We encode the 3D scene into a flipped TSDF (f-TSDF) with grid

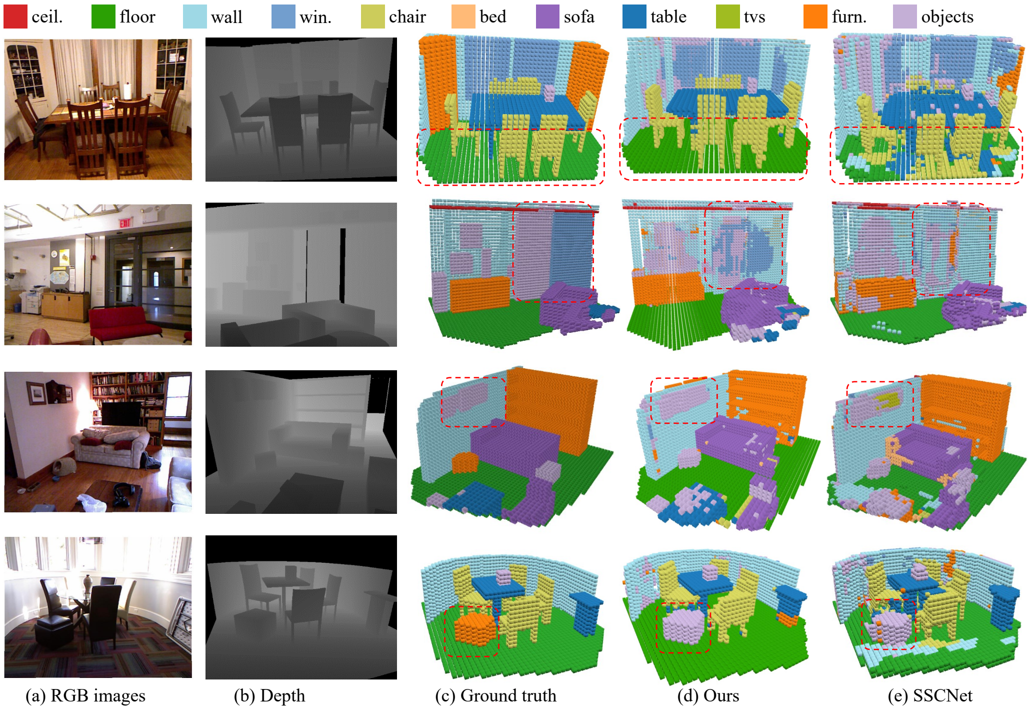


Figure 8.5: For a scene shown in (a), Semantic Scene Completion (SSC) takes (b) depth image as input and output dense 3D semantic volume. (c) gives the ground truth. (d) and (e) are the predictions of the proposed PALNet and SSCNet Song et al. [2017].

size of 0.02 m and a truncation value of 0.24 m resulting in a volume of resolution 240 × 144 × 240 as the input of the 3D branch. For the importance factor I, the base value λ, and the constant α are set to 1.0 and 0.5 respectively.

The details of the network architecture are shown in Figure. 8.4. During training, we minimize the PA-Loss using SGD with a momentum 0.9 and a weight decay 0.0001. The model is trained with 4 GTX 1080Ti GPUs for 40 epochs with batch size 4. The initial learning rate is set to 0.01, and it is reduced by a factor of 0.1 every 10 epochs.

8.4.3 Evaluation Metric

For semantic scene completion (SSC), we measure the intersection over union (IoU) between the ground truth and the predicted volume, excluding voxels outside the view or the room. The IoU score is calculated for each category and then it is averaged over all the classes to obtain mean IoU (mIoU) score. For the scene completion (SC) task, we treat all non-empty object classes as one category and evaluate precision, recall, and IoU of the binary predictions on occupied voxels.

8.4.4 Comparisons with the State-of-the-arts

We set the new state-of-the-art performance for both scene completion (SC) and semantic scene completion (SSC) on both NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets.

We compare PALNet with previous state-of-the-art methods as represented in Table 8.1 on NYU dataset and PALNet achieves the best performance in both SC and SSC tasks. SSCNet and VVNet are pre-trained on the synthetic dataset SUNCG Song et al. [2017]. Compared to SSCNet, the IoUs of PALNet for SC and SSC tasks increased by 4.7% and 3.6%, respectively. Compared to VVNet Guo and Tong [2018], PALNet leads the SC task by 0.2% and the SSC task by 1.2%. DDR-SSC Li et al. [2019] and TS3D Garbade et al. [2018] use additional color information. Our method achieves 3.7% and 1.5% higher IoU in SSC than these methods, respectively. Table 8.1 also lists the IoU for each object class, and our method is relatively more accurate than most of the state-of-the-art methods in each category.

We also compare our method with Zheng *et al*. Zheng et al. [2013] and Firman *et al*. Firman et al. [2016] on the NYUCAD dataset. The results are shown in Table 8.2, and our PALNet achieves the best performance among all the methods as well.

8.4.5 Qualitative Results

Figure. 8.5 illustrates the results of SSC task on NYUCAD dataset generated by different methods. In Figure. 8.5, (a) is the color image, (b) is the input depth, (c) is the ground truth, (d) and (e) are the results of PALNet and SSCNet Song et al. [2017]. As can be seen, PALNet gives more accurate predictions than SSCNet Song et al. [2017] such as the floor and the chairs in the first row. Although the windows in the second row are hard to distinguish, our method still achieves better semantic segmentation results. The paintings in the third row also demonstrate the superiority of our approach. In the fourth row, the ground-truth *furniture* circled by the red dashed rectangle is very similar to the object in terms of shape and neither SSCNet nor PALNet can correctly identify the furniture beside the chairs.

Table 8.3: Results of variant PALNets and SSCNet Song et al. [2017].

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SSCnet Song et al. [2017]	75.4	**96.3**	73.2	32.5	92.6	49.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	25.1	40.0
PALNet-3D	83.3	94.5	79.1	49.1	92.3	56.2	6.8	36.6	**62.5**	57.6	34.6	5.9	45.6	30.7	43.5
PALNet-2D	83.5	93.4	78.8	52.2	92.0	56.6	**20.1**	40.2	61.1	58.8	33.9	**17.1**	45.3	30.0	46.1
PALNet-hybrid	**87.2**	91.7	**80.8**	**54.8**	**92.8**	**60.3**	15.3	**43.1**	60.7	**59.9**	**37.6**	8.1	**48.6**	**31.7**	**46.6**

8.5 Ablation study

We validate the proposed PALNet with a set of ablation tests on NYUCAD Firman et al. [2016] dataset. Specifically, we examine three things as listed below: 1) effectiveness of PA-Loss; 2) hybrid structure *i.e.* combined 2D depth and 3D grid.

8.5.1 PA-Loss

In order to prove the effectiveness of the proposed Position Aware Loss(PA-Loss), we design two sets of comparative experiments. Firstly, we train PALNet with different loss functions (PA-Loss and other loss functions) to compare their training results. Then, we apply PA-Loss to SSCNet to demonstrate that PA-Loss can improve the accuracy of SSCNet. The details of these ablation studies are presented as follows.

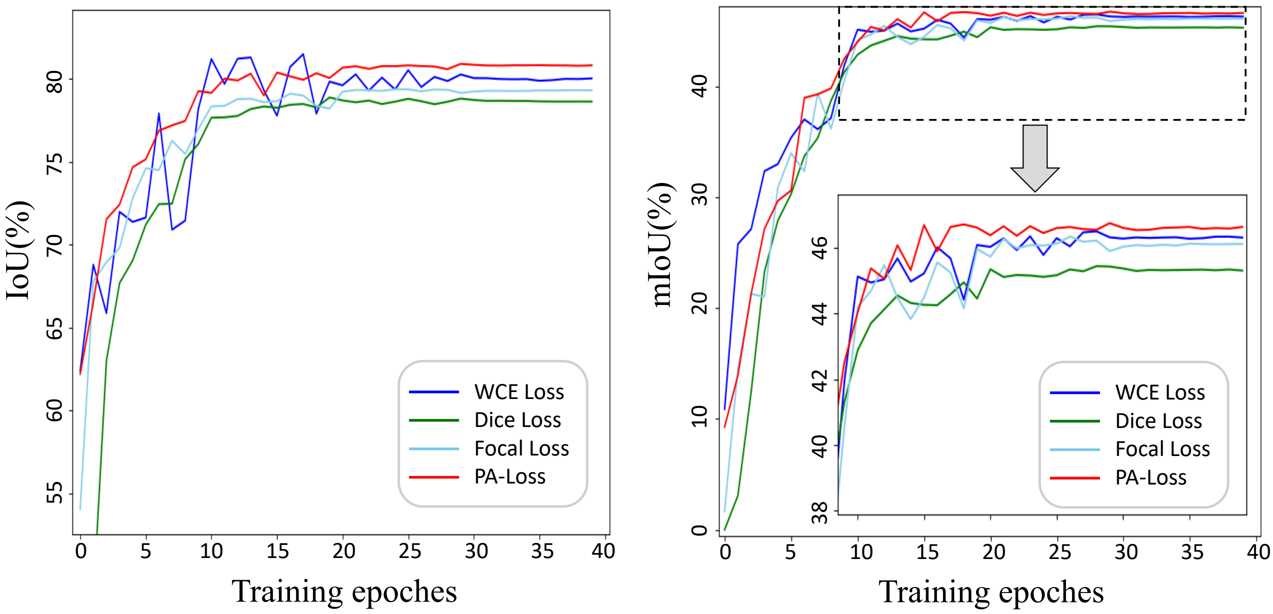


Figure 8.6: Training curves of network with different loss functions for scene completion (left) and semantic scene completion (right).

Comparison with Different Loss Functions

The three comparison methods are weighted cross-entropy (WCE) loss, focal loss and dice loss.
Weighted Cross-entropy Loss. The weighted cross-entropy loss is defined as follows:

$$L_{WCE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c y_{nc} \log \hat{y}_{nc} \tag{8.4}$$

where N is the number of total voxels, C is the number of classes. Voxels belong to each class have different weights w_c. Specifically, the ratio of each category is counted including empty voxels on the training set and use the reciprocal of the ratio is used as the weight of each category. $\hat{y}_{nc}$ and y_{nc} denote the prediction and ground truth of voxel n belong to class c.
Focal Loss. Focal loss adds a factor $(1 - p_t)^\gamma$ to the standard cross-entropy criterion, as follows:

$$L_{FL} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} (1 - \hat{y}_{nc})^\gamma y_{nc} \log \hat{y}_{nc} \tag{8.5}$$

In the experiment, we set $\gamma = 2$ as it is suggested to reduce the relative weights for easy classified examples.
Multi-class Dice Loss. We generalize the standard dice loss for multi-class segmentation by applying binary version on each class iteratively. The multi-class dice loss is defined as:

$$L_{MDL} = \sum_{c=1}^{C} 1 - \frac{2\sum_{n=1}^{N} y_{nc}\hat{y}_{nc}}{\sum_{n=1}^{N} y_{nc}^2 + \sum_{n=1}^{N} \hat{y}_{nc}^2} \tag{8.6}$$

We use these three loss functions to train the PALNet from scratch and evaluate the accuracy of

Table 8.4: Results of PALNet trained with various loss functions.

Loss function	Precision	Recall	IoU	mIoU
WCE Loss	79.4	**95.9**	79.9	46.3
Dice Loss	**88.2**	83.3	78.64	45.3
Focal Loss	85.6	91.8	79.3	46.1
PA-Loss	87.2	91.7	**80.8**	**46.6**

Table 8.5: Results of SSCNet trained with PA-Loss and Weighted Cross-Entropy Loss (WCE Loss). With SSCNet* denotes our implementation of SSCNet Song et al. [2017].

Methods	Loss function	Precision	Recall	IoU	mIoU
SSCNet	WCE Loss	75.4	**96.3**	73.2	40.0
SSCNet*	WCE Loss	76.5	95.7	74.8	42.1
SSCNet*	PA-Loss	**81.6**	91.6	**76.0**	**43.4**

the network after each epoch. As shown in Table. 8.4, the network trained with PA-Loss provides more accurate results than that trained with other loss functions on both SC and SSC tasks. In the task of semantic scene completion, PALNet trained with PA-Loss achieves 0.3%, 1.3% and 0.5% higher mIoU than that trained with weighted cross-entropy loss, dice loss and focal loss.

In addition, we find that training convergence speed of using PA-Loss is slightly faster than that of other loss functions. We plot the testing accuracy of each epoch as a curve, as shown in Figure. 8.6 for scene completion (left) and semantic scene completion (right).

Apply PA-Loss to SSCNet

To further verify the effectiveness and generalization ability of PA-Loss, we apply PA-Loss to SSCNet Song et al. [2017] in SSC task. We reimplemented SSCNet in PyTorch and denote our implementation as SSCNet* in the following discussions. We train SSCNet* with WCE Loss and PA-Loss separately and evaluate the performance of SSCNet* on NYUCAD dataset. The experimental results are shown in Table. 8.5. The results reported in SSCNet Song et al. [2017] are also listed for reference. Our reimplemented SSCNet* gets comparable (slightly better) accuracy than SSCNet Song et al. [2017]. Compared to SSCNet* trained with WCE Loss, the one trained with our PA-Loss gets better accuracy on both SC and SSC tasks.

8.5.2 Hybrid Structure Combined 2D Depth and 3D Voxels

We explore the contribution of the 2D and 3D parts of the hybrid structure. To investigate the effect of each stream, we develop two variants of the PALNet where each variant contains only one of the two input branches. Note that the element-wise add operation is no longer needed in the single-stream network and the feature maps are fed into the multi-level feature aggregation stage immediately. Since the 3D branch is extended from SSCNet , therefore, to conclusively exemplify our method we select SSCNet for comparison in this section.

As shown in Table 8.3, both variants with one stream as input achieve better performance than SSCNet Song et al. [2017] on the SC and SSC tasks. The combination of 2D and 3D streams increases the accuracy of PALNet in SC to 80.8% and SSC to 46.6%.

Table 8.6: The number of parameters, inference speed and GPU memory usage of SSCNet Song et al. [2017] and the proposed PALNet.

Method	Params (k)	FLOPs (G)	Speed (FPS)	Memory (M)	Network Depth	SSC IoU(%)
SSCNet	930.0	163.8	0.7	5305	14	40.0
PALNet	**223.0**	**78.8**	**1.2**	**3717**	**25**	**46.6**

The variant network with 2D depth as the input achieves 46.1% IoU on the SSC task and is 2.6% higher than that of the other variant with 3D voxels as the input. The significant advantages of the 2D network over 3D network strongly suggest that the details in depth are very useful for semantic scene completion. The 3D branch with manually designed f-TSDF gets better result (0.3%) than the 2D branch on the task of scene completion. This is reasonable due to the carefully designed f-TSDF encodes the geometry information well. The symbolic information of f-TSDF can explicitly tell the network where the occluded space is, and help the network to pay attention to these parts for corresponding shape completion.

In Table 8.6, the parameters, Flops, inference speed, memory and accuracy of the networks among SSCNet Song et al. [2017] and PALNet are listed out. By taking advantage of the bottle-neck residual network, PALNet has much deeper structure than SSCNet Song et al. [2017] and achieve better accuracy.

8.6 Conclusions

We introduce PALNet which takes both the depth and TSDF as inputs for semantic scene completion. The feature from 2D-stream are projected and concatenated with the feature in 3D-stream and trained in an end-to-end manner. We also propose a position importance aware loss, PA-Loss, which leads to slightly faster training convergence and better performance. The experiments on both synthetic and real datasets validate the superior performance of the proposed method. The two interesting topics that are worth exploring in our future work include: (1) a better trade-off between 2D CNNs and 3D CNNs to employ less 3D convolutional layers without sacrificing the performance, and (2) to extend our PALNet framework with RGB-D as input.

Bibliography

L. Chen, K. Francis, and W. Tang. Semantic augmented reality environment with material-aware physical interactions. In *Proc. IEEE ISMAR*, pages 135–136. IEEE, 2017.

W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Indoor scene understanding with geometric and semantic contexts. *IJCV*, 112(2):204–220, 2015.

K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar. Depth really matters: Improving visual salient region detection with depth. In *Proc. BMVC*, pages 98.1–98.11, 2013. ISBN 1-901725-49-9. doi: 10.5244/C.27.98.

M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, pages 5431–5440, 2016.

M. Garbade, J. Sawatzky, A. Richard, and J. Gall. Two stream 3d semantic scene completion. *arXiv:1804.03550*, 2018.

A. Geiger and C. Wang. Joint 3d object and layout inference from a single rgb-d image. In *German Conference on Pattern Recognition*, pages 183–195, 2015.

J. Guo, P. Ren, A. Gu, J. Xu, and W. Wu. Locally adaptive learning loss for semantic image segmentation. *arXiv preprint arXiv:1802.08290*, 2018.

R. Guo, C. Zou, and D. Hoiem. Predicting complete 3d models of indoor scenes. *arXiv preprint arXiv:1504.02437*, 2015.

Y. X. Guo and X. Tong. View-volume network for semantic scene completion from a single depth image. *arXiv:1806.05361*, 2018.

S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *Int. J. Computer Vision*, 112(2):133–149, Apr 2015. ISSN 1573-1405.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. CVPR*, pages 770–778, 2016.

S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, pages 559–568. ACM, 2011.

J. Li, Y. Liu, D. Gong, Q. Shi, X. Yuan, C. Zhao, and I. Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019.

D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, pages 1417–1424, 2013.

P. H. Lindsay and D. A. Norman. *Human information processing: An introduction to psychology*. Academic press, 2013.

F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. IEEE Conf. 3D Vision*, pages 565–571. IEEE, 2016.

T. Nguyen, T. Ozaslan, I. D. Miller, J. Keller, G. Loianno, C. J. Taylor, D. D. Lee, V. Kumar, J. H. Harwood, and J. Wozencraft. U-net for mav-based penstock inspection: an investigation of focal loss in multi-class segmentation for corrosion identification. *arXiv preprint arXiv:1809.06576*, 2018.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. ECCV*, pages 746–760, 2012.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198, 2017.

L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision*, pages 537–547, 2017.

D. Van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International journal of virtual reality*, 9(2):1, 2010.

J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. In *IROS*, pages 2442–2447, 2017.

V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *Proc. ICRA*, pages 75–82, 2015.

P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *Winter Conference Applications of Computer Vision*, pages 1451–1460, 2018a.

Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Adversarial semantic scene completion from a single depth image. In *International Conference on 3D Vision*, pages 426–434. IEEE, 2018b.

J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, pages 82–90, 2016.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. 2016.

J. Zhang, H. Zhao, A. YaoE, Y. Chen, L. Zhang, and H. LiaoE. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, pages 733–749, 2018.

B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, pages 3127–3134, 2013.

Chapter 9

3D Gated Recurrent Fusion for Semantic Scene Completion

This paper tackles the problem of data fusion in the semantic scene completion (SSC) task, which can simultaneously deal with semantic labeling and scene completion. RGB images contain texture details of the object(s) which are vital for semantic scene understanding. Meanwhile, depth images capture geometric clues of high relevance for shape completion. Using both RGB and depth images can further boost the accuracy of SSC over employing one modality in isolation. We propose a 3D gated recurrent fusion network (GRFNet), which learns to adaptively select and fuse the relevant information from depth and RGB by making use of the gate and memory modules. Based on the single-stage fusion, we further propose a multi-stage fusion strategy, which could model the correlations among different stages within the network. Extensive experiments on two benchmark datasets demonstrate the superior performance and the effectiveness of the proposed GRFNet for data fusion in SSC. Code will be made available.

9.1 Introduction

Understanding the surroundings is a fundamental capability for many real-world applications such as augmented reality Chen et al. [2017a], robot grasping Varley et al. [2017], or autonomous navigation Doan et al. [2019]. Different abstractions are possible, and even complementary. Semantic labeling of the scene allows for a high level reasoning, while 3D geometry completion enables basic spatial capabilities. Semantic scene completion (SSC) aims at solving both simultaneously.

An RGB-D sensor allows acquiring depth information from the scene along side the RGB image. On the one hand, RGB image contains rich details about the color and texture, which are the primary cues for the semantic scene understanding. On the other hand, depth carries more clues about the object geometry and distance information, which are much reliable in reflecting the position, shape, and occlusion relationship between objects within the scene. Many vision applications have already benefit from using both modalities in their tasks, such as object detection Chen et al. [2017b]; Gupta et al. [2014], video segmentation Emre Yurdakul and Yemez [2017]; Fu et al. [2017]; Sultana et al. [2018], action recognition Hu et al. [2018]; Ijjina and Chalavadi [2017]; Zhang et al. [2018a], or visual SLAM Kerl et al. [2013]; Lu and Song [2015]; Whelan et al. [2013]. Recent studies Garbade et al. [2018];

Li et al. [2019] in SSC also demonstrate that employing both, RGB image and depth, can outperform using only one modality Song et al. [2017].

However, fusing the information from RGB and depth is still an unsolved problem, and becomes an obstacle which hinders the performance of SSC. Albeit some recent works conduct data fusion between RGB and depth, they usually employ some, "manually" set, basic operation to fuse the data. Those includes *sum fusion* Hazirbas et al. [2016]; Li et al. [2019], *max fusion* Kang et al. [2014], *concatenate fusion* Couprie et al. [2013]; Guo and Chen [2018], *transform fusion* Wang et al. [2016] and *bilinear fusion* Lin et al. [2015]. Nevertheless, RGB and depth data are not equivalent quantities, while still providing complementary yet redundant information. Therefore, we propose to extract the information in a selective manner from both modalities, and fuse them accordingly with respect to the specific task.

We present the Gated Recurrent Fusion (GRF) block, which can provide adaptive selection and aggregation of RGB and depth information. On the one hand, the *gate* component in the GRF fusion block selects, in an adaptive manner, various positions of different importance in aligned RGB-D frames regarding to the contribution from both modalities. The *gate* effectively selects valid information while filters out the irrelevant one. On the other hand, the *memory* component in the GRF fusion block effectively preserves the complementary information, which can compensate the missing or ambiguous details of the data obtained from different modalities.

Furthermore, the GRF fusion block offers the flexibility to be cascaded to a multi-stage configuration that combines high-level and low-level features. GRF fusion block is extended from the Gated Recurrent Unit (GRU) Cho et al. [2014]. Based on the GRF fusion block, we build the GRFNet for the semantic scene completion, and provide single- and multi-stage fusion versions of GRFNet. In the single-stage fusion version, depth and RGB features are fed into the same GRF fusion block individually. In the multi-stage fusion version, depth and RGB features of different stages form an interleaved sequence and are input into the same GRF fusion block consecutively.

The multi-stage version takes advantage of both low-level and high-level features, and achieves better performance than the single-stage version.

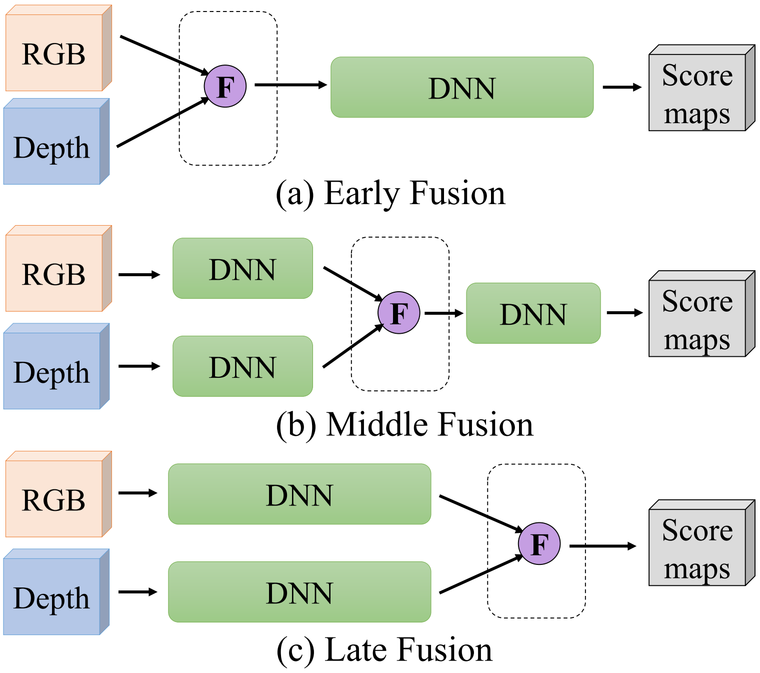
In summary, the contributions of this work are mainly two-fold:

- An end-to-end 3D-GRF based network, GRFNet, is presented for fusing RGB and depth information in the SSC task, through employing *gate* and *memory* components, the selection and fusion between two modalities can be conducted effectively. To the best of our knowledge, this is the first time that gated recurrent network is employed for data fusion in the SSC task.
- Within the framework of GRFNet, single-stage and multi-stage fusion strategies are proposed. While outperforming existing fusing strategies in the SSC task already with the single stage, the multi-stage fusion proves to give the best results.

Extensive experiments demonstrate that the proposed GRFNet achieves superior performance on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets.

9.2 Related Work

In this section, we briefly look through the deep learning based methods for SSC, with emphasis on the discussion of the existing multi-modality fusion strategies.



(a) Early Fusion

(b) Middle Fusion

(c) Late Fusion

Figure 9.1: Fusion at different stages. Early fusion contains more low-level features, while the input of the late fusion contains more abstract high-level features.

9.2.1 Semantic Scene Completion

The goal of SSC is to produce a complete 3D voxel representation for a scene from a single-view input. Specifically, Song *et al.* (Song et al. [2017]) propose an end-to-end 3D convolutional network (SSCNet) which is based on the single-view depth as input that can simultaneously predict the results of scene completion and semantic labeling. SSCNet has high computational costs due to the adoption of 3D convolutions. Zhang *et al.* (Zhang et al. [2018b]) introduce spatial group convolution (SGC) into SSC for accelerating the computation of 3D dense prediction task. Meanwhile, Han *et al.* (Han et al. [2017]) employ the long short-term memory (LSTM) to recover missing parts of 3D shapes. Dai *et al.* (Dai et al. [2018]) use a coarse-to-fine strategy to handle large scenes with varying spatial extent. Although the depth-based approach has made significant progress, the absence of texture details prevents improving SSC.

In order to incorporating the color information, TS3D Garbade et al. [2018] introduces the RGB image into SSC and uses a 2D network to acquire semantic segmentation results. Semantic outputs of the RGB stream are concatenated with inputs of the depth stream to obtain the completed 3D scene. DDR-SSC Li et al. [2019] uses two parallel feature extraction branches with the same structure to obtain information from RGB and depth simultaneously. A multi-stage structure with element-wise addition is employed to perform feature fusion. Thanks to the semantic information provided by RGB, the semantic labeling accuracy of both TS3D and DDR-SSC has significantly been improved compared to SSCNet. However, none of these methods takes into account the selective fusion of multi-modal information, which limit those algorithms to achieve better performance.

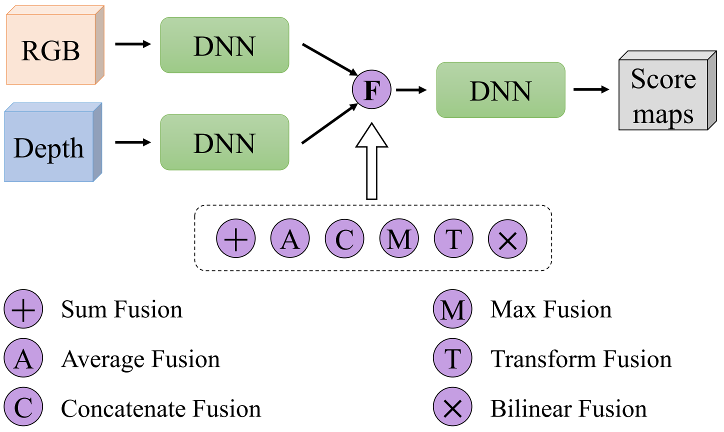


Figure 9.2: Several typical single-stage fusion methods.

9.2.2 Fusion Schemes

The RGB-D information fusion is important to many vision applications. In general, the fusion scheme can be divided into three categories, *e.g.* early fusion Couprie et al. [2013], middle fusion Ren et al. [2012] and late fusion Simonyan and Zisserman [2014]; Yue-Hei Ng et al. [2015], as shown in Figure 9.1. According to the stages of fusion, these schemes can also be divided into single-stage fusion and multi-stage fusion Hazirbas et al. [2016]; Park et al. [2017].

Single-Stage Fusion There are several general patterns for single-stage fusion as shown in Figure 9.2. Specifically, Sum fusion Hazirbas et al. [2016]; Li et al. [2019] computes the sum of the two feature maps at the same spatial locations. Average fusion is essentially a weighted sum fusion with equal weights. Max fusion Kang et al. [2014] takes the feature with the maximal value from multiple feature maps. Concatenate fusion stacks the features with channels Couprie et al. [2013]; Guo and Chen [2018]. Wang *et al.* (Wang et al. [2016]) propose an encoder-decoder architecture which exchanges the information of multi-modal data in the latent space. Bilinear fusion Lin et al. [2015] computes an outer matrix product of the two features at each pixel location.

There are a few methods that consider the complementary and selectivity of data fusion. Specifically, Li *et al.* (Li et al. [2016]) develop a novel LSTM model to fuse scene contexts adaptively. Cheng *et al.* (Cheng et al. [2017]) use the concatenated feature maps of RGB and depth to learn an array G to weight the contribution of one input modality and $1 - G$ to weight the other input modality. Wang *et al.* (Wang and Gong [2019]) use the same strategy as Cheng et al. [2017] to fuse the feature maps from RGB and Depth in saliency detection. However, Li *et al.* (Li et al. [2016]) only consider the complementarity of information but ignore the selectivity of the data, the other two methods only consider the selectivity of information and cannot guarantee the complementarity of information. Moreover, these methods are single-stage fusion and lack of scalability.

Multi-stage Fusion According to the way for aggregating multi-modal information, this paper divides multi-stage fusion algorithms into merge fusion, cross fusion, and external fusion. Hazirbas *et al.* (Hazirbas et al. [2016]) adopt the merge fusion structure to fuse the two branches of features extracted from RGB and depth images. The feature maps from depth are fused into the RGB branch by stages with an element-wise summation. Wang *et al.* (Wang et al. [2016]) use cross fusion to merge the common features of RGB and depth, and keep the modality specific features separated

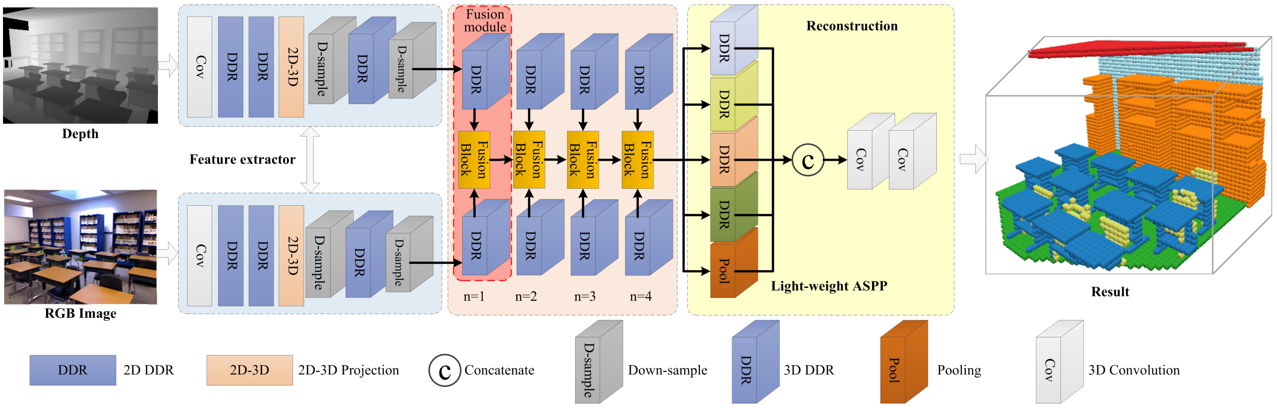


Figure 9.3: The network architecture of GRFNet is extended from Dimensional Decomposition Residual (DDR) network Li et al. [2019]. GRFNet has two feature extractors to capture the features from depth and RGB images respectively. The feature extractor contains a projection layer to map 2D feature to 3D space. The GRF fusion block (denoted by yellow boxes in the middle) replaces the original fusion unit to take full advantage of the multi-modal information. With two DDR plus their corresponding GRF fusion block to form a fusion module, also named single-stage fusion module (denoted by the red box in one column). GRFNet is composed of a multi-stage (4-stage here) fusion module. Then we use light-weight ASPP to obtain multiple receptive fields information. Different colors of the DDR block denote various receptive fields. Then the network uses two 3D convolutions to predict occupancies and object labels simultaneously.

from each other. Both Park *et al.* (Park et al. [2017]) and Li *et al.* (Li et al. [2019]) use an external fusion mechanism. Specifically, Li *et al.* (Li et al. [2019]) capture features of RGB and depth image at different levels, these features at each level are fused separately and then assembled all at once before the reconstruction part. Park *et al.* (Park et al. [2017]) propose RDFNet to fuse multi-modal features separately by multiple fusion blocks, and refine the fused features one by one through a set of refine blocks. In RDFNet, each fusion introduces an additional fusion block with a new set of extra parameters. The artificially designed fusion blocks are complex and require multiple parameters that are not easy to migrate to other applications. These multi-stage fusion methods use high-level and low-level features achieving high accuracy. However, each fusion block within the multi-stage mostly adopts concatenation or summation, ignoring the adaptive selection of the multi-modal data.

On the contrary, our proposed GRF fusion block extends the standard gated recurrent unit (GRU), where the gate and the memory structures can adaptively select and preserve valid information. Besides, GRFNet adopts the form of a recurrent network. When performing multi-stage fusion, GRF modules exploit parameter sharing. And experiments show that both of the proposed single- and multi-stage GRFNets achieve better accuracy than previous methods.

9.3 Methodology

Our proposal, GRFNet, extends the network architecture of DDR-SSC Li et al. [2019], and focuses on improving the fusion strategy. We subtly adopt the gate structure and memory mechanism in GRU unit to form a multi-modal feature GRF fusion block with the power of autonomous selectivity and adaptive memory preservation. Moreover, taking advantage of its recurrent nature, we further propose a multi-stage fusion strategy to utilize both low-level and high-level features with introducing insignificant parameters.

In the feature extractor part, the network uses dimensional decomposition residual (DDR) blocks to extract the local textures and the geometry information. A projection layer is employed to connect the 2D and 3D parts. The multi-stage fusion module consists of four single-stage fusion modules that can effectively combines the RGB features and depth features. The fused features are fed into the subsequent light-weight atrous spatial pyramid pooling (LW-ASPP Li et al. [2019]). After that, another two point-wise convolutional layers are used to predict the semantic labels for each voxel in the 3D volume.

The network maps each voxel to one of the labels $C = c_0, c_1, \cdots c_N$, where N is the number of semantic classes, and c_0 represents the empty voxel.

9.3.1 Gated Recurrent Unit

Gated recurrent unit (GRU) Cho et al. [2014] is a popular model in recurrent neural networks (RNN) and has an outstanding performance in many natural language processing (NLP) tasks Chorowski et al. [2015]; Kim et al. [2016]. A GRU has two gate structures, a memory structure, and can be reused recurrently. However, few researchers have explored the power of GRU in the field of 3D vision, especially for feature fusion. We find that GRU highly aligns with our requirements for an effective multi-modal fusion strategy in SSC.

The gate structure in GRU enables the selective fusion of multi-modal features. The memory structure ensures that valid information can be retained for future fusion purpose. The characteristics of its recurrent network enable GRU to be reused in the multi-stage fusion while sharing the same set of parameters. Compared to GRU, the structure of Long short-term memory (LSTM) Hochreiter and Schmidhuber [1997] is more complicated and has an extra forget gate with more parameters. ConvGRU Ballas et al. [2015] is a convolutional version of GRU. We extend ConvGRU to 3D convolutional in our GRFNet, and modify it to fit the feature fusion in SSC task.

9.3.2 Gated Recurrent Fusion Block of RGB-D Features

As shown in Figure 9.4, at fist step (left), gated recurrent fusion (GRF) block takes one of the RGB-D features as input. The outputs of this step will be used as the hidden state. Then, in the second step (right), GRF fusion block takes the features of another modality as input. These two steps reuse the GRF fusion block and share the same set of parameters. Next, we will use the first step with input f^d as an example to explain in detail the principle and work flow of the GRF fusion block.

In contrast to the commonly used GRU for encoding information in a temporal way, the way we use GRF to fuse RGB+Depth features is somehow different. Specifically, the GRU handles the fusion of RGB and depth information in a ‘modality’, rather than a ‘sequential’, way. And for the multi-stage fusion process, similar as other deep neural networks, it is more like the low-level multi-modal feature to guide the following high-level multi-modal feature to be merged.

Hidden State The hidden state h_p, along with the current input f to control the reset and update gates. The output of the previous stage will be used as the hidden state of current stage. At the first

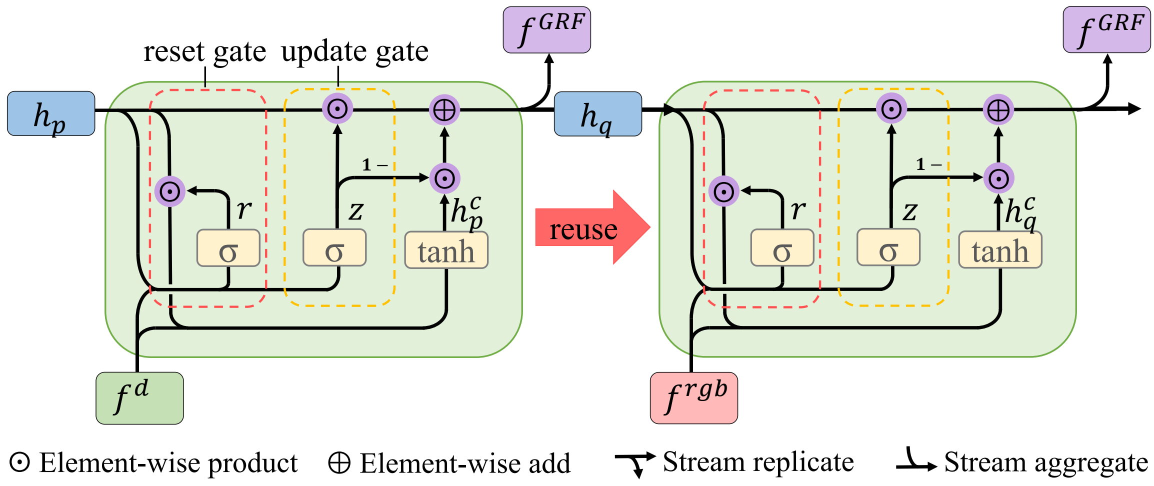


Figure 9.4: GRF fusion block. At step p, the input of GRF fusion block is one of the features from depth or RGB, and in the next step, input is the other. Both GRF fusion blocks share the same set of parameters.

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
Lin *et al.* (Lin et al. [2013])	58.5	49.9	36.4	0.0	11.7	13.3	14.1	9.4	29.0	24.0	6.0	7.0	16.2	1.1	12.0
Geiger *et al.* (Geiger and Wang [2015])	65.7	58.0	44.4	10.2	62.5	19.1	5.8	8.5	40.6	27.7	7.0	6.0	22.6	5.9	19.6
SSCNet Song et al. [2017]	57.0	**94.5**	55.1	15.1	**94.7**	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
EsscNet Zhang et al. [2018b]	**71.9**	71.9	56.2	17.5	75.4	25.8	6.7	15.3	**53.8**	42.4	11.2	0	33.4	11.8	26.7
DDR-SSC Li et al. [2019]	71.5	80.8	61.0	21.1	92.2	**33.5**	6.8	14.8	48.3	42.3	13.2	**13.9**	35.3	13.2	30.4
GRFNet	68.4	85.4	**61.2**	**24.0**	91.7	33.3	**19.0**	**18.1**	51.9	**45.5**	**13.4**	13.3	**37.3**	**15.0**	**32.9**

Table 9.1: Results on the NYU dataset Silberman et al. [2012]. Bold numbers represent the best scores.

step of fusion between depth feature f^d and RGB feature f^{rgb}, that is $p = 1$, we use the sum fusion of two modal features to initialize the hidden state, as $h_0 = f^d + f^{rgb}$.

Reset Gate At step p, the hidden state h_p and the current input f^d together to decide the status of the reset gate r by

$$r = \sigma\left(W_r\left(f^d, h_p\right)\right) \tag{9.1}$$

The two feature stream f^d and h_p are concatenated and fed into a convolution operation. W_r represents the corresponding weights in the convolution. The sigmoid function σ converts each value in the feature tensor into the range of $(0, 1)$ and acts as a gate signal.

Update Gate The update gate z is also decided by the hidden state h_p and input features f^d. Through another convolution operation with weight W_z and the sigmoid function σ, we get z as,

$$z = \sigma\left(W_z\left(f^d, h_p\right)\right) \tag{9.2}$$

Theoretically, reset and update gates essentially learn a set of weights that control the amount of information that is retained or discarded, experimental studies in section 9.4 show the effectiveness of the reset and update gates.

Adaptive Memory Through the element-wise product $\odot$, the reset gate r determines how much

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
Zheng *et al.* (Zheng et al. [2013])	60.1	46.7	34.6	-	-	-	-	-	-	-	-	-	-	-	-
Firman *et al.* (Firman et al. [2016])	66.5	69.7	50.8	-	-	-	-	-	-	-	-	-	-	-	-
SSCNet Song et al. [2017]	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	33.9	57.0	**59.5**	28.3	8.1	44.8	25.1	40.0
TS3D Garbade et al. [2018]	80.2	91.0	74.2	33.8	**92.9**	46.8	**27.0**	27.9	**61.6**	51.6	27.6	**26.9**	44.5	22.0	42.1
DDR-SSC Li et al. [2019]	**88.7**	88.5	79.4	**54.1**	91.5	56.4	14.9	37.0	55.7	51.0	28.8	9.2	44.1	27.8	42.8
GRFNet	87.2	91.0	**80.1**	50.3	91.8	**58.1**	18.4	**42.7**	60.6	52.8	**34.6**	11.5	**46.6**	**30.8**	**45.3**

Table 9.2: Results on the NYUCAD dataset Zheng et al. [2013]. Bold numbers represent the best scores.

information in the past needs to be "memorized".

$$h'_p = r \odot h_p \tag{9.3}$$

when the reset gate r is close to 1, the "memorized" information h'_p will be kept and then passed to the current fusion operation with current input feature f^d. The preserved "memory" h'_p and feature f^d are concatenated together to perform a linear transformation (convolution), and then activated by a tanh function.

$$h^c_p = \tanh\left(W_h\left(f^d, h'_p\right)\right) \tag{9.4}$$

h^c_p acts similarly to the memory cell in the LSTM and helps the GRF fusion block to remember long term information within the multi-stage fusion.

Selective Fusion $z \odot h_p$: Indicates how much of the previous features should be preserved. $(1-z)\odot h^c_p$: Indicates how much of the current information h^c_p should be added. Similar to the former, here $(1-z)$ forgets some unimportant information in h^c_p. Or, it can be viewed as a choice of some information in h^c_p.

Combined with f^d and h_p, the fusion result at step p is,

$$h_q = z \odot h_p + (1-z) \odot h^c_p \tag{9.5}$$

This operation ignores some information in previous hidden state h_p, and adds some information from the current step. Update gate z is equivalent to the *forget gate* in LSTM, and $1-z$ is equivalent to the *input gate* in LSTM. In this way, the *forget gate* z and the *input gate* $(1-z)$ are linked. That is, if the previous information is ignored with a weight of z, then the information for the current input h^c_p would be selected with a weight of $(1-z)$. In our case, if the information in previous stage is depth feature and current input is RGB feature, this enables the complementary information to be effectively merged. Accordingly, the output of the current step $f^{GRF}_q = h_p$ will also be passed to the next step.

Single-stage Fusion Module The bimodal information passes through multiple DDRs for feature extraction, and each process of the DDR corresponds to a stage. That is, single-stage fusion module has only one layer of DDR from RGB and depth branch, and the fusion block is performed after the DDR. In specific, the GRF module has two input features, f^d and f^{rgb}. At step 1, the hidden state is initialised as mentioned above. We feed f^d into GRF fusion block, and get the output h_1. Then at step 2, we reuse the same structure and the same parameters in the GRF fusion block. The input hidden

NYU	scene completion			semantic scene completion											
method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
single-stage GRFNet	66.5	**85.9**	60.1	**27.5**	**92.9**	28.1	10.7	14.9	**60.1**	33.8	**17.3**	10.1	30.4	14.7	31.0
multi-stage GRFNet	**68.4**	85.4	**61.2**	24.0	91.7	**33.3**	**19.0**	**18.1**	51.9	**45.5**	13.4	**13.3**	**37.3**	**15.0**	**32.9**
NYUCAD	scene completion			semantic scene completion											
single-stage GRFNet	**88.4**	89.1	79.7	50.0	91.4	56.4	**18.7**	41.3	56.8	52.7	33.5	**16.3**	45.2	30.0	44.8
multi-stage GRFNet	87.2	**91.0**	**80.1**	**50.3**	**91.8**	**58.1**	18.4	**42.7**	**60.6**	**52.8**	**34.6**	11.5	**46.6**	**30.8**	**45.3**

Table 9.3: Results of single-stage GRFNet and multi-stage GRFNet on both NYU and NYUCAD dataset.

Method (NYU)	GS	MM	prec.	recall	IoU	mIoU
Concatenate Fusion			70.6	76.2	57.6	25.9
Sum Fusion			67.6	79.4	57.6	25.7
Max Fusion			67.6	79.4	57.5	25.6
Gated Fusion	✓		**70.8**	77.5	58.6	27.6
LSTM Fusion	✓	✓	68.0	82.3	59.6	28.3
GRF Fusion	✓	✓	66.5	**85.9**	**60.1**	**31.0**
Method (NYUCAD)	GS	MM	prec.	recall	IoU	mIoU
Concatenate Fusion			87.3	83.5	74.3	37.8
Sum Fusion			81.4	89.3	74.3	37.7
Max Fusion			81.9	87.8	73.3	36.5
Gated Fusion	✓		82.1	**91.3**	76.0	40.2
LSTM Fusion	✓	✓	83.5	**91.3**	77.5	41.4
GRF Fusion	✓	✓	**88.4**	89.1	**79.7**	**44.8**

Table 9.4: Results of different single-stage fusion methods on the NYU and NYUCAD dataset. GS denotes Gate Structure, and MM represents Memory Mechanism. With IoU denotes the accuracy of semantic completion and mIoU denotes the accuracy of semantic scene completion.

state is replaced by h_1, and the input is the features f^{rgb} extracted from the RGB image. As shown in Figure 9.4, we use the red line with an arrow to indicate the reuse of GRF fusion block at step 2.

Multi-stage GRF Fusion Module Features extracted by the earlier-stage DDR are at relatively low-level, while those by the later-stage DDR are at relatively high-level regarding to the semantic meaning representation. For multi-stage fusion, the features of the two modal data at each stage will be formed as a sequence. Taking the N-stage RGB-D fusion as an example, the feature sequence is $F = \left(f_1^d, f_1^{rgb}, f_2^d, f_2^{rgb}, \cdots, f_N^d, f_N^{rgb}\right)$. Each feature tensor in F will be fed into the GRF fusion block serially. The GRU fusion block will be reused $2N$ times and all these fusion stages share the same group of parameters. Different with the single-stage fusion module which performs the multi-modal feature fusion at only one stage of the network, multi-modal features are fused in multi-stages which covers both of the high-level and low-level features. It is not only helpful to recover the details of the scene, but also important to propagate information among different stages.

Using the low-level feature to guide high-level feature is a common and reasonable approach in computer vision community. The proposed GRF module can preserves the complementary information and compensates the missing details. Particularly, the color and texture details in RGB image as well as the geometry and distance information in depth are complementary to each other. The "gate" structure in the GRF fusion block controls the feature fusion by learning weights (between 0 and 1). Moreover, for multi-stage fusion, the GRF fusion block has the potential to manage the interleaved modalities. To be specific, the bimodal information has gradually transformed into abstract semantic information through the network, and the difference between their distributions is gradually reduced. Due the above merits, we employ the multi-stage GRF fusion module in our network. The effectiveness of multi-stage fusion module are supported and reflected by our experiments in section 9.4.

9.3.3 Training Protocol

The loss function used in our training process is the softmax cross-entropy loss, and it is performed on the unnormalized network outputs y:

$$\mathcal{L} = -\sum_{c=0}^{N} w_c \hat{y}_{i,c} \log\left(\frac{e^{y_{ic}}}{\sum_{c'}^{N} e^{y_{ic'}}}\right) \tag{9.6}$$

where $\hat{y}_{i,c}$ are the one-hot ground truth vectors, *i.e.* $\hat{y}_{i,c} = 1$ if voxel i is labeled by class c, otherwise $\hat{y}_{i,c} = 0$. N is the number of classes, and w_c is the loss weight, for balancing different classes, and the setting following SSCNet Song et al. [2017]. To compute the loss function, we ignore all voxels outside the field of view but include all voxels inside the view (empty, non-empty and occluded voxels).

We train the network from scratch with the initial learning rate 0.01 which is reduced by a factor of 0.1 after every ten epochs. We set the weight of empty voxels w_0 to 0.05 for data balancing and increase it by 0.05 for every 40 training epochs. Our model is trained using the SGD optimizer with a momentum of 0.9, weight decay of 10^{-4} and batch size is 4.

Please note, the order in which the modalities are fed into GRF fusion block has been always fixed (fist Depth, then RGB) for all of the experiments, however, according to our preliminary experiments, using different input order (first RGB, then Depth) has a minor impact on the performance.

9.4 Experiments

9.4.1 Datasets and Metrics

Datasets We evaluate the proposed method and compare it with the state-of-the-art methods on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] datasets. NYU consists of 1449 indoor scenes, including 795 training samples and 654 testing samples. The RGBD images of NYU are captured via a Kinect RGBD sensor, and the 3D semantic scene completion labels are from Rock *et al*. (Rock et al. [2015]). The annotations are fitted into the scenes by CAD models. NYUCAD uses the depth maps generated from the projections of the 3D annotations to reduce the misalignment of depths and the annotations.

Metrics The primary evaluation metric is the voxel-level intersection over union (IoU) between the predicted labels and ground-truth labels. For semantic scene completion, the IoU is calculated for each category on both the observed and occluded voxels. For scene completion, all non-empty classes are treated as one category, IoU, precision, and recall of the binary predictions are evaluated on the occupied voxels.

9.4.2 Comparisons with the State-of-the-art Methods

In the task of semantic scene completion, our GRFNet outperforms all existing methods and achieves the start-of-the-art accuracy. The results on NYU Silberman et al. [2012] and NYUCAD Firman et al. [2016] are shown in Table 9.1 and Table 9.2, respectively. The GRFNet improves the average IoU over DDR-SSC Li et al. [2019] by 2.5% on both NYU and NYUCAD datasets.

The experiments demonstrate that the proposed fusion block effectively utilizes multi-modality information. Since we focus on presenting a practical multi-modal data fusion approach (GRF), we maintain the consistency of the network structure for a fair comparison to prove that the improvement in accuracy comes from the GRF fusion block. In specific, our network framework is the same as DDR-SSC except for the fusion block.

9.4.3 Quantitative Analysis

Table 9.1 shows the quantitative results on NYU dataset Silberman et al. [2012] acquired by our method and other state-of-the-art methods. Approaches of Lin *et al.* (Lin et al. [2013]) and Geiger *et al.* (Geiger and Wang [2015]) are traditional methods. SSCNet Song et al. [2017], EsscNet Zhang et al. [2018b], and DDR-SSC Li et al. [2019] are CNN-based approaches. Compared to the classical approach SSCNet, the IoUs of GRFNet increase 6.1% and 8.2% for SC and SSC tasks, respectively. In the SSC task, our GRFNet gets 6.2% higher accuracy than EsscNet and achieves higher IoU in almost every category. SSCNet and EsscNet only use depth information, while DDR-SSC uses a multi-stage fusion structure to take advantage of the depth and RGB images. Our GRFNet also uses RGB-D information and achieves a 2.5% higher average IoU than DDR-SSC. Regarding the individual class accuracy, the IoUs for each category are also listed out in Table 9.1.

As shown in Table 9.2, GRFNet achieves outstanding performance on NYUCAD dataset as well. Specifically, compared to Zheng *et al.*'s(Zheng et al. [2013]) and Firman *et al.*'s(Firman et al. [2016]) methods, GRFNet significantly improves the accuracy in two metrics. Since SSCNet only employs depth as input, the proposed GRFNet which use RGB and depth information achieves much more accurate results. Although TS3D Garbade et al. [2018] and DDR-SSC use both RGB and depth information, these methods only adopt simple fusion strategy. On the contrary, GRFNet benefited from the novel fusion block, to obtain 0.7% and 2.5% improvements compared to DDR-SSC, and 5.9% and 3.2% improvements compared to TS3D for SC and SSC tasks respectively. In summary, our approach achieves higher accuracy on most indicators than previous methods, especially the average IoU, which reflects the overall performance.

9.4.4 Qualitative Analysis

Figure 9.5 visualizes results of the semantic scene completion generated by the proposed GRFNet (c), DDR-SSC (d) and SSCNet (e). We mark the difference in visual quality with a red dotted box for reference. As can be seen, compared with both SSCNet and DDR-SSC, the scene completion results of our GRFNet are much more abundant in detail and less error-prone. More visualization results and analyses are provided in the supplemental materials.

Method	Prec.	Recall	IoU	mIoU
Sum Fusion(DDR-SSC)	**71.5**	80.8	61.0	30.4
LSTM Fusion	68.0	83.2	60.2	30.2
GRF Fusion	68.4	**85.4**	**61.2**	**32.9**

Table 9.5: Results of different multi-stage fusion methods on NYU dataset. With IoU represents the accuracy of scene completion, and mIoU denotes the accuracy of semantic scene completion.

9.4.5 Ablation Study

To study the effect of different components and design choices we perform an ablation study. We choose DDR-SSC Li et al. [2019] as the baseline, which is the most relevant work to the proposed GRFNet. Since our focus is the fusion strategy, the GRF module will be analyzed in detail below.

Single-stage Fusion To verify the effectiveness of our GRF fusion module, we compare the single-stage GRFNet with a variety of conventional fusion methods, including Concatenate Fusion, Sum Fusion, Max Fusion, and Gated Fusion. For better comparison, we replace the fusion block in the framework (as shown in Figure 9.3) by the compared single-stage fusion methods. The results of the comparison are shown in Table 9.4. We have two findings as following: 1) The fusion strategy using the gate structure is better than the one without the gate structure; 2) The memory mechanism can further enhance fusion effects.

As shown in Table 9.4, Sum Fusion, Max Fusion, and Concatenate Fusion achieve similar performance. And they are significantly lower than the other three modules in which contain adaptive selection mechanism. LSTM fusion and GRF Fusion have a memory mechanism, but Gated Fusion does not; therefore, the accuracy of the first two methods is better. LSTM is more complex and has more parameters than GRF. However, GRF Fusion is still 2.7% and 3.4% more accurate than LSTM on NYU and NYUCAD regarding to SSC accuracy, respectively.

Multi-stage Fusion

1). **Multi-stage Strategy** In Table 9.3, we compare the performance of single-stage GRFNet and the multi-stage GRFNet. Single stage-fusion can only fuse information at one of the network stages. While multi-stage GRFNet can use both the low-level and high-level information and get higher accuracy than the single-stage version on both datasets.

2). **Fusion Strategy** In DDR-SSC Li et al. [2019], sum fusion is used to fuse the four stages of features separately. The fusion results are cascaded and handed over to subsequent networks for semantic label prediction. GRF module employs the recurrent structure that uses the previous fusion results as the input for the next fusion stage, hence the information for each stage can be combined without additional cascading operations. As can been in Table 9.5, multi-stage GRFNet gets 0.2% higher average IoU than DDR-SSC on SC and 2.5% higher on SSC. And GRF fusion is 1% higher than LSTM fusion on SC and 2.7% higher on SSC.

Parameters and Flops of Different Fusion Stages

In Figure 9.6, parameters and FLOPs of our network with different fusion stages are listed out. As can be seen, with the increasing of fusion stages, parameters increase slightly, which mainly due to the reuse of GRF fusion block, the only source for more parameters is the new added DDR block (for both Depth and RGB channel). On the contrast, FLOPs increase dramatically, which mainly come from

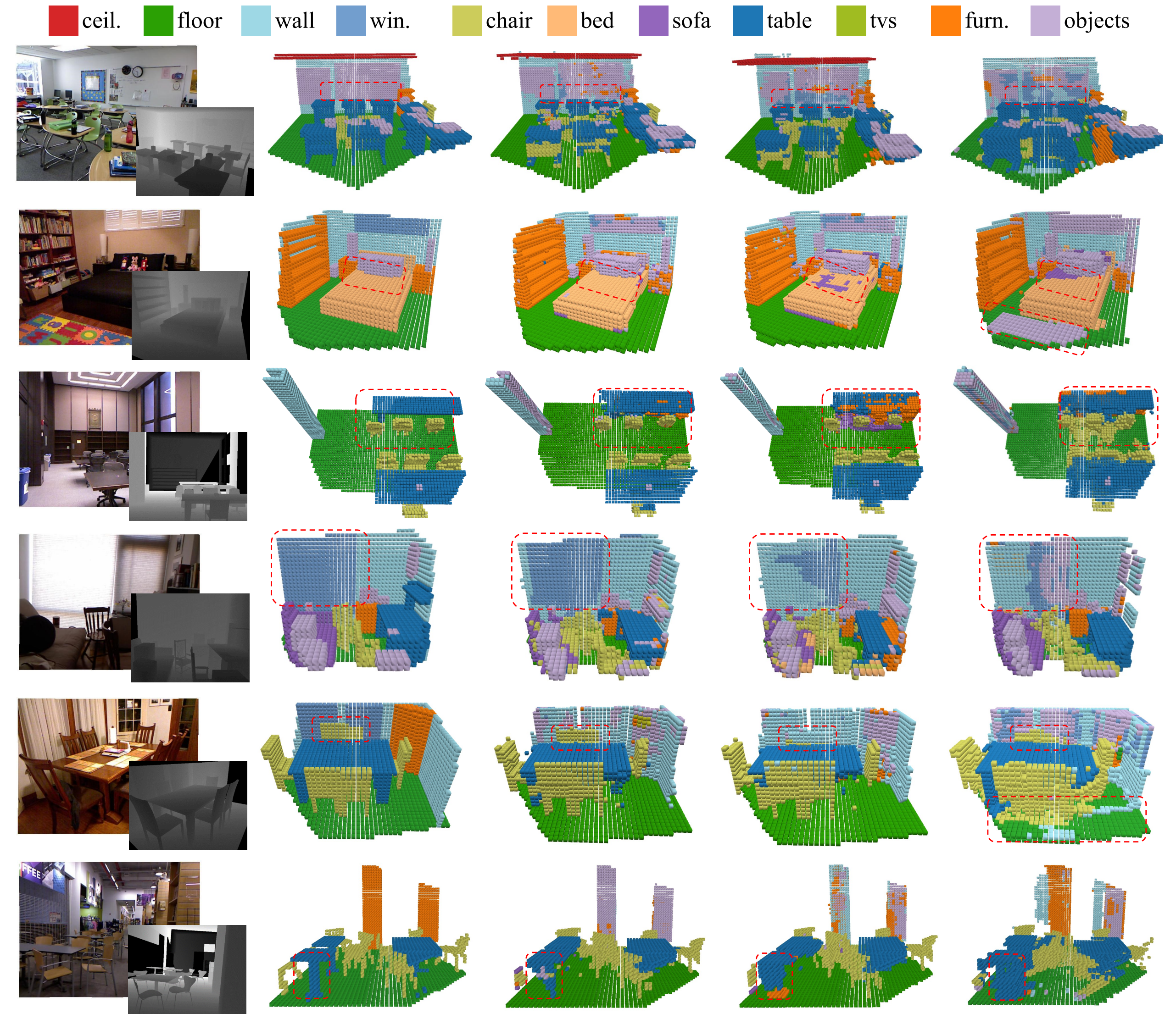
Fusion stages	Params [k]	FLOPs [G]
1	794.59	193.47
2	803.39	366.65
3	812.19	539.84
4	820.99	713.02

Table 9.6: Params and FLOPs of multi-stage GRFNets with different number of fusion stages.

the GRF fusion block and small portion come from DDR blocks. In our implementation, GRF fusion block still employ 3D convolutions, thus bring in relatively high compution costs. As we point out before, our focus of this work is to provide a new strategy for fusing the two-modal data in SSC, which can be improved by light-weight operations.

9.5 Conclusion

In this paper, we propose GRFNet with a novel gated recurrent fusion module to fuse RGB and depth information. Different from the existing fusion strategies, we emphasize the importance of the adaptive selectivity of information and the memory mechanism within the fusion block. Moreover, we further extend the single-stage GRFNet to a multi-stage version, which can fuse both low-level and high-level feature at different stages. Our approach has significant advantages over previous methods in multi-modal data fusion and achieves the state-of-the-art performance in semantic scene completion. Extensive comparison experiments and ablation studies verify the effectiveness of the proposed method. In the future, one of our research interests would be to consider making the proposed GRFNet light-weight, for instance, replacing the 3D convolution of GRF fusion block with DDR.



(a) RGB and Depth images (b) Ground truth (c) GRFNet (d) DDR-SSC (e) SSCNet

Figure 9.5: Qualitative results on NYUCAD. From left to right: Input RGB-D image, ground truth, results generated by our GRFNet, DDR-SSC Li et al. [2019], and SSCNet Song et al. [2017]. Overall, our completed semantic 3D scenes are less cluttered and show a higher voxel class accuracy compared to the others.

Bibliography

N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. *arXiv:1511.06432*, 2015.

L. Chen, K. Francis, and W. Tang. Semantic augmented reality environment with material-aware physical interactions. In *Proc. IEEE ISMAR*, pages 135–136. IEEE, 2017a.

X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, pages 1907–1915, 2017b.

Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, pages 3029–3037, 2017.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *NeurIPS*, pages 577–585, 2015.

C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013.

A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, pages 4578–4587, 2018.

A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid. Scalable place recognition under appearance change for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9319–9328, 2019.

E. Emre Yurdakul and Y. Yemez. Semantic segmentation of rgbd videos with recurrent fully convolutional neural networks. In *ICCV*, pages 367–374, 2017.

M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, pages 5431–5440, 2016.

H. Fu, D. Xu, and S. Lin. Object-based multiple foreground segmentation in rgbd video. *TIP*, 26(3): 1418–1427, 2017.

M. Garbade, J. Sawatzky, A. Richard, and J. Gall. Two stream 3d semantic scene completion. *arXiv:1804.03550*, 2018.

A. Geiger and C. Wang. Joint 3d object and layout inference from a single rgb-d image. In *German Conference on Pattern Recognition*, pages 183–195, 2015.

Y. Guo and T. Chen. Semantic segmentation of rgbd images based on deep depth regression. *Pattern Recognition Letters*, 109:55–64, 2018.

S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, pages 345–360. Springer, 2014.

X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *ICCV*, 2017.

C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *ACCV*, pages 213–228. Springer, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

J.-F. Hu, W.-S. Zheng, J. Pan, J. Lai, and J. Zhang. Deep bilinear learning for rgb-d action recognition. In *ECCV*, pages 335–351, 2018.

E. P. Ijjina and K. M. Chalavadi. Human action recognition in rgb-d videos using motion sequence information and deep learning. *Pattern Recognition*, 72:504–516, 2017.

L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740, 2014.

C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *IROS*, pages 2100–2106. IEEE, 2013.

Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

J. Li, Y. Liu, D. Gong, Q. Shi, X. Yuan, C. Zhao, and I. Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019.

Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, pages 541–557. Springer, 2016.

D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, pages 1417–1424, 2013.

T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015.

Y. Lu and D. Song. Robust rgb-d odometry using point and line features. In *ICCV*, pages 3934–3942, 2015.

S. J. Park, K. S. Hong, and S. Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017.

X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, pages 2759–2766, 2012.

J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *CVPR*, pages 2484–2493, 2015.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198, 2017.

M. Sultana, A. Mahmood, S. Javed, and S. K. Jung. Unsupervised rgbd video object segmentation using gans. *arXiv:1811.01526*, 2018.

J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. In *IROS*, pages 2442–2447, 2017.

J. Wang, Z. Wang, D. Tao, S. See, and G. Wang. Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks. In *ECCV*, pages 664–679, 2016.

N. Wang and X. Gong. Adaptive fusion for rgb-d salient object detection. *arXiv:1901.01369*, 2019.

T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald. Robust real-time visual odometry for dense rgb-d mapping. In *ICRA*, 2013.

J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.

H. Zhang, Y. Li, P. Wang, Y. Liu, and C. Shen. Rgb-d based action recognition with light-weight 3d convolutional networks. *arXiv preprint arXiv:1811.09908*, 2018a.

J. Zhang, H. Zhao, A. YaoE, Y. Chen, L. Zhang, and H. LiaoE. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, pages 733–749, 2018b.

B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, pages 3127–3134, 2013.

Chapter 10

Conclusion and Future Directions

This chapter provides a summary of the work presented, and a brief discussion of some future directions following up on those discussed in the previous chapters.

This thesis is composed of a body of work that taking use of deeep learning for robot perception applications, in contrast with the previous methods, which mainly care about the performance of network regardless how heavy the network is and how fast the inference is, we cover topics of object detection, video segmentation, semantic scene completion, with the focus on handling heavy occlusion, fast inference, quick adaptation, data imbalance, modal fusion etc, which provide a reference for other works that put their efforts in robot perception.

In particular, we first propose a Pairwise-NMS as the post-processing step for object detection. Feeding two overlapping proposals to a pairwise-relationship network can smartly tell the Pairwise-NMS if there are two objects or one/zero object contained. It can cure the drawback of GreedyNMS for "suppressing all" under heavy occlusions, and is more robust than Soft-NMS in crowded scenes. Moreover, Pairwise-NMS can consistently improve the performance and neatly couple with GreedyNMS. And can be as an ingredient to be integrated into learning-based detectors including Faster-RNN, DPM without losing the efficiency. We believe that our work is beneficial to instance segmentation and multi-object tracking in crowded scenes. In the future, there will be two research problems of our interests. One of the problems is to integrate Pairwise-NMS into learning based detectors such as Faster-RCNN, YOLO, SSD to perform joint training and inference, and the other is to explore a more general rule to handle a cluster of multiple objects at the same time.

Next, we explore applying meta-learning into video object segmentation system. A closed form optimizer, i.e., ridge regression, is utilized to update the meta learner, which achieves fast speed while maintains the superior accuracy. Through iteratively meta-learned, the network is capable of conducting *fast mapping* on unseen objects with a few examples available. Compared to the fine-tuning methods, our algorithm with similar performance but just a smaller fraction time is required, which is appeal to the real-world applications. In addition, a block splitting mechanism is delivered to speed up the training process, which also has the benefits of reducing parameters and saving memory. In future work, we would like to use other basic optimizers, such as, Newton's methods and logistic regression. Meanwhile, based on the flexible design of our meta-learner, instead of inferring the rest frames from the given whole annotation of the first frame. Inferring whole object from only part of annotation or user feedback is also worth to investigate.

In another work, we deliver a common module, video loss, for video object segmentation, which is tailored to overcome the limitation of fine-tuning methods, during the phase of training *parent network*, dilute the instance information, hence delay the overall training process. Considering in CNN, the shallow layers usually contain much rich details of object(s) which are the key cues to specify different instances, while the deeper layers have more stronger generalization ability to recognize generic objects. Various video losses are proposed as the constraints to supervise the training process of *parent network*, which is effective in removing the noisy objects. Once the training process is finished, the *parent network* is well prepared to adapt to the instance quickly during *online fine-tuning*. One of our future interests will be extending the video loss into other fine-tuning methods such OSVOS-S, OnVOS. Another one will be with the help of the network search technique to automatically decide the training epochs and learning rate.

Targeting at handling unsupervised domain adaptation (UDA) problem, a novel method, which is composed of three components, is proposed. Firstly, in contrast to the previous methods, which assume there is a distribution center in the embedding space for both of the source domain and the target domain, we instead use the clustering method, that are k exemplars work together to simulate the distribution center of each class. Secondly, class-balanced self-training strategy is utilized for generating pseudo labels in the target domain, which will be incorporated in the training loop for capturing much more appearance information from the target domain. Thirdly, in order to better align the features from the source domain and target domain, serial adapter and parallel adapter are employed to achieve this purpose. At the moment, we fix the number of exemplars within each class as five, which given much better performance than using single exemplar per class. In the future, smartly choose the number of exemplars within each class would be one of our interests.

We also proposes a novel structure for handling the semantic scene completion problem. Specifically, an end-to-end light-weight Dimensional Decomposition Residual (DDR) network is delivered for scene completion and semantic scene labeling. The two contributions are the proposed factorized convolution layer and a novel two-modality fusion mechanism. The former is effective to reduce the parameters within the network, and the later can fuse the depth and color image seamlessly in multi-level, the state-of-the-art results are achieved for both SSC and SC task on two public datasets. In the future, considering to differentiate instances of the indoor scene as well as to incorporate the shuffle layer into the proposed light-weight network will be our research interests.

We introduce PALNet which takes both the depth and TSDF as inputs for semantic scene completion. The feature from 2D-stream are projected and concatenated with the feature in 3D-stream and trained in an end-to-end manner. We also propose a position importance aware loss, PA-Loss, which leads to slightly faster training convergence and better performance. The experiments on both synthetic and real datasets validate the superior performance of the proposed method. The two interesting topics that are worth exploring in our future work include: (1) a better trade-off between 2D CNNs and 3D CNNs to employ less 3D convolutional layers without sacrificing the performance, and (2) to extend our PALNet framework with RGB-D as input.

Lastly, we propose GRFNet with a novel gated recurrent fusion module to fuse RGB and depth information. Different from the existing fusion strategies, we emphasize the importance of the adaptive selectivity of information and the memory mechanism within the fusion block. Moreover, we further extend the single-stage GRFNet to a multi-stage version, which can fuse both low-level and high-level feature at different stages. Our approach has significant advantages over previous methods in multi-modal data fusion and achieves the state-of-the-art performance in semantic scene completion. Extensive comparison experiments and ablation studies verify the effectiveness of the proposed method. In the future, one of our research interests would be to consider making the proposed GRFNet light-weight, for instance, replacing the 3D convolution of GRF fusion block with DDR.