DISS. ETH NO. 30019

ADVANCED ATTENTION MECHANISMS FOR DENSE PREDICTION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by

GUOLEI SUN
Master of Science in King Abdullah University of Science and Technology (KAUST)

born on 30 Mar 1993

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Serge Belongie, co-examiner
Prof. Dr. Efstratios Gavves, co-examiner
Dr. Danda Pani Paudel, co-examiner

2024

ABSTRACT

In the era of deep learning, significant progresses have been witnessed for dense prediction tasks such as image/video segmentation, object counting, and depth estimation. However, dense prediction under challenging scenarios is far from being solved and performances of existing algorithms on those situations are far from satisfaction. Those challenging situations include: 1) understanding camouflaged scenes where the foreground is blended in the background and thus difficult to differentiate; 2) learning from weak supervision/annotation; 3) comprehending long-term dynamic scenes (videos). In this thesis, we focus on these challenging scenarios for dense prediction. Motivated by the power of attention mechanisms in various language and vision tasks, we focus on developing methods based on advanced attention mechanisms.

First, we discuss camouflaged/indiscernible object counting. Indiscernible scene understanding has attracted a lot of attention in the vision community. We further advance the frontier of this field by systematically studying a new challenge named indiscernible object counting (IOC), the goal of which is to count objects that are blended with respect to their surroundings. Due to a lack of appropriate IOC datasets, we present a large-scale dataset IOCfish5K which contains a total of 5,637 high resolution images and 659,024 annotated center points. IOCfish5K is superior to existing datasets with indiscernible scenes because of its larger scale, higher image resolutions, more annotations, and denser scenes. All these aspects make it the most challenging dataset for IOC so far, supporting progress in this area. For benchmarking purposes, we select 14 mainstream methods for object counting and carefully evaluate them on IOCfish5K. Furthermore, we propose IOCFormer to combine density and regression branches in a unified framework, a new strong baseline that exploits global and local attention. It can effectively tackle object counting under indiscernible scenes. Experiments show that IOCFormer achieves state-of-the-art scores on IOCfish5K.

Second, we discuss weakly supervised semantic segmentation, where only weak supervisions (image-level labels) are available for training. Current popular solutions leverage object localization maps from classifiers as supervision signals, and struggle to make the localization maps capture more complete object content. Rather than previous efforts that primarily focus on intra-image information, we address the value of cross-image

semantic relations for comprehensive object pattern mining by developping advanced attention mechanisms. To achieve this, two neural co-attentions are incorporated into the classifier to complimentarily capture cross-image semantic similarities and differences. This helps the classifier discover more object patterns and better ground semantics in image regions. In addition to boosting object pattern learning, the co-attention can leverage context from other related images to improve localization map inference, hence eventually benefiting semantic segmentation learning. More essentially, our algorithm provides a unified framework that handles well different WSSS settings, i.e., learning WSSS with (1) precise image-level supervision only, (2) extra simple single-label data, and (3) extra noisy web data. It sets new state-of-the-arts on all these settings, demonstrating well its efficacy and generalizability. Moreover, our approach ranked 1st place in the Weakly-Supervised Semantic Segmentation Track of CVPR2020 Learning from Imperfect Data Challenge.

Third, we discuss dynamic scene (video) semantic segmentation by using multi-frames attentions. The contextual information plays a core role in segmentation. As for video semantic segmentation, the contexts include static contexts and motional contexts, corresponding to static content and moving content in a video clip, respectively. Both static and motional contexts are studied in previous works. However, there is no research about how to simultaneously learn static and motional contexts which are highly correlated and complementary to each other. To address this problem, we propose a Coarse-to-Fine Feature Mining (CFFM) technique to learn a unified presentation of static contexts and motional contexts. This technique consists of two parts: coarse-to-fine feature assembling and cross-frame feature mining. The former operation prepares data for further processing, enabling the subsequent joint learning of static and motional contexts. The latter operation mines useful information from the sequential frames to enhance the features of the target frame by non-self attention mechanism. The enhanced features can be directly applied for the final prediction. Experimental results on popular benchmarks demonstrate that the proposed CFFM performs favorably against state-of-the-art methods.

Last, we discuss video semantic segmentation by mining hyper-relations among multi-frames attentions. Previous efforts and CFFM are mainly devoted to exploiting new techniques to calculate the cross-frame affinities such as optical flow and attention. Instead, this work contributes from a different angle by mining relations among cross-frame affinities, upon which better temporal information aggregation could be achieved. We explore

relations among affinities in two aspects: single scale intrinsic correlations and multi-scale relations. Inspired by traditional feature processing, we propose Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). At last, the cross-frame affinities strengthened by SAR and MAA are adopted for adaptively aggregating temporal information. Our experiments show that the proposed method outperforms state-of-the-art VSS methods by clear margins.

ZUSAMMENFASSUNG

Im Zeitalter des Deep Learning wurden signifikante Fortschritte bei dichten Vorhersageaufgaben wie der Bild-/Video-Segmentierung, der Objektzählung und der Tiefenschätzung verzeichnet. Dennoch ist die dichte Vorhersage unter herausfordernden Szenarien weit davon entfernt, gelöst zu sein, und die Leistungen bestehender Algorithmen in solchen Situationen lassen oft zu wünschen übrig. Diese herausfordernden Situationen umfassen: 1) das Verständnis getarnter Szenen, in denen der Vordergrund mit dem Hintergrund verschmilzt und daher schwer zu unterscheiden ist; 2) das Lernen aus schwacher Überwachung/Kennzeichnung; 3) das Verstehen von langfristigen dynamischen Szenen (Videos). In dieser Arbeit konzentrieren wir uns auf diese herausfordernden Szenarien für die dichte Vorhersage. Angeregt durch die Leistungsfähigkeit von Aufmerksamkeitsmechanismen in verschiedenen Sprach- und Bildaufgaben konzentrieren wir uns auf die Entwicklung von Methoden, die auf fortschrittlichen Aufmerksamkeitsmechanismen basieren.

Zunächst diskutieren wir die Zählung getarnter/undeutlicher Objekte. Das Verständnis undeutlicher Szenen hat in der Vision-Community viel Aufmerksamkeit erregt. Wir treiben die Grenzen dieses Bereichs weiter voran, indem wir systematisch eine neue Herausforderung namens "undeutliche Objektzählung" (IOC) untersuchen, deren Ziel es ist, Objekte zu zählen, die im Hinblick auf ihre Umgebung verschmelzen. Aufgrund eines Mangels an geeigneten IOC-Datensätzen präsentieren wir einen umfangreichen Datensatz IOCfish5K, der insgesamt 5.637 hochauflösende Bilder und 659.024 annotierte Zentrumspunkte enthält. IOCfish5K ist aufgrund seiner größeren Skala, höheren Bildauflösungen, mehr Annotationen und dichteren Szenen den bestehenden Datensätzen mit undeutlichen Szenen überlegen. All diese Aspekte machen ihn bisher zum anspruchsvollsten Datensatz für IOC und unterstützen den Fortschritt in diesem Bereich. Für Benchmarking-Zwecke wählen wir 14 gängige Methoden für die Objektzählung aus und bewerten sie sorgfältig anhand von IOCfish5K. Darüber hinaus schlagen wir IOCFormer vor, um Dichte- und Regressionszweige in einem einheitlichen Rahmen zu kombinieren, eine neue starke Baseline, die globale und lokale Aufmerksamkeit nutzt. Sie kann die Objektzählung in undeutlichen Szenen effektiv bewältigen. Experimente zeigen, dass IOCFormer Spitzenwerte auf IOCfish5K erreicht.

Zweitens diskutieren wir die schwach überwachte semantische Segmentierung, bei der nur schwache Überwachungen (Bildniveauetiketten) für das Training verfügbar sind. Aktuelle beliebte Lösungen nutzen Objektlokalisierungskarten von Klassifikatoren als Überwachungssignale und kämpfen darum, dass die Lokalisierungskarten mehr vollständigen Objektinhalt erfassen. Anstatt sich wie bisher hauptsächlich auf intraimage Informationen zu konzentrieren, adressieren wir den Wert von kreuzbildlichen semantischen Beziehungen für umfassendes Objektmustermining durch die Entwicklung fortschrittlicher Aufmerksamkeitsmechanismen. Um dies zu erreichen, werden zwei neuronale Co-Aufmerksamkeiten in den Klassifikator integriert, um komplementär kreuzbildliche semantische Ähnlichkeiten und Unterschiede zu erfassen. Dies hilft dem Klassifikator, mehr Objektmuster zu entdecken und Semantik besser in Bildregionen zu verankern. Neben der Verbesserung des Lernens von Objektmustern kann die Co-Aufmerksamkeit Kontexte aus anderen verwandten Bildern nutzen, um die Inferenz von Lokalisierungskarten zu verbessern und letztendlich das Lernen der semantischen Segmentierung zu verbessern. Darüber hinaus bietet unser Algorithmus einen einheitlichen Rahmen, der verschiedene Einstellungen für WSSS gut behandelt, d. H. das Lernen von WSSS mit (1) präzisen Bildniveau-Überwachungen nur, (2) zusätzlichen einfachen Einzelbeschriftungsdaten und (3) zusätzlichen rauschigen Webdaten. Es setzt neue State-of-the-Arts in all diesen Einstellungen, was seine Wirksamkeit und Generalisierbarkeit gut demonstriert. Darüber hinaus belegte unser Ansatz den ersten Platz im Track für schwach überwachte semantische Segmentierung der CVPR2020 Learning from Imperfect Data Challenge.

Drittens diskutieren wir die semantische Segmentierung dynamischer Szenen (Videos) durch Verwendung von Mehrbildaufmerksamkeiten. Kontextinformationen spielen eine Kernrolle bei der Segmentierung. Was die semantische Segmentierung von Videos betrifft, so umfassen die Kontexte statische Kontexte und Bewegungskontexte, die statische Inhalte und bewegliche Inhalte in einem Videoclip entsprechen. Beide statischen und bewegten Kontexte wurden in früheren Arbeiten untersucht. Es gibt jedoch keine Forschung darüber, wie gleichzeitig statische und bewegte Kontexte gelernt werden können, die hoch korreliert und komplementär zueinander sind. Um dieses Problem anzugehen, schlagen wir eine Technik namens "Coarse-to-Fine Feature Mining" (CFFM) vor, um eine einheitliche Darstellung von statischen und bewegten Kontexten zu lernen. Diese Technik besteht aus zwei Teilen: grob-zu-feiner Merkmalssammlung und Kreuzbild-Feature-Mining. Die erstere Operation bereitet Daten für die weitere Verarbeitung vor, was das nachfolgende gemeinsame Lernen von statischen und bewegten Kontexten ermöglicht. Die letztere Operation fördert nützliche Informationen aus den sequentiellen Bildern, um die Merkmale des Zielbildes durch einen Nicht-Selbst-Aufmerksamkeitsmechanismus zu verbessern. Die verbesserten Merkmale können direkt für die endgültige Vorhersage angewendet werden. Experimentelle Ergebnisse auf gängigen Benchmarks zeigen, dass das vorgeschlagene CFFM gegenüber State-of-the-Art-Methoden deutlich besser abschneidet.

Zuletzt diskutieren wir die semantische Segmentierung von Videos durch das Abbauen von Hyperbeziehungen zwischen Mehrbildaufmerksamkeiten. Frühere Anstrengungen und CFFM sind hauptsächlich darauf ausgerichtet, neue Techniken zur Berechnung der kreuzbildlichen Affinitäten wie optischer Fluss und Aufmerksamkeit zu nutzen. Stattdessen trägt diese Arbeit aus einem anderen Blickwinkel bei, indem sie Beziehungen zwischen kreuzbildlichen Affinitäten abbaut, auf deren Basis eine bessere Aggregation temporaler Informationen erreicht werden kann. Wir untersuchen Beziehungen zwischen Affinitäten in zwei Aspekten: Einzelmaßstabs-intrinsische Korrelationen und Multimaßstabs-Beziehungen. Inspiriert von der traditionellen Merkmalsverarbeitung schlagen wir "Single-scale Affinity Refinement" (SAR) und "Multi-scale Affinity Aggregation" (MAA) vor. Schließlich werden die durch SAR und MAA gestärkten kreuzbildlichen Affinitäten zur adaptiven Aggregation temporaler Informationen verwendet. Unsere Experimente zeigen, dass die vorgeschlagene Methode State-of-the-Art VSS-Methoden deutlich übertrifft.

PUBLICATIONS

The following publications are included as a whole or in parts in this thesis:

1. Sun, G., Wang, W., Dai, J. & Van Gool, L. *Mining cross-image semantics for weakly supervised semantic segmentation* in *European Conference on Computer Vision* (2020), 347.
2. Sun, G., Liu, Y., Ding, H., Probst, T. & Van Gool, L. *Coarse-to-fine feature mining for video semantic segmentation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 3126.
3. Sun, G., Liu, Y., Tang, H., Chhatkuli, A., Zhang, L. & Van Gool, L. *Mining relations among cross-frame affinities for video semantic segmentation* in *European Conference on Computer Vision* (2022), 522.
4. Sun, G., An, Z., Liu, Y., Liu, C., Sakaridis, C., Fan, D.-P. & Van Gool, L. *Indiscernible Object Counting in Underwater Scenes* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 13791.
5. Wang, W., Sun, G. & Van Gool, L. Looking beyond single images for weakly supervised semantic segmentation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

Furthermore, the following publications for which I made contributions are part of my PhD research, but not included in this thesis. The topics of these publications are outside of the scope of the material covered here:

1. Sun, G., Liu, Y., Probst, T., Paudel, D. P., Popovic, N. & Van Gool, L. Boosting crowd counting with transformers. *arXiv preprint arXiv:2105.10926-Accepted to Machine Intelligence Research* (2023).
2. Sun, G., Probst, T., Paudel, D. P., Popović, N., Kanakis, M., Patel, J., Dai, D. & Van Gool, L. *Task switching network for multi-task learning* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 8291.
3. Sun, G., Khan, S., Li, W., Cholakkal, H., Khan, F. S. & Van Gool, L. *Fixing localization errors to improve image classification* in *European Conference on Computer Vision* (2020), 271.
4. Cholakkal, H., Sun, G., Khan, S., Khan, F. S., Shao, L. & Van Gool, L. Towards partial supervision for generic object counting in natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 1604 (2020).
5. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L. & Timofte, R. *Swinir: Image restoration using swin transformer* in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2021), 1833.
6. Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J. & Shao, L. *Camouflaged object detection* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 2777.

7. Liang, J., Sun, G., Zhang, K., Van Gool, L. & Timofte, R. *Mutual affine network for spatially variant kernel estimation in blind image super-resolution* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 4096.

ACKNOWLEDGEMENTS

During the last four years, it was my great pleasure to work with some of the most brilliant minds in the world. I was also grateful to receive lots of support from my friends and family. Without numerous help and care, this thesis would not be possible.

First, I would like to thank my supervisor Prof. Luc Van Gool for supervising my PhD study at the Computer Vision Lab (CVL), ETH Zurich. I learned a lot from our discussions, his suggestions, and the way he manages projects and labs. Under his leadership, the lab attracts the best researchers in computer vision/artificial intelligence/machine learning and provides the best resources (software, hardware, environment) to me. I can not think of a better place to pursue my PhD than CVL.

In my opinion, one of the best advantages to conduct a PhD study at CVL is the chance to work with and learn from the best young researchers (rising stars) in computer vision and artificial intelligence. I greatly thank the supervisions received from Prof. Dr. Radu Timofte, Dr. Thomas Probst, Dr. Danda Pani Paudel, Dr. Wenguan Wang, Dr. Dengxin Dai, Dr. Yun Liu, Dr. Deng-Ping Fan, and Dr. Christos Sakaridis. From discussions with them, I learned a lot on how to conduct high-quality research, manage various projects, and present research works. I would like to show special thanks to my close collaborator Dr. Yun Liu for his great ideas and insights in computer vision. I am always impressed by his ability to think deeply and completely when dealing with research problems.

I would like to thank my co-authors, Jingyun Liang, Dr. Hao Tang, Nikola Popovic, Ce Liu, Menelaos Kanakis, Dr. Henghui Ding, Dr. Wen Li, Dr. Zongwei Wu, and Zhaochong An. I want to emphasize their intelligence and diligence when preparing papers for conferences and journals. They are the best researchers to collaborate with.

I specially thank external collaborators, Prof. Fahad Shahbaz Khan, Prof. Hisham Cholakkal, Prof. Salman Khan, and Prof. Ling Shao from MBZUAI and Inception Institue of Artificial Intelligence (IIAI). They brought me into the amazing field of computer vision. I thank Dr. Brian Price, Dr. Kenji Enomoto, Dr. Joon-Young Lee, and TJ Rhodes for their help during my internship at Adobe Research.

My further thanks go to the people with whom I played ping-pong, explored Zurich city/Switzerland, swam in Zurich lake, participated in

SOLA, and talked about life/world. They are Dr. Yulun Zhang, Dr. Suryansh Kumar, Dr. Deng-Ping Fan, Dr. Yawei Li, Rui Gong, Dr. Qin Wang, Dr. Gurkirt Singh, Mengya Liu, Dr. Goutam Bhat, Dr. Ren Yang, Jan-Nico Zaech, and Shuo Wang. It is my great pleasure to spend joyful times with them.

I further thank all the CVL members and alumnus. I remember all the times when we met in the office, had coffee in the kitchen, attend lab activities, and so on. Thank you for your company, joy, and laughter.

Finally, I thank all my family members and loved ones who are always there to support me without conditions during this process. They always help me make important and right decisions. Without them, this thesis would be impossible.

CONTENTS

1 Introduction 1
  1.1 Overview 4
    1.1.1 Indiscernible Object Counting 4
    1.1.2 Weakly Supervised Semantic Segmentation . . 5
    1.1.3 Video Semantic Segmentation 6
2 IOC 9
  2.1 Introduction 9
  2.2 Related Works 13
    2.2.1 Generic Object Counting 13
    2.2.2 Dense Object Counting 13
    2.2.3 Indiscernible Object Counting 14
  2.3 The IOCfish5K Dataset 15
    2.3.1 Image Collection 15
    2.3.2 Image Annotation 15
    2.3.3 Dataset Details 16
  2.4 IOCFormer 17
    2.4.1 Network Structure 18
    2.4.2 Density-Enhanced Transformer Encoder 19
    2.4.3 Loss Function 21
  2.5 Experiments 23
    2.5.1 Experimental Setting 23
    2.5.2 Counting Results and Analysis 25
    2.5.3 Ablation Study 26
  2.6 Additional Visual Results 28
  2.7 Conclusion 29
3 MCIS 35
  3.1 Introduction 35
  3.2 Related Work 37
  3.3 Methodology 39
    3.3.1 Co-attention Classification Network 39
    3.3.2 Co-Attention Classifier Guided WSSS Learning 45
    3.3.3 Detailed Network Architecture 47
  3.4 Experiment 48
    3.4.1 Experiment 1: Learn WSSS only from PASCAL VOC Data 49

3.4.2 Experiment 2: Learn WSSS with Extra Simple Single-Label Data . . . 50
3.4.3 Experiment 3: Learn WSSS with Extra Web-Sourced Data . . . 50
3.4.4 Experiment 4: Performance on WSSS Track of LID$_{20}$ Challenge . . . 51
3.4.5 Ablation Studies . . . 52
3.4.6 Additional Visual Results . . . 54
3.5 Conclusion . . . 56
4 CFFM . . . 57
4.1 Introduction . . . 57
4.2 Related Work . . . 59
4.2.1 Image Semantic Segmentation . . . 59
4.2.2 Video Semantic Segmentation . . . 60
4.2.3 Transformer . . . 61
4.3 Methodology . . . 63
4.3.1 Technical Motivation . . . 63
4.3.2 Coarse-to-Fine Feature Assembling . . . 65
4.3.3 Cross-frame Feature Mining . . . 66
4.3.4 Complexity Analysis . . . 67
4.3.5 Difference with STT . . . 68
4.4 Experiments . . . 68
4.4.1 Experimental Setup . . . 68
4.4.2 Comparison with State-of-the-art Methods . . . 70
4.4.3 Ablation Study . . . 73
4.5 Conclusion . . . 75
5 MRCFA . . . 77
5.1 Introduction . . . 77
5.2 Related Works . . . 79
5.2.1 Image Semantic Segmentation . . . 79
5.2.2 Video Semantic Segmentation . . . 80
5.3 Methodology . . . 81
5.3.1 Preliminaries . . . 82
5.3.2 Single-scale Affinity Refinement . . . 82
5.3.3 Multi-scale Affinity Aggregation . . . 84
5.3.4 Selective Token Masking . . . 85
5.4 Experiments . . . 87
5.4.1 Experimental Setup . . . 87
5.4.2 Ablation Studies . . . 89

5.4.3 Segmentation Results 93
5.4.4 Additional Visual Results 98
5.5 Conclusion 100
6 Conclusion and Outlook 101
6.1 Contributions 101
6.2 Discussion and Future Research 103

Bibliography 107

1

INTRODUCTION

Visual perception, aiming to equip machines with the capacity to comprehend the environments through vision signals akin to humans, has always been a pivotal focus in computer vision and artificial intelligence. When it comes to comprehend images/videos, achieving dense visual perception requires algorithms to predict dense outputs (maps), including tasks such as object detection, object counting, and semantic/instance segmentation. The features are of vital importance to the dense prediction. Traditional methods mostly leveraged low-level features such as pixel density and gradients, to segment images into different parts. However, their efficacy was largely constrained by the limited representation capacity of features which hardly convey high-level semantic information. In recent years, we witness the power of deep neural networks in generating high-quality and representative features, containing both high-level semantics and fine-grained details. Consequently, deep-learning-based algorithms [1–4] have dominated the realm of dense vision prediction.

In the era of deep learning, a number of breakthroughs have emerged in computer vision. Modern network architectures [5–7], utilizing convolutional layers, were proposed to significantly improve classification accuracy on large-scale image datasets [8]. Benefiting from the strong representation ability of these networks and the availability of large-scalse datasets [9–11] on downstream tasks, great progresses have been made for dense prediction tasks. Those CNN-based methods achieve state-of-the-art performances on object counting [2, 12], detection [1, 13], semantic segmentation [3, 4], and instance segmentation [1], surpassing their traditional counterparts by large margins. Recently, motivated by the success of transformers in natural language processing, transformer-based approaches [14–17] exhibit stronger representation ability due to the usage of self-attention layers which could model global information, and further improve performances on dense prediction tasks.

Despite the promising advancements in the field, previous research on dense prediction tasks has the following weaknesses. First, the main research focus is on general/normal/common scenes, where excellent results are achieved. However, challenging and difficult cases are often neglected, for which existing methods do not perform well. Second, most existing

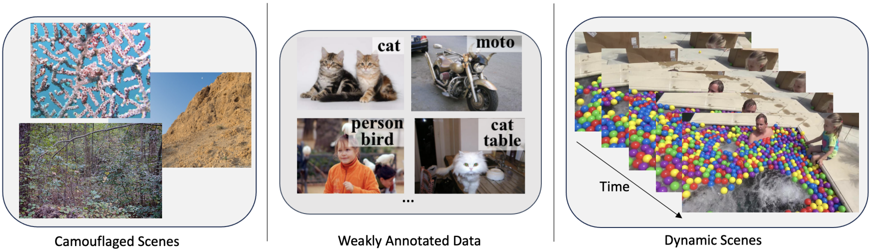


FIGURE 1.1: Illustrations of challenging scenarios for dense prediction. From *left* to *right*: camouflaged scenes where foreground objects are blended in the environments, weakly annotated data where only weak annotations are available for training, and dynamic and complex scenes where video frames are dependent on each other and temporal dependency exists.

methods are data-hungry and require densely-annotated data for training. However, obtaining annotations for dense prediction tasks consumes lots of human labors and time. The requirement for dense annotations is also counter-intuitive since humans are never supervised by these annotations and still perform dense prediction tasks extremely well. Third, existing research focuses more on image understanding and pays much less attention on video understanding. However, real scenes are dynamic and video is a more realistic data modality. Therefore, more research efforts are required for video dense prediction. Motivated by these weaknesses, this thesis addresses the following problems: **(1)** dense prediction under camouflaged scenes where foreground objects are seamless blended in the environments and difficult to distinguish using existing approaches; **(2)** dense prediction under weak annotations where only weak supervisions are available for training; **(3)** dense prediction for video understanding where dynamic and complex scenes are the perception target. Since these problems represent the visual perception tasks under challenging scenarios, we refer them as three "challenging dense prediction" tasks, as shown in Fig. 1.1.

Concurrently, attention mechanisms are widely used in various domains and show excellent performances. It stems from the observation that humans and animals naturally possess an ability to pay more attention to important regions/objects - an ability crucial in dealing with complex environments. In normal dense prediction, attention mechanisms are used to make neural networks focus on important regions or locations in the feature maps, images, or videos while disregarding irrelevant parts. As

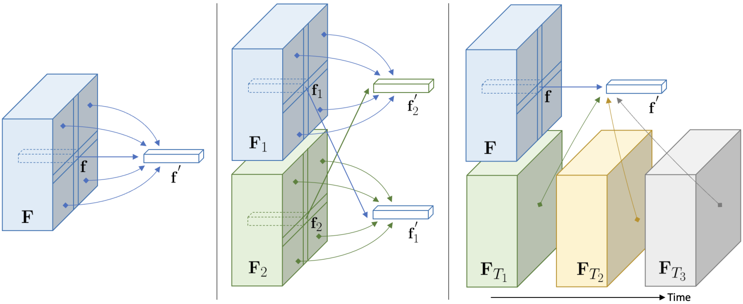


FIGURE 1.2: Illustrations of attention mechanisms. *Left*: singe-image attention where features from the same image are used to enhance the per-pixel feature: $(\mathbf{f}, \mathbf{F}) \rightarrow \mathbf{f}^{'}$. *Middle*: cross-image attention where features from another image are used to enhance the per-pixel feature: $(\mathbf{f}_1, \mathbf{F}_2) \rightarrow \mathbf{f}_1^{'}$ or $(\mathbf{f}_2, \mathbf{F}_1) \rightarrow \mathbf{f}_2^{'}$. *Right*: video attention where features from video frames are used to enhance the per-pixel feature: $(\mathbf{f}, \mathbf{F}_{T_1}, \mathbf{F}_{T_2}, \mathbf{F}_{T_3}) \rightarrow \mathbf{f}^{'}$. $\mathbf{F}$, and $\mathbf{F}_*$ represent feature maps for images or video frames.

a result, various attention mechanisms are developed, including channel attention [18], spatial attention [19], and temporal attention [20]. Inspired by the efficacy of attention mechanisms and the fact that humans usually utilize the natural ability of attention to tackle challenging perception tasks, this thesis mainly focuses on exploiting attention mechanisms for the aforementioned challenging dense prediction tasks.

In this thesis, the objective of developing advanced attention mechanisms for challenging dense prediction tasks is to enhance per-pixel feature representation. Consider a single pixel of an image, its feature could be refined through attention using various sources of information: (1) other pixels' features within the same image; (2) other pixels' features in another related image; and (3) other pixels' features in multiple related images. Therefore, different from existing studies on attention techniques, we consider the attention from three different aspects to enhance the per-pixel feature in an image: (1) *single-image attention* which uses features within the same image, as shown in Fig. 1.2 (*left*); (2) *cross-image attention* which uses features from another related image, as depicted in Fig. 1.2 (*middle*); (3) *video attention* which employs features from multiple video frames, as illustrated in Fig. 1.2 (*right*). For each challenging task, we study one attention mechanism, to demonstrate the effectiveness of enhancing per-pixel feature. To be more specific, for the first challenging task, we leverage single-image attention by merging global and location contextual information within an image. For the second challenging task, we pair two images as an input and then exploit cross-image attention between two images. Lastly, for the third challenging task, we deal with multiple video frames and exploit video attention among frames.

To sum up, the aim of this thesis is to investigate the challenging dense prediction tasks and demonstrate the effectiveness of enhancing per-pixel feature representation through developing advanced attention mechanisms: single-image attention, cross-image attention, and video attention. These dense prediction tasks have a wide range of applications and are of great significance to both academia and industry. Our contributions include the development of *four* robust algorithms for dense prediction tasks and explorations of designing advanced attention mechanisms to achieve this purpose.

In the following, we will introduce the structure of this thesis.

1.1 OVERVIEW

The content of this thesis is organized into three parts: 1) indiscernible object counting for which the focus is on camouflaged/indiscernible scene understanding and single-image attention is explored; 2) weakly supervised semantic segmentation for which the focus is on learning from weak supervisions/annotations and cross-image attention is explored; 3) video semantic segmentation for which the focus is on understanding the dynamic and complex scenes and video attention is explored. The three parts correspond to the three challenging dense prediction tasks, as mentioned above.

1.1.1 *Indiscernible Object Counting*

The first part of this thesis delves into dense prediction on camouflaged or indiscernible scenes, where foreground objects are seamlessly blended in the background due to similar colors or textures. This phenomenon is prevalent in natural environments and real-life situations. For instance, wild animals evolve to have colors similar to their surroundings, and during rainy or heavy foggy days, objects on the street blend into the background due to unfavorable lighting conditions. Recently, due to its wide applications, camouflaged scene understanding [21–24] has attracted more and more attention in computer vision community. Large datasets and successful networks have been proposed for camouflaged object detection and instance segmentation.

However, no existing work focuses on camouflaged object counting, the aim of which is to count the number of foreground objects in camouflaged environments. In Chapter 2, we introduce a new task named indiscernible object counting. Since there do not exist large-scale datasets for this problem, we present a large dataset IOCfish5K with dense and accurate annotations, i.e., points located in the center of objects. For benchmarking purposes, we evaluate existing common object counting methods on our dataset, which shows that they do not perform well on camouflaged/indiscernible scenes. Furthermore, a novel framework IOCFormer, exploiting both local and global attention to strengthen each per-pixel feature, is proposed to deal with the object counting under the challenging indiscernible scenes. This method exploits single-image attention since only information within the same image is used to refine the per-pixel feature of a image. The enhanced features from the attention mechanism can better distinguish the indiscernible objects from the background. Consequently, the proposed approach achieves state-of-the-art performance on this dataset with indiscernible scenes.

1.1.2 *Weakly Supervised Semantic Segmentation*

The second part of this thesis explores dense prediction under weak supervisions. For dense prediction tasks, acquiring dense annotations is not only labor-intensive, but also time-consuming. What's more, humans can perform dense prediction tasks by learning from a few instructions, without the need for expensive dense supervisions. Therefore, it is crucial to develop dense prediction models which could get rid of dense supervisions and learn from weak supervisions.

For semantic segmentation, there are different forms of weak annotations such as scribbles [25], bounding boxes [26, 27], points [28], and image-level labels [29–31]. Among them, image-level labels are the easiest to obtain, while using them to train segmentation models is the most challenging. In Chapter 3, we propose a novel method, MCIS, which effectively exploits the data with image-level supervisions by mining the relations between paired images through advanced co-attention mechanisms. Specifically, we propose to pair two images as an input, and MCIS then mines the common and unshared semantics within two images through co-attention and contrastive co-attention modules, respectively. This method leverages the value of cross-image attention as it uses information from another image to enhance the per-pixel feature of an image. The proposed method achieves state-of-the-art performances under different weakly-supervised settings. Our approach also won the first prize in Weakly Supervised Semantic Segmentation Track of CVPR2020 Learning from Imperfect Data (LID) Workshop.

1.1.3 *Video Semantic Segmentation*

While previous parts focus on static scenes, the third part of this thesis studies dense prediction for dynamic scenes. We focus on semantic segmentation on videos, a fundamental dense prediction task. The goal of video semantic segmentation (VSS) is to predict a pre-defined category for each pixel in each frame of a video. Compared to image semantic segmentation, VSS is much less explored mainly because of the lack of large-scale datasets. Annotating all pixels of all video frames is extremely time-consuming and laborious. However, recent efforts have proposed large-scale datasets for VSS with high quality to facilitate the research for this task. Studying and improving this realistic and fundamental task have become urgent.

In Chapter 4, we propose a Coarse-to-Fine Feature Mining network (CFFM) for video semantic segmentation by exploiting cross-frame attentions. Contextual information is key to semantic segmentation [3, 4, 32–45]. For our task, there exist two kinds of contexts: static and motional contexts. The former refers to the static content within consecutive video frames while the latter represents the moving content. Static contexts are well studied in image semantic segmentation [3, 4, 32, 39, 40, 42–44] while the moving contexts are studied in existing VSS methods [46–58]. However, there is no works on learning static and motional contexts in a unified framework. To mitigate this gap, CFFM is proposed and has the ability to jointly learn static and motional information. To be more specific, contextual information on previous frames is mined in a coarse-to-fine manner depending on the distance of the frame with respect to the target frame. After that, contexts are exploited to refine the features for the target frame through non-self attention and help produce better segmentation. This method exploits video attention due to the use of information from previous frames to enhance the per-pixel of the current frame. Experiments show that CFFM achieves promising performance in terms of segmentation accuracy as well as temporal consistency among predictions.

In Chapter 5, we address video semantic segmentation from a different perspective, i.e., mining hyper relations among cross-frame attentions.

CFFM and previous VSS methods [46, 47, 50, 51, 54, 56, 58] exploit the contexts by computing affinities between the target and the contexts, which can be done through attention [59, 60] and optical flow [61]. After that, the affinities are used to refine the features for the target. There are two disadvantages in this process. First, affinities are directly used without further processing. We argue that there exists local information in a affinity map, which could be used to refine itself. For example, a location in the affinity map is correlated to its surrounding locations, similar to the assumption of convolution layers. Second, when computing affinities, previous methods usually use single-scale features while multi-scale affinities contain more information and should be used instead. Therefore, we propose a new method MRCFA, namely, Mining Relations among Cross-Frame Affinities. Similar to CFFM, this approach also exploits video attention. MRCFA further boosts performance on VSS benchmarks. What's more, due to the design of efficient module, our approach shows better trade-off between segmentation performance and efficiency.

2

INDISCERNIBLE OBJECT COUNTING IN UNDERWATER SCENES

2.1 INTRODUCTION

Object counting – to estimate the number of object instances in an image – has always been an essential topic in computer vision. Understanding the counts of each category in a scene can be of vital importance for an intelligent agent to navigate in its environment. The task can be the end goal or can be an auxiliary step. As to the latter, counting objects has been proven to help instance segmentation [2], action localization [62], and pedestrian detection [63]. As to the former, it is a core algorithm in surveillance [64], crowd monitoring [65], wildlife conservation [66], diet patterns understanding [67] and cell population analysis [68].

Previous object counting research mainly followed two directions: generic or common object counting (GOC) [2, 69–71] and dense object counting (DOC) [72–78]. The difference between these two sub-tasks lies in the studied scenes, as shown in Fig. 2.1. GOC tackles the problem of counting object(s) of various categories in natural/common scenes [69], i. e., images from PASCAL VOC [9] and COCO [10]. The number of objects to be estimated is usually small, i. e., less than 10. DOC, on the other hand, mainly counts objects of a foreground class in crowded scenes. The estimated count can be hundreds or even tens of thousands. The counted objects are often persons (crowd counting) [74, 79], vehicles [76, 80] or plants [77]. Thanks to large-scale datasets [9, 72, 73, 81–83] and deep convolutional neural networks (CNNs) trained on them, significant progress has been made both for GOC and DOC. However, to the best of our knowledge, there is no previous work on counting indiscernible objects.

Under indiscernible scenes, foreground objects have a similar appearance, color, or texture to the background and are thus difficult to be detected with a traditional visual system. The phenomenon exists in both natural and artificial scenes [21, 22]. Hence, scene understanding for indiscernible scenes has attracted increasing attention since the appearance of some pioneering works [21, 84]. Various tasks have been proposed and formalized: camouflaged object detection (COD) [21], camouflaged instance segmentation (CIS) [22] and video camouflaged object detection (VCOD) [23, 24]. How-

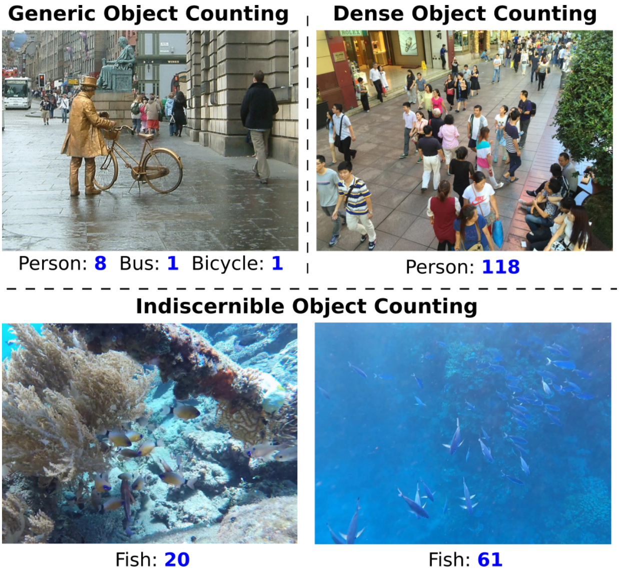


Figure 2.1: Illustration of different counting tasks. *Top left*: Generic Object Counting (GOC), which counts objects of various classes in *natural scenes*. *Top right*: Dense Object Counting (DOC), which counts objects of a foreground class in *scenes packed with instances*. *Down*: Indiscernible Object Counting (IOC), which counts objects of a foreground class in *indiscernible scenes*. Can you find all fishes in the given examples? For GOC, DOC, and IOC, the images shown are from PASCAL VOC [9], ShanghaiTech [72], and the new IOCfish5K dataset, respectively.

ever, no previous research has focused on counting objects in indiscernible scenes, which is an important aspect.

In this chapter, we study the new *indiscernible object counting* (**IOC**) task, which focuses on counting foreground objects in indiscernible scenes. Fig. 2.1 illustrates this challenge. Tasks such as image classification [5, 85], semantic segmentation [3, 86] and instance segmentation [1, 87] all owe their progress to the availability of large-scale datasets [8–10]. Similarly, a high-quality dataset for IOC would facilitate its advancement. Although existing datasets [21, 22, 88] with instance-level annotations can be used for IOC, they have the following limitations: 1) the total number of annotated objects in these datasets is limited, and image resolutions are low; 2) they only contain scenes/images with a small instance count; 3) the instance-level mask annotations can be converted to point supervision by computing the centers of mass, but the computed points do not necessarily fall inside the objects.

To facilitate the research on IOC, we construct a large-scale dataset, IOCfish5K. We collect 5,637 images with indiscernible scenes and annotate them with 659,024 center points. Compared with the existing datasets, the proposed IOCfish5K has several advantages: 1) it is the largest-scale dataset for IOC in terms of the number of images, image resolution, and total object count; 2) the images in IOCfish5K are carefully selected and contain diverse indiscernible scenes; 3) the point annotations are accurate and located at the center of each object. Our dataset is compared with existing DOC and IOC datasets in Table 2.1, and example images are shown in Fig. 2.2.

Based on the proposed IOCfish5K dataset, we provide a systematic study on 14 mainstream baselines [70, 72, 74, 79, 89–96]. We find that methods which perform well on existing DOC datasets do not necessarily preserve their competitiveness on our challenging dataset. Hence, we propose a simple and effective approach named IOCFormer. Specifically, we combine the advantages of density-based [93] and regression-based [79] counting approaches. The former can estimate the object density across the image, while the latter directly regresses the coordinates of points, which is straightforward and elegant. IOCFormer contains two branches: density and regression. The density-aware features from the density branch help make indiscernible objects stand out through the proposed density-enhanced transformer encoder (DETE). Then the refined features are passed through a conventional transformer decoder, after which predicted object points are generated. Experiments show that IOCFormer outperforms all considered algorithms, demonstrating its effectiveness on IOC. To summarize, our contributions are three-fold.

- We propose the new indiscernible object counting (IOC) task. To facilitate research on IOC, we contribute a large-scale dataset IOCfish5K, containing 5,637 images and 659,024 accurate point labels.
- We select 14 classical and high-performing approaches for object counting and evaluate them on the proposed IOCfish5K for benchmarking purposes.
- We propose a novel baseline, namely IOCFormer, which integrates density-based and regression-based methods in a unified framework. In addition, a novel density-based transformer encoder is proposed to gradually exploit density information from the density branch to help detect indiscernible objects.

Dataset	Year	Indiscernible Scene	#Ann. IMG	Avg. Resolution	Free View	Count Statistics				Web
						Total	Min	Ave	Max	
UCSD [65]	2008	✗	2,000	158×238	✗	49,885	11	25	46	Link
Mall [83]	2012	✗	2,000	480×640	✗	62,325	13	31	53	Link
UCF_CC_50 [97]	2013	✗	50	2101×2888	✓	63,974	94	1,279	4,543	Link
WorldExpo'10 [98]	2016	✗	3,980	576×720	✗	199,923	1	50	253	Link
ShanghaiTech B [72]	2016	✗	716	768×1024	✗	88,488	9	123	578	Link
ShanghaiTech A [72]	2016	✗	482	589×868	✓	241,677	33	501	3,139	Link
UCF-QNRF [73]	2018	✗	1,535	2013×2902	✓	1,251,642	49	815	12,865	Link
Crowd_surv [99]	2019	✗	13,945	840×1342	✗	386,513	2	35	1420	Link
GCC (synthetic) [100]	2019	✗	15,212	1080×1920	✗	7,625,843	0	501	3,995	Link
JHU-CROWD++ [82]	2019	✗	4,372	910×1430	✓	1,515,005	0	346	25,791	Link
NWPU-Crowd [81]	2020	✗	5,109	2191×3209	✓	2,133,375	0	418	20,033	Link
NC4K [88]	2021	✓	4,121	530×709	✓	4,584	1	1	8	Link
CAMO++ [22]	2021	✓	5,500	N/A	✓	32,756	N/A	6	N/A	Link
COD [101]	2022	✓	5,066	737×964	✓	5,899	1	1	8	Link
IOCfish5K (Ours)	2023	✓	5,637	1080×1920	✓	659,024	0	117	2,371	Link

Table 2.1: Comparisons of statistics of existing datasets for dense object counting (DOC) and indiscernible object counting (IOC). Please refer to the links for individual datasets.

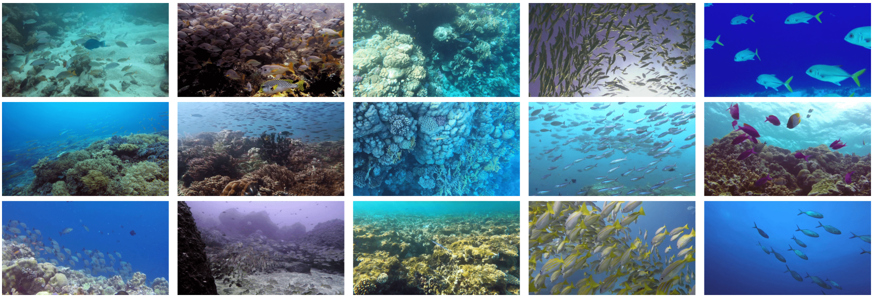


FIGURE 2.2: Example images from the proposed IOCfish5K. From *left* column to *right* column: typical samples, indiscernible & dense samples, indiscernible & less dense samples, less indiscernible & dense samples, less indiscernible & less dense samples.

2.2 RELATED WORKS

2.2.1 *Generic Object Counting*

Generic/common object counting (GOC) [2], also referred to as everyday object counting [69], is to count the number of object instances for various categories in natural scenes. The popular benchmarks for GOC are PASCAL VOC [9] and COCO [10]. The task was first proposed and studied in the pioneering work [69], which divided images into non-overlapping patches and predicted their counts by subitizing. LC [2] used image-level count supervision to generate a density map for each class, improving counting performance and instance segmentation. RLC [12] further reduced the supervision by only requiring the count information for a subset of training classes rather than all classes. Differently, LCFCN [70] exploited point-level supervision and output a single blob per object instance.

2.2.2 *Dense Object Counting*

Dense Object Counting (DOC) [72, 73, 75–77, 102, 103] counts the number of objects in dense scenarios. DOC contains tasks such as crowd counting [72, 73, 75, 81, 104, 105], vehicle counting [76, 80], plant counting [77], cell counting [68] and penguin counting [106]. Among them, crowd counting, i. e., counting people, attracts the most attention. The popular benchmarks for crowd counting include ShanghaiTech [72], UCF-QNRF [73], JHU-CROWD++ [75], NWPU-Crowd [81] and Mall [83]. For vehicle counting, researchers mainly use TRANCOS [76], PUCPR+ [80], and CAPRK [80]. For DOC on other categories, the available datasets are MTC [77] for counting plants, CBC [68] for counting cells, and Penguins [106] for counting penguins. DOC differs from GOC because DOC has far more objects to be counted and mainly focuses on one particular class.

Previous DOC works can be divided into three groups based on the counting strategy: detection [107], regression [65, 79, 94, 97], and density map generation [14, 74, 89, 93, 96]. Counting-by-detection methods first detect the objects and then count. Though intuitive, they are inferior in performance since detection performs unfavorably on crowded scenes. Counting-by-regression methods either regress the global features to the overall image count [65, 97] or directly regress the local features to the point coordinates [79, 94]. Most previous efforts focus on learning a density map, which is a single-channel output with reduced spatial size. It represents the fractional number of objects at each location, and its spatial integration equals the total count of the objects in the image. The density map can be learned by using a pseudo density map generated with Gaussian kernels [74, 90] or directly using a ground-truth point map [14, 91, 93].

For architectural choices, the past efforts on DOC can also be divided into CNN-based [70, 74, 89, 94, 108] and Transformer-based methods [14, 79, 109]. By nature, convolutional neural networks (CNNs) have limited receptive fields and only use local information. By contrast, Transformers can establish long-range/global relationships between the features. The advantage of transformers for DOC is demonstrated by [14, 109, 110].

2.2.3 *Indiscernible Object Counting*

Recently, indiscernible scene understanding has become popular [22, 23, 84, 101, 111]. It contains a set of tasks specifically focusing on detection, instance segmentation and video object detection/segmentation. It aims to analyze scenes with objects that are difficult to recognize visually [21, 23].

In this chapter, we study the new task of indiscernible object counting (IOC), which lies at the intersection of dense object counting (DOC) and indiscernible scene understanding. Recently proposed datasets [22, 88, 101] for concealed scene understanding can be used as benchmarks for IOC by converting instance-level masks to points. However, they have several limitations, as discussed in §2.1. Therefore, we propose the first large-scale dataset for IOC, IOCfish5K.

2.3 THE IOCFISH5K DATASET

2.3.1 *Image Collection*

Underwater scenes contain many indiscernible objects (*Sea Horse*, *Reef Stonefish*, *Lionfish*, and *Leafy Sea Dragon*) because of limited visibility and active mimicry. Hence, we focus on collecting images of underwater scenes.

We started by collecting Youtube videos of underwater scenes, using general keywords (*underwater scene*, *sea diving*, *deep sea scene*, etc..) and category-specific ones (*Cuttlefish*, *Mimic Octopus*, *Anglerfish*, *Stonefish*, etc..). In total, we collected 135 high-quality videos with lengths from tens of seconds to several hours. Next, we kept one image in every 100 frames (3.3 sec) to avoid duplicates. This still leaded to a large number of images, some showing similar scenes or having low quality. Hence, at the final step of image collection, 6 professional annotators carefully reviewed the dataset and removed those unsatisfactory images. The final dataset has 5,637 images, some of which are shown in Fig. 2.2. This step cost a total of 200 human hours.

2.3.2 *Image Annotation*

Annotation principles. The goal was to annotate each animal with a point at the center of its visible part. We have striven for *accuracy* and *completeness*. The former indicates that the annotation point should be placed at the object center, and each point corresponds to exactly one object instance. The latter means that no objects should be left without annotation.

Annotation tools. To ease annotation, we developed a tool based on open-source Labelimg[1]. It offers the following functions: generate a point annotation in an image by clicking, drag/delete the point, mark the point when encountering difficult cases, and zoom in/out. These functions help annotators to produce high-quality point annotations and to resolve ambiguities by discussing the marked cases.

Annotation process. The whole process is split into *three* steps. First, all annotators (6 experts) were trained to familiarize themselves with their tasks. They were instructed about sea animals and well-annotated samples. Then each of them was asked to annotate 50 images. The annotations were checked and evaluated. When an annotator passed the evaluation, he/she could move to the next step. Second, images were distributed to 6

---

[1] `https://github.com/heartexlabs/labelImg`

Datasets	# IMG (0-50)	# IMG (51-100)	# IMG (101-200)	# IMG (>200)	Total
NC4K [88]	4,121	0	0	0	4,121
COD [101]	5,066	0	0	0	5,066
IOCfish5K	2,663	1,000	957	1,017	5,637

Table 2.2: Comparison of datasets *w.r.t.* image distribution across various density (count) ranges. We compute the number of images for each dataset under four density ranges.

annotators, giving each annotator responsibility over part of the dataset. The annotators were required to discuss confusing cases and reach a consensus. Last, they checked and refined the annotations in two rounds. The second step cost 600 human hours, while each checking round in the third step cost 300 hours. The total cost of annotation process amounted to 1,200 human hours.

2.3.3 *Dataset Details*

The proposed IOCfish5K dataset contains 5,637 high-quality images, annotated with 659,024 points. Table 2.2 shows the number of images within each count range (0-50, 51-100, 101-200, and above 200). Of all images in IOCfish5K, 957 have a medium to high object density, i. e., between 101 and 200 instances. Furthermore, 1,017 images (18% of the dataset) show very dense scenes (>200 objects per image).

To standardize the benchmarking on IOCfish5K, we randomly divide it into three non-overlapping parts: train (3,137), validation (500), and test (2,000). For each split, the distribution of images across different count ranges follows a similar distribution, as shown in Fig. 2.3. This is due to the random sampling. We can also observe that there are plenty of images that have more than 50 object instances, which makes our dataset also valuable for density object counting.

Table 2.1 compares the statistics of IOCfish5K with previous datasets. The advantages of IOCfish5K over existing datasets are four-fold. **(1)** IOCfish5K is the largest-scale object counting dataset for indiscernible scenes. It is superior to its counterparts such as NC4K [88], CAMO++ [22], and COD [101] in terms of size, image resolution and the number of annotated points. For example, the largest existing IOC dataset CAMO++ [22] con-

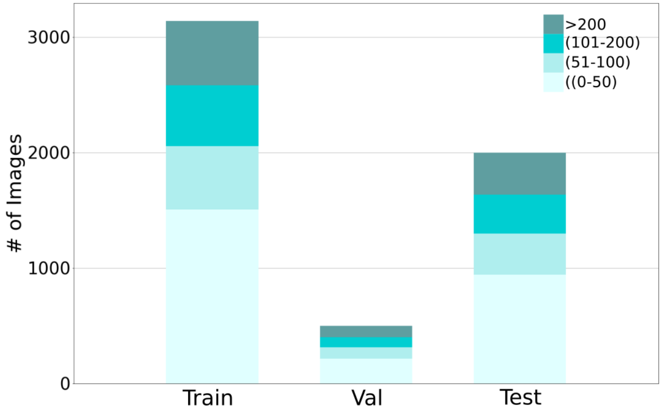


FIGURE 2.3: Image distributions under different density (count) ranges (<50, 51 to 100, 101 to 200, and >200) in training, validation (val), and test sets of IOCfish5K.

tains a total of 32,756 objects, compared to 659,024 points in IOCfish5K. **(2)** IOCfish5K has far denser images, which makes it currently the most challenging benchmark for IOC. As shown in Table 2.2, 1,974 images have more than 100 objects. **(3)** Although IOCfish5K is specifically proposed for IOC, it has some advantages over the existing DOC datasets. For instance, compared with JHU-CROWD++ [75], which is one of the largest-scale DOC benchmarks, the proposed dataset contains more images with a higher resolution. **(4)** IOCfish5K focuses on underwater scenes with sea animal annotations, which makes it different from all existing datasets shown in Table 2.1. Hence, the proposed dataset is also valuable for *transfer learning* and *domain adaptation* of DOC [112–115].

2.4 IOCFORMER

We first introduce the network structure of our proposed IOCFormer model, which consists of a density and a regression branch. Then, the novel density-enhanced transformer encoder, which is designed to help the network better recognize and detect indiscernible objects, is explained.

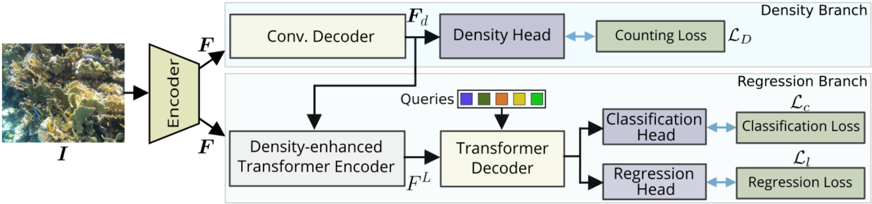


FIGURE 2.4: Overview of the proposed IOCFormer. Given an input image, we extract a feature map using an encoder, which is processed by a density branch and regression branch. The density-enhanced transformer encoder exploits the object density information from the density branch to generate more relevant features for the regression. Refer to §2.4 for more details.

2.4.1 *Network Structure*

As mentioned, mainstream methods for object counting fall into two groups: counting-by-density [74, 93] or counting-by-regression [79, 94]. The density-based approaches [74, 93] learn a map with the estimated object density across the image. Differently, the regression-based methods [79, 94] directly regress to coordinates of object center points, which is straightforward and elegant. As for IOC, foreground objects are difficult to distinguish from the background due to their similar appearance, mainly in color and texture. The ability of density-based approaches to estimate the object density level could be exploited to make (indiscernible) foreground objects stand out and improve the performance of regression-based methods. In other words, the advantages of density-based and regression-based approaches could be combined. Thus, we propose IOCFormer, which contains two branches: a density branch and a regression branch, as in Fig. 2.4. The density branch's information helps refine the regression branch's features.

Formally, we are given an input image $\boldsymbol{I}$ with ground-truth object points $\{(x_i, y_i)\}_{i=1}^{K}$ where (x_i, y_i) denotes the coordinates of the i-th object point and K is the total number of objects. The goal is to train an object counting model which predicts the number of objects in the image. We first extract a feature map $\boldsymbol{F} \in \mathbb{R}^{h \times w \times c_1}$ (h, w, and c_1 denote height, weight, and the number of channels, respectively) by sending the image through an encoder. Next, $\boldsymbol{F}$ is processed by the density and the regression branches.

The density branch inputs $\boldsymbol{F}$ into a convolutional decoder which consists of two convolutions with 3×3 kernels. A density-aware feature map $\boldsymbol{F}_d \in \mathbb{R}^{h \times w \times c_2}$ is obtained, where c_2 is the number of channels. Then a density head (a convolution layer with 1×1 kernel and ReLU activation) maps $\boldsymbol{F}_d$ to a single-channel density map $\boldsymbol{D} \in \mathbb{R}^{h \times w}$ with non-negative values. Similar to [93], the counting loss (L_1 loss) used in the density branch is defined as:

$$\mathcal{L}_D = \big|\|\boldsymbol{D}\|_1 - K\big|, \tag{2.1}$$

where $\|\cdot\|_1$ denotes the entry-wise L_1 norm of a matrix. The density map $\boldsymbol{D}$ estimates the object density level across the spatial dimensions. Hence, the feature map $\boldsymbol{F}_d$ before the density head is density-aware and contains object density information, which could be exploited to strengthen the feature regions with indiscernible object instances.

As to the regression branch, the feature map $\boldsymbol{F}$ from the encoder and the density-aware feature map $\boldsymbol{F}_d$ from the density branch are first fed into our density-enhanced transformer encoder, described in detail in §2.4.2. After this module, the refined features, together with object queries, are passed to a typical transformer decoder [116]. The decoded query embeddings are then used by the classification head and regression head to generate predictions. The details are explained in §2.4.3.

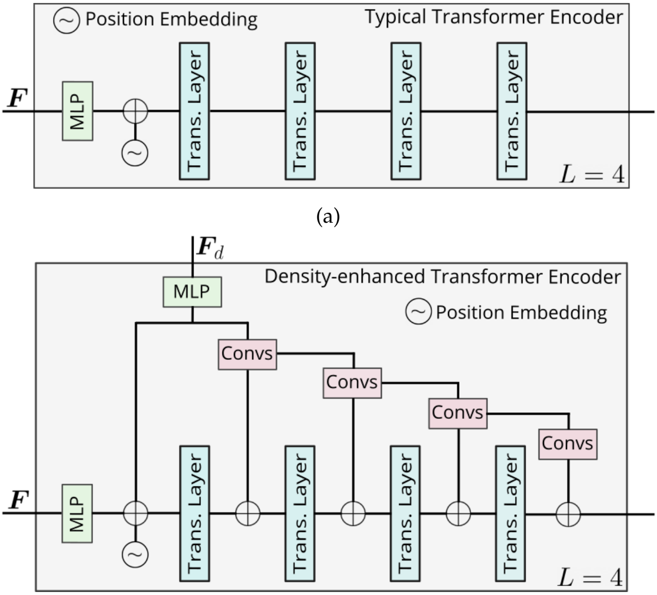
2.4.2 *Density-Enhanced Transformer Encoder*

Here, we explain the density-enhanced transformer encoder (DETE) in detail. The structure of the typical transformer encoder (TTE) and the proposed DETE is shown in Fig. 2.5. Different from TTE, which directly processes one input, DETE takes two inputs: the features ($\boldsymbol{F}$) extracted by the initial encoder and the density-aware features ($\boldsymbol{F}_d$) from the density branch. DETE uses the density-aware feature map to refine the encoder feature map. With information about which image areas have densely distributed objects and which have sparsely distributed objects, the regression branch can more accurately predict the positions of indiscernible object instances.

We first project $\boldsymbol{F}$ to $\hat{\boldsymbol{F}} \in \mathbb{R}^{h \times w \times c}$, and $\boldsymbol{F}_d$ to $\hat{\boldsymbol{F}}_d \in \mathbb{R}^{h \times w \times c}$ by using an MLP layer so that the number of channels (c) matches. The input to the first transformer layer is the combination of $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{F}}_d$ and position embedding $E \in \mathbb{R}^{hw \times c}$. This process is given by:

$$F^1 = \mathrm{Rs}(\hat{\boldsymbol{F}}) + \mathrm{Rs}(\hat{\boldsymbol{F}}_d) + E; \quad F^2 = \mathrm{Trans}(F^1), \tag{2.2}$$

where $\mathrm{Rs}(\cdot)$ denotes the operation of reshaping the feature map by flattening its spatial dimensions, and $\mathrm{Trans}(\cdot)$ denotes a transformer layer. After



(a)

(b)

FIGURE 2.5: Comparison between typical transformer encoder (a) and our density-enhanced transformer encoder (b) when $L = 4$.

that, additional transformer layers are used to further refine the features, as follows:

$$\begin{aligned}
&\boldsymbol{F}_d^1 = \hat{\boldsymbol{F}}_d, \\
&\boldsymbol{F}_d^i = \mathrm{Convs}(\boldsymbol{F}_d^{i-1}), \;\; i = 2, 3, ..., L-1, \\
&F^{i+1} = \mathrm{Trans}(F^i + \mathrm{Rs}(\boldsymbol{F}_d^i)), \;\; i = 2, 3, ..., L-1,
\end{aligned} \tag{2.3}$$

where $\mathrm{Convs}(\cdot)$ denotes a convolutional block containing two convolution layers. The total number of transformer layers is L which also represents the total times of merging transformer and convolution features. After Eq. (2.3), we obtain the density-refined features $F^L \in \mathbb{R}^{hw \times c}$ which are forwarded to the transformer decoder.

The benefit of our DETE can also be interpreted from the perspective of *global* and *local* information. Before each transformer layer in Eq. (2.3), we merge features from the previous transformer layer (global) and features from the convolutional block (local). During this process, the global and local information gradually get combined, which boosts the representation ability of the module.

2.4.3 *Loss Function*

After the DETE module, we obtain density-refined features F^L. Next, the transformer decoder takes the refined features F^L and trainable query embeddings $Q \in \mathbb{R}^{n \times c}$ containing n queries as inputs, and outputs decoded embeddings $\hat{Q} \in \mathbb{R}^{n \times c}$. The transformer decoder consists of several layers, each of which contains a self-attention module, a cross-attention layer and a feed-forward network (FFN). For more details, we refer to the seminal work [116]. $\hat{Q}$ contains n decoded representations, corresponding to n queries. Following [79], every query embedding is mapped to a confidence score by a classification head and a point coordinate by a regression head. Let $\{p_i, (\hat{x}_i, \hat{y}_i)\}_{i=1}^n$ denote the predictions for all queries, where p_i is the predicted confidence score determining the likelihood that the point belongs to the foreground and $(\hat{x}_i, \hat{y}_i)$ is the predicted coordinate for the i-th query. Then we conduct a Hungarian matching [79, 117] between predictions $\{p_i, (\hat{x}_i, \hat{y}_i)\}_{i=1}^n$ and ground-truth $\{(x_i, y_i)\}_{i=1}^K$. Note that n is bigger than K so that each ground-truth point has a matched prediction. The Hungarian matching is based on the k-nearest-neighbors matching objective [79]. Specifically, the matching cost depends on three parts: the distance between predicted points and ground-truth points, the confidence score of the predicted points, and the difference between predicted and

Method	Publication	Val (500)			Test (2,000)		
		MAE↓	MSE↓	NAE↓	MAE↓	MSE↓	NAE↓
MCNN [72]	CVPR'16	81.62	152.09	3.53	72.93	129.43	4.90
CSRNet [74]	CVPR'18	43.05	78.46	1.91	38.12	69.75	2.48
LCFCN [70]	ECCV'18	31.99	81.12	0.77	28.05	68.24	1.12
CAN [89]	CVPR'19	47.77	83.67	2.10	42.02	74.46	2.58
DSSI-Net [90]	ICCV'19	33.77	80.08	1.25	31.04	69.11	1.68
BL [91]	ICCV'19	19.67	44.21	0.39	20.03	46.08	0.55
NoisyCC [92]	NeurIPS'20	19.48	41.76	0.39	19.73	46.85	0.46
DM-Count [93]	NeurIPS'20	19.65	42.56	0.42	19.52	45.52	0.55
GL [118]	CVPR'21	18.13	44.57	0.33	18.80	46.19	0.47
P2PNet [94]	ICCV'21	21.38	45.12	0.39	20.74	47.90	0.48
KDMG [119]	TPAMI'22	22.79	47.32	0.90	22.79	49.94	1.17
MPS [95]	ICASSP'22	34.68	59.46	2.06	33.55	55.02	2.61
MAN [96]	CVPR'22	24.36	40.65	2.39	25.82	45.82	3.16
CLTR [79]	ECCV'22	17.47	37.06	0.29	18.07	41.90	0.43
IOCFormer (Ours)	CVPR'23	**15.91**	**34.08**	**0.26**	**17.12**	**41.25**	**0.38**

Table 2.3: Comparison with state-of-the-art methods on the validation and test set. The best results are highlighted in **bold**.

ground-truth average neighbor distance [79]. After the matching, we compute the classification loss $\mathcal{L}_c$, which boosts the confidence score of the matched predictions and suppresses the confidence score of the unmatched ones. To supervise the predicted coordinates' learning, we also compute the localization loss $\mathcal{L}_l$, which measures the L_1 distance between the matched predicted coordinates and the corresponding ground-truth coordinates. For more details, we refer to [79]. The final loss function is defined as:

$$\mathcal{L} = \lambda \mathcal{L}_D + \mathcal{L}_c + \mathcal{L}_l, \tag{2.4}$$

where λ is set to 0.5. The density and the regression branch are jointly trained using Eq. (2.4). During inference, we take the predictions from the regression branch.

2.5 EXPERIMENTS

2.5.1 *Experimental Setting*

Compared models. Since there is no algorithm specifically designed for IOC, we select 14 recent open-source DOC methods for benchmarking. The details of these methods are as follows.

- MCNN [72]: It proposes a multi-column convolutional neural network that contains different convolution branches with different receptive fields. The ground-truth density map is calculated using geometry-adaptive kernels.

- CSRNet [74]: It aims at conducting crowd counting under highly congested scenes. CSRNet exploits dilated convolutions in this task and achieve promising results.

- LCFCN [70]: This method predicts a blob for each object instance by using only point supervision. It achieves excellent performance in crowd counting as well as generic object counting.

- CAN [89]: CAN processes encoded features (VGG-16) with different receptive fields, which are then combined using the learned weights. The final context-aware features are passed to estimate the density map.

- DSSI-Net [90]: It focuses on tackling the problem of large-scale variation in crowd counting and proposes structured feature enhancement and dilated multi-scale structural similarity loss to generate better density maps.

- BL [91]: Different from previous works which adopt L_1 or L_2 loss for supervising the learning of density maps, BL proposes a Bayesian loss which directly uses point annotations to learn density probability.

- NoisyCC [92]: NoisyCC explicitly models the annotation noise in crowd counting with a joint Gaussian distribution. A low-rank covariance approximation is derived to improve the efficiency [92].

- DM-Count [93]: This method proposes to exploit distribution matching for crowd counting. The optimal transport algorithm is used to minimize the gap between the predicted density map and the ground-truth point map.

- GL [118]: GL proposes a perspective-guided optimal transport cost function for crowd counting. It is currently the most powerful loss for crowd counting and achieves state-of-the-art performance on mainstream DOC datasets compared to other loss functions.
- P2PNet [94]: It directly predicts a number of point proposals (location and confidence score). Then Hungarian algorithm [117] is used to match proposals and point annotations. It is a purely point-based algorithm for crowd counting [94] and achieves impressive performance on DOC datasets.
- KDMG [119]: Different from previous density-based methods, which generates ground-truth density map by convolving the point map with a/an (adaptive) Gaussian kernel, KDMG proposes a density map generator that is jointly trained with counting model.
- MPS [95]: This method generates multi-scale features for the crowd image and benefits from the joint learning of crowd counting as well as localization.
- MAN [96]: It deals with the problem of large-scale variations in crowd counting by integrating global attention, local attention, and instance attention in a unified framework. MAN achieves state-of-the-art performance on mainstream datasets such as JHU++ and NWPU.
- CLTR [79]: It directly predicts the point locations by adopting a transformer encoder and decoder structure to process the features. The trainable embeddings are used to extract object locations from the encoded features.

For the above methods, CAN, CSRNet and MCNN use the SGD optimization algorithm for training the network, while others use Adam optimizer [120]. For IOCFormer, the initial learning rate is set as 1e-5 and the weight decay is 5e-4. Following [79], our approach is trained by 1500 epochs. Also, P2PNet and CLTR are based on regression, while others are on density map estimation.

Implementation details. For methods such as MCNN and CAN, we use open-source re-implementations for our experiments. For the other methods, we use official codes and default parameters. All experiments are conducted on PyTorch [121] and NVIDIA GPUs. L in DETE is set to 6 and the number of queries (n) is set as 700. Following [79], our IOCFormer uses ResNet-50 [5] as encoder, pretrained on Imagenet [8]. Other modules/parameters are randomly initialized. For data augmentations, we use random resizing and horizontal flipping. The images are randomly cropped to 256 × 256 inputs. Each batch contains 8 images, and the Adam optimizer [120] is used. During inference, we split the images into patches of the same size as during training. Following [79], we use a threshold (0.35) to filter out background predictions.

Metrics. To evaluate the effectiveness of the baselines and the proposed method, we compute Mean Absolute Error (MAE), Mean Square Error (MSE), and Mean Normalized Absolute Error (NAE) between predicted counts and ground-truth counts for all images, following [79, 81, 93].

2.5.2 *Counting Results and Analysis*

We present the results of 14 mainstream crowd-counting algorithms and IOCFormer in Table 2.3. All methods follow the same evaluation protocol: the model is selected via the val set. Based on the results, we observe:

- Among all previous methods, the recent CLTR [79] outperforms the rest, with 18.07, 41.90, 0.43 on the test set for MAE, MSE, and NAE, respectively. The reason is that this method uses a transformer encoder to learn global information and a transformer decoder to directly predict center points for object instances.

- Some methods (MAN [96] and P2PNet [94]) perform competitively on DOC datasets such as JHU++ [82] and NWPU [81], but perform worse on IOCfish5K. For example, MAN achieves 53.4 and 209.9 for MAE and MSE on JHU++, outperforming other methods, including CLTR which achieves 59.5 and 240.6 for MAE and MSE. However, MAN underperforms on IOCfish5K, compared to CLTR, DM-Count, NoisyCC, and BL. This shows that methods designed for DOC do not necessarily work well for indiscernible objects. Hence, IOC requires specifically designed solutions.

- These methods, including BL, NoisyCC, DM-Count, and GL, which propose new loss functions for crowd counting, perform well despite being simple. For example, GL achieves 18.80, 46.19, and 0.47 for MAE, MSE, and NAE on the test set.

Different from previous methods, IOCFormer is specifically designed for IOC with two novelties: (1) combining density and regression branches in a

Datasets	MAE		MSE	
	IOCfish5K	JHU-CROWD++	IOCfish5K	JHU-CROWD++
CSRNet [74]	**38.12**	85.90	**69.75**	309.20
DSSI-Net [90]	31.04	**133.50**	69.11	**416.50**
BL [91]	20.03	75.00	46.08	299.90
NoisyCC [92]	19.73	67.70	46.85	258.50
MAN [96]	25.82	**53.40**	45.82	**209.90**
CLTR [79]	**18.07**	59.50	**41.90**	240.60

TABLE 2.4: Counting performance comparison between IOCfish5K and JHU-CROWD++ [75] for existing algorithms. The best result is shown in **red** while the inferior one is shown in **blue**. The results for JHU-CROWD++ are from relevant papers. It shows that the method which performs well on JHU-CROWD++ does not necessarily work favorably on IOCfish5K and vice versa.

unified framework, which improves the underlying features; (2) density-based transformer encoder, which strengthens the feature regions where objects exist. On both the val and test sets, IOCFormer is superior to all previous methods for MAE, MSE, and NAE. Besides the quantitative results, we also show qualitative results of some approaches in Fig. 2.6.

Cross-dataset analysis. We compare the performance of various existing methods on IOCfish5K and JHU-CROWD++ [75] in Table 2.4. We observe that the order of top-performing methods on IOCfish5K do not follow the same trend as the ranking on JHU-CROWD++ for both MAE and MSE, which validates that there is a domain gap between IOC and DOC. For example, CLTR [79] is the best method on our dataset while MAN [96] outperforms other approaches on JHU-CROWD++. Similarly, CSRNet [74] performs not as favorably as others on our dataset while DSSI-Net [90] takes that position on JHU-CROWD++.

2.5.3 *Ablation Study*

Impact of the density branch and DETE. As mentioned, the proposed model combines a density and a regression branch in a unified framework, aiming to combine their advantages. In Table 2.5, we show the results of separately training the density branch and the regression branch. We also provide results of jointly training the density branch and regression branch without using the proposed DETE. The comparison shows that

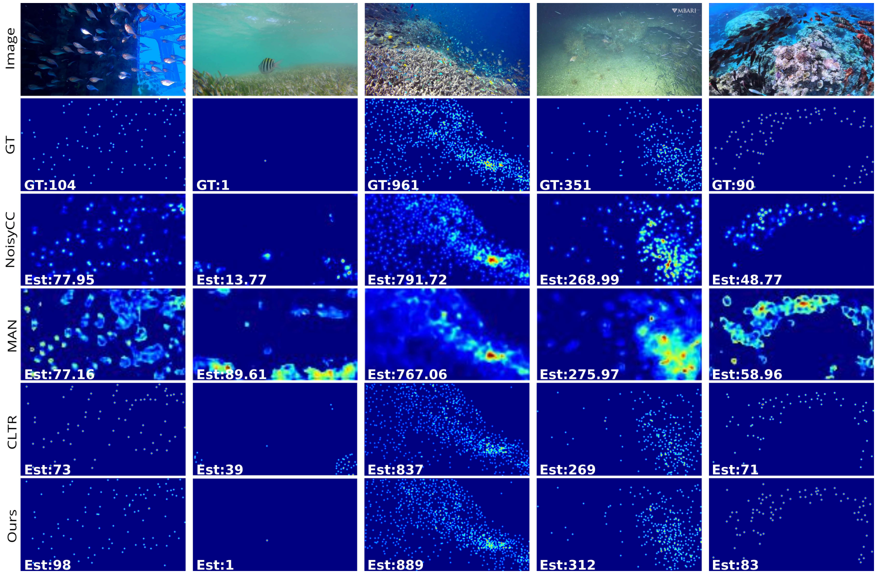


FIGURE 2.6: Qualitative comparisons of various algorithms (NoisyCC [92], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

Methods	DETE	MAE↓	MSE↓	NAE↓
DB	✗	18.25	39.77	0.29
Regression	✗	17.47	37.06	0.29
DB+Regression	✗	16.94	35.92	**0.26**
	✓	**15.91**	**34.08**	**0.26**

TABLE 2.5: Impact of density branch (DB) and DETE on IOCfish5K val set. For DB+Regression without using DETE, a typical transformer encoder (TTE) is used instead.

the regression branch, though straightforward, performs better than only using the density branch. Furthermore, training both branches together without DETE gives better performance than using only the regression branch. The improvement could be explained from the perspective of multi-task learning [122–124]. The added density branch, which could

L	MAE↓	MSE↓	NAE↓
2	16.75	35.87	0.28
4	16.59	35.23	0.26
6	15.91	34.08	0.26
8	**15.72**	**33.63**	**0.24**

Table 2.6: Impact of the number of transformer layers or convolutional blocks in DETE.

be regarded as an *additional task*, helps the encoder learn better features. By establishing connections between the density and regression branches, better performance is obtained. Compared to the variant without DETE, our final model has a clear superiority by reducing MAE from 16.94 to 15.91 and MSE from 35.92 to 34.08. The results validate the effectiveness of DETE for enhancing the features by exploiting the information generated from the density branch.

Impact of L. We change the number of Trans or Convs in DETE and report results in Table 2.6. By increasing L, we obtain better performance, showing the capability of our DETE to produce relevant features. We use $L = 6$ in our main setting to balance complexity and performance.

2.6 ADDITIONAL VISUAL RESULTS

In this section, we show more visual results, which qualitatively compare our method with more algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79]) in Fig. 2.7-2.11. We have the following observations. **(1)** On those samples, our method achieves the best MAE by predicting a more accurate overall count compared to all other approaches. **(2)** Unlike density-based methods (MCNN, BL, NoisyCC, DM-Count, and MAN), which only estimate the density level across the image, IOCFormer can also generate accurate locations (coordinates) for object instances. **(3)** Compared to regression-based methods (P2PNet and CLTR), our point predictions are visually better, demonstrating the effectiveness of the proposed method in localizing objects in camouflaged scenes. This is due to our designed module DETE, which exploits object density information from the density branch to help refine the features in the regression branch and make camouflaged objects stand out.

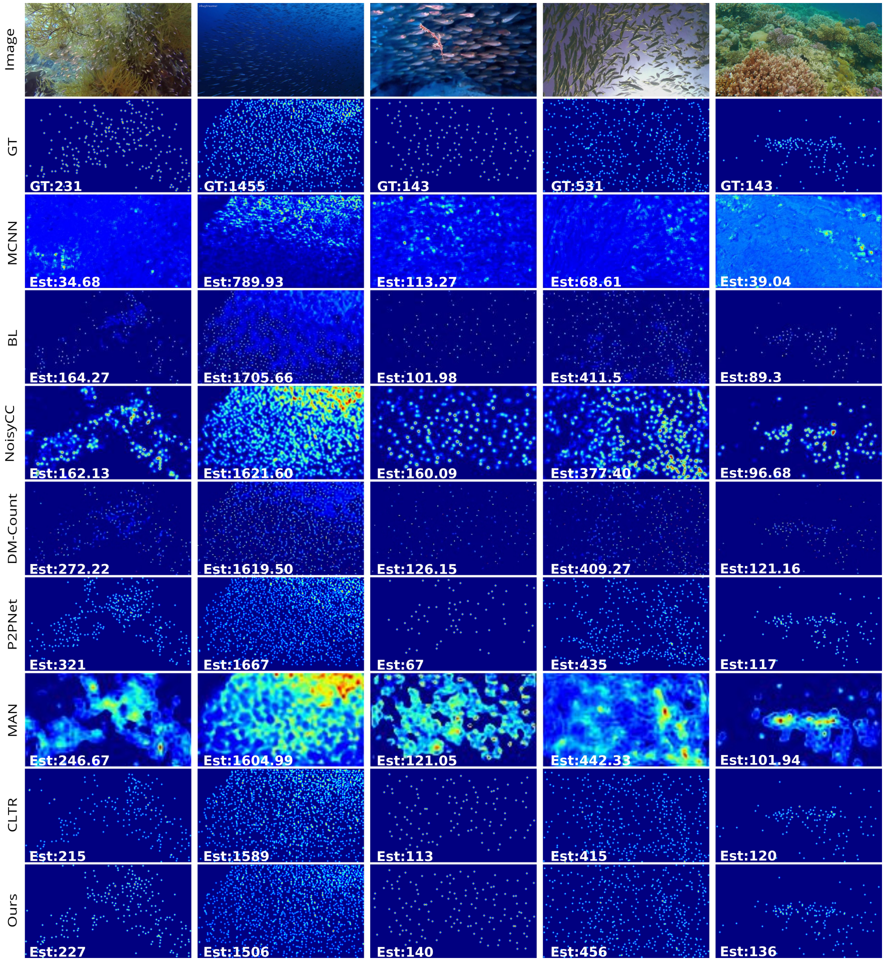


FIGURE 2.7: Qualitative comparisons of various algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

2.7 CONCLUSION

We provide a rigorous study of a new challenge named indiscernible object counting (IOC), which focuses on counting objects in indiscernible scenes. To address the lack of a large-scale dataset, we present the high-

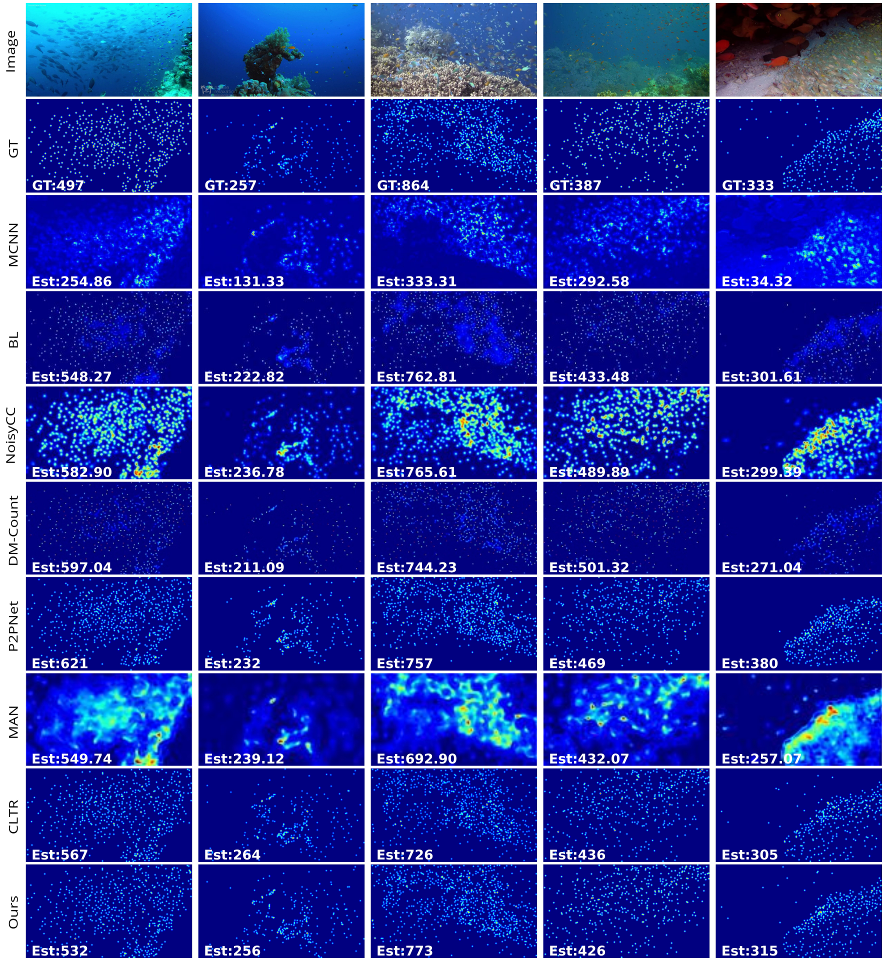


FIGURE 2.8: Qualitative comparisons of various algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

quality IOCfish5K which mainly contains underwater scenes and has point annotations located at the center of object (mainly fish) instances. A number of existing mainstream baselines are selected and evaluated on IOCfish5K, proving a domain gap between DOC and IOC.

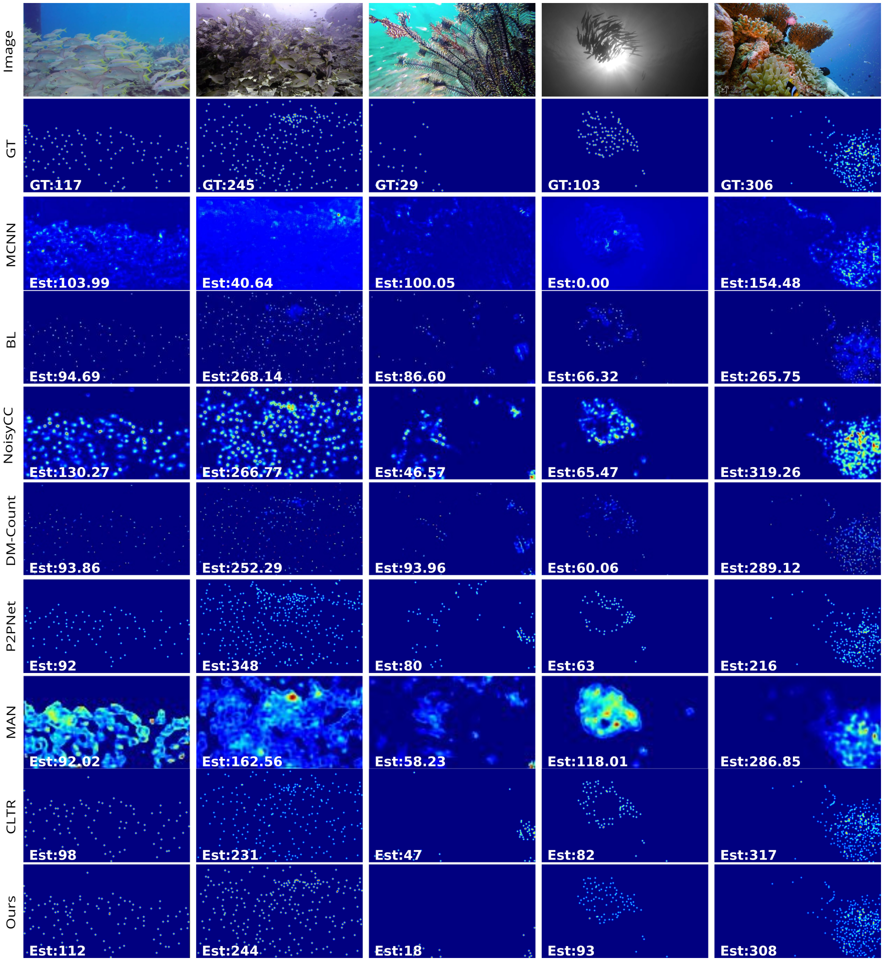


FIGURE 2.9: Qualitative comparisons of various algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

In addition, we propose a dedicated method for IOC named IOCFormer, exploiting single-image attention to enhance per-pixel features. Specifically, it is equipped with two novel designs: combining a density and regression branch in a unified model and a density-enhanced transformer encoder which transfers object density information from the density to the regression

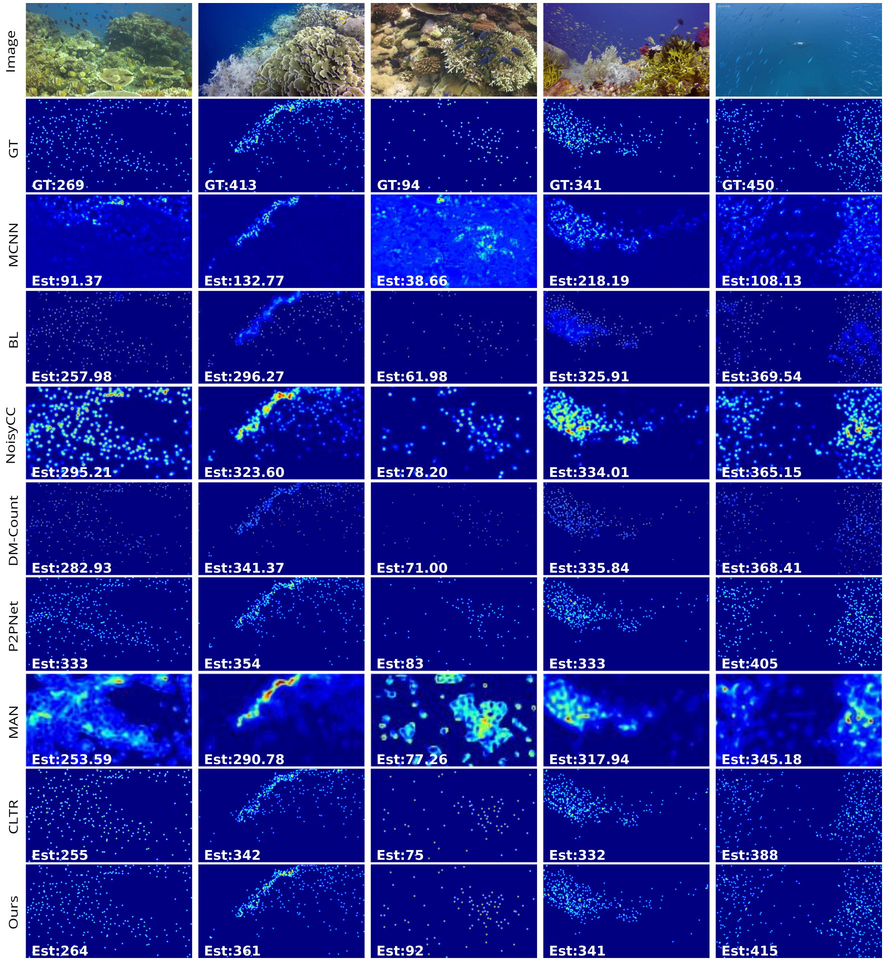


Figure 2.10: Qualitative comparisons of various algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

branch. IOCFormer achieves SOTA performance on IOCfish5K. To sum up, our dataset and method provide an opportunity for future researchers to dive into this new task.

For future work, there are several directions. (1) To improve performance and efficiency. Although our method achieves state-of-the-art performance,

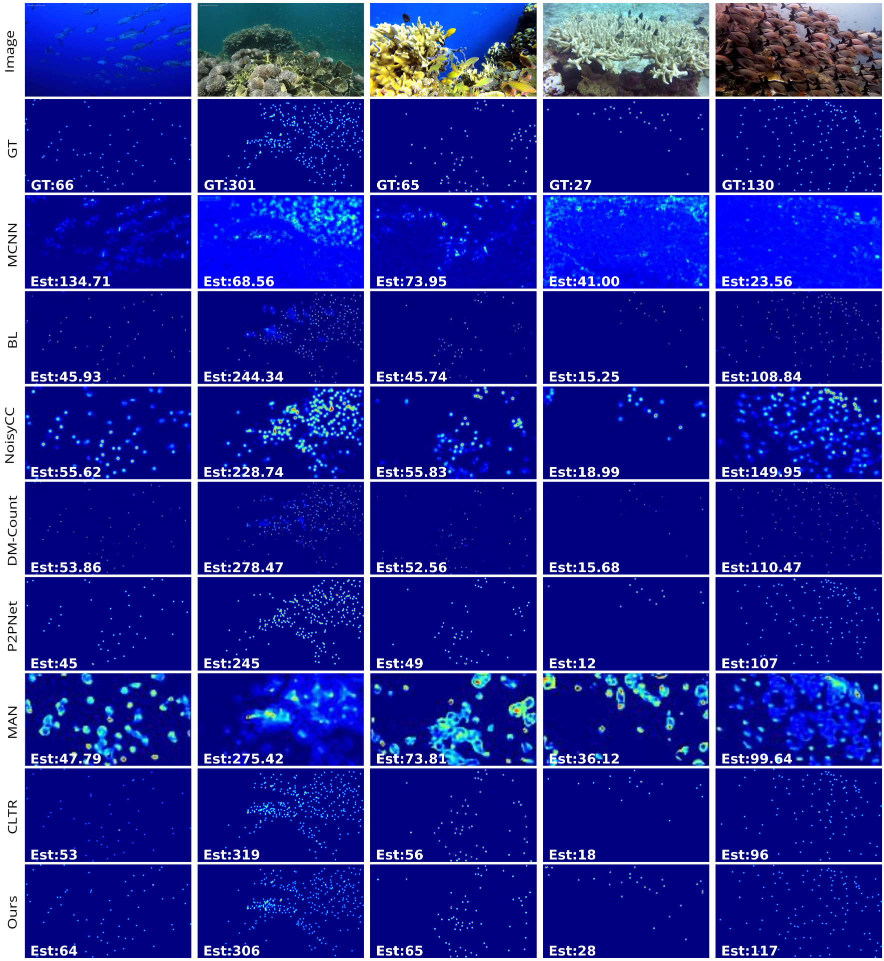


FIGURE 2.11: Qualitative comparisons of various algorithms (MCNN [72], BL [91], NoisyCC [92], DM-Count [93], P2PNet [94], MAN [96], CLTR [79], and ours). The GT or estimated counts for each case are shown in the lower left corner. Best viewed with zooming.

there is room to further improve the counting results on IOCfish5K in terms of MAE, MSE, and NAE. Also, efficiency is important when deploying counting models in real applications. (2) To study domain adaptation among IOC and DOC. There are many more DOC datasets than IOC datasets and how to improve IOC using available DOC datasets is a practical problem to tackle. (3) To obtain a general counting model which can count everything (people, plants, cells, fish, etc..).

3

MINING CROSS-IMAGE SEMANTICS FOR WEAKLY SUPERVISED SEMANTIC SEGMENTATION

3.1 INTRODUCTION

Recently, modern deep learning based semantic segmentation models [3, 40], trained with massive manually labeled data, achieve far better performance than before. However, the fully supervised learning paradigm has the main limitation of requiring intensive manual labeling effort, which is particularly expensive for annotating pixel-wise ground-truth for semantic segmentation. Numerous efforts are motivated to develop semantic segmentation with weaker forms of supervision, such as bounding boxes [26], scribbles [25], points [28], image-level labels [29], etc. Among them, a prominent and appealing trend is using only image-level labels to achieve weakly supervised semantic segmentation (WSSS), which demands the least annotation efforts and is followed in this work.

To tackle the task of WSSS with only image-level labels, current popular methods are based on network visualization techniques [125, 126], which discover discriminative regions that are activated for classification. These methods use image-level labels to train a classifier network, from which class-activation maps are derived as pseudo ground-truths for further supervising pixel-level semantics learning. However, it is commonly evidenced that the trained classifier tends to over-address the most discriminative parts rather than entire objects, which becomes the focus of this area. Diverse solutions are explored, typically adopting: *image-level* operations, such as region hiding and erasing [127, 128], *regions growing* strategies that expand the initial activated regions [129, 130], and *feature-level* enhancements that collect multi-scale context from deep features [131, 132].

These efforts generally achieve promising results, which demonstrates the importance of discriminative object pattern mining for WSSS. However, as shown in Fig. 3.1(a), they typically use only single-image information for object pattern discovering, ignoring the rich semantic context among the weakly annotated data. For example, with the image-level labels, not only the semantics of each individual image can be identified, the cross-image semantic relations, i. e., two images whether sharing certain semantics, are also given and should be used as cues for object pattern mining. Inspired by

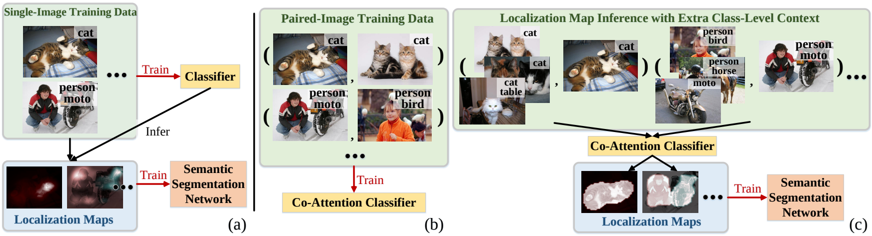


Figure 3.1: (a) Current WSSS methods only use single-image information for object pattern discovering. (b-c) Our co-attention classifier leverages cross-image semantics as class-level context to benefit object pattern learning and localization map inference.

this, rather than relying on *intra-image* information only, we further address the value of *cross-image* semantic correlations for complete object pattern learning and effective class-activation map inference (see Fig. 3.1(b-c)). In particular, our classifier is equipped with a differentiable co-attention mechanism that addresses semantic homogeneity and difference understanding across training *image pairs*. More specifically, two kinds of co-attentions are learned in the classifier. The former one aims to capture cross-image common semantics, which enables the classifier to better ground the common semantic labels over the co-attentive regions. The latter one, called contrastive co-attention, focuses on the rest, unshared semantics, which helps the classifier better separate semantic patterns of different objects. These two co-attentions work in a cooperative and complimentary manner, together making the classifier understand object patterns more comprehensively.

In addition to benefiting object pattern learning, our co-attention provides an efficient tool for precise localization map inference (see Fig. 3.1(c)). Given a training image, a set of related images (i. e., sharing certain common semantics) are utilized by the co-attention for capturing richer context and generating more accurate localization maps. Another advantage is that our co-attention based classifier learning paradigm brings an efficient data augmentation strategy, due to the use of training image pairs. Overall, our co-attention boosts object discovering during both the classifier's training phase as well as localization map inference stage. This provides the possibility of obtaining more accurate pseudo pixel-level annotations, which facilitate final semantic segmentation learning.

Our algorithm is a unified and elegant framework, which generalizes well different WSSS settings. Recently, to overcome the inherent limitation in WSSS without additional human supervision, some efforts resort to extra image-level supervision from simple single-class data readily available from other existing datasets [133, 134], or cheap web-crawled data [135–138]. Although they improve the performance to some extent, complicated techniques, such as energy function optimization [136, 139], heuristic constraints [137, 140], and curriculum learning [138], are needed to handle the challenges of domain gap and data noise, restricting their utility. However, due to the use of paired image data for classifier training and object map inference, our method has good tolerance to noise. In addition, our method also handles domain gap naturally, as the co-attention effectively addresses domain-shared object pattern learning and achieves domain adaption as a part of co-attention parameter learning. We conduct extensive experiments on PASCAL VOC 2012 [9], under three WSSS settings, i. e., learning WSSS with **(1)** PASCAL VOC image-level supervision only, **(2)** extra simple single-label data, and **(3)** extra web data. Our algorithm sets state-of-the-art on each case, verifying its effectiveness and generalizability.

3.2 RELATED WORK

Weakly Supervised Semantic Segmentation. Recently, lots of WSSS methods have been proposed to alleviate labeling cost. Various weak supervision forms have been explored, such as bounding boxes [26, 27], scribbles [25], point supervision [28], etc. Among them, image-level supervision, due to its less annotation demand, gains most attention and is also adopted in our approach.

Current popular solutions for WSSS with image-level supervision rely on network visualization techniques [125, 126], especially the Class Activation Map (CAM) [126], which discovers image pixels that are informative for classification. However, CAM typically only identifies small discriminative parts of objects. Therefore, numerous efforts are made towards expanding the CAM-highlighted regions to the whole objects. In particular, some representative approaches make use of *image-level* hiding and erasing operations to drive a classifier to focus on different parts of objects [127, 128, 141]. A few ones instead resort to a *regions growing* strategy, i. e., view the CAM-activated regions as initial "seeds" and gradually grow the seed regions until cover the complete objects [129, 130, 142, 143]. Meanwhile, some researchers investigate to directly enhance the activated regions on *feature-level* [30, 131, 132]. When constructing CAMs, they collect multi-scale context, which is achieved by dilated convolution [131], multi-layer feature fusion [132], saliency-guided iterative training [130], or stochastic feature selection [30]. Some others accumulate CAMs from multiple training phases [31], or self-train a difference detection network to complete the CAMs with trustable information [144]. In addition, a recent trend is to utilize class-agnostic saliency cues to filter out background responses [30, 128, 130, 131, 141, 142, 145] during pseudo ground-truth generation.

Since the supervision provided in above problem setting is so weak, another category of approaches explores to leverage more image-level supervision from other sources. There are mainly two types: (1) exploring simple and single-label examples [133, 134] (e. g., images from existing datasets [146, 147]); or (2) utilizing near-infinite yet noisy web-sourced image [135–138] or video [136, 139, 148] data (also referred as *webly supervised semantic segmentation* [149]). In addition to the common challenge of domain gap between the extra data and target semantic segmentation dataset, the second-type methods need to handle data noise.

Past efforts only consider each image individually, while only few exceptions [135, 145] address cross-image information. [135] simply applies off-the-shelf co-segmentation [150] over the web images to generate foreground priors, instead of ours encoding the semantic relations into network learning and inference. For [145], although also exploiting correlations within image pairs, the core idea is to use extra information from a support image to supplement current visual representations. Thus the two images are expected to better contain the same semantics, and unmatched semantics would bring negative influences. In contrast, we view both semantic homogeneity and difference as informative cues, driving our classifier to more explicitly identify the common as well as unshared objects, respectively. Moreover, [145] only utilizes single image to infer the activated objects, but our method comprehensively leverages the cross-image semantics in both classifier training and localization map inference stages. More essentially, our framework is neat and flexible, which is not only able to learn WSSS from clean image-level supervision, but general enough to naturally make use of extra noisy web-crawled or simple single-label data, contrarily to previous efforts which are limited to specific training settings and largely dependent on complicated optimization methods [136, 139] or heuristic constraints [137].

Deterministic Neural Attention. Differentiable attention mechanisms enable a neural network to focus more on relevant elements of the input than on irrelevant parts. With their popularity in the field of natural language processing [116, 151–154], attention modeling is rapidly adopted in various computer vision tasks, such as image recognition [18, 19, 155–157], domain adaptation [158, 159], human pose estimation [160–162], object detection [163] and image generation [164–166]. Further, co-attention mechanisms become an essential tool in many vision-language applications and sequential modeling tasks, such as visual question answering [167–170], visual dialog [171, 172], vision-language navigation [173], and video segmentation [174, 175], showing its effectiveness in capturing the underlying relations between different entities. Inspired by the general idea of attention mechanisms, this work leverages co-attention to mine semantic relations within training image pairs, which helps the classifier network learn complete object patterns and generate precise object localization maps.

3.3 METHODOLOGY

Problem Setup. Here we follow current popular WSSS pipelines: given a set of training images with image-level labels, a *classification network* is first trained to discover corresponding discriminative object regions. The resulting *object localization maps* over the training samples are refined as pseudo ground-truth masks to further supervise the learning of a *semantic segmentation network*.

Our Idea. Unlike most previous efforts that treat each training image *individually*, we explore cross-image semantic relations as class-level context for understanding object patterns more *comprehensively*. To achieve this, two neural co-attentions are designed. The first one drives the classifier to learn common semantics from the co-attentive object regions, while the other one enforces the classifier to focus on the rest objects for unshared semantics classification.

3.3.1 *Co-attention Classification Network*

Let us denote the training data as $\mathcal{I} = \{(\boldsymbol{I}_n, \boldsymbol{l}_n)\}_n$, where $\boldsymbol{I}_n$ is the n^{th} training image, and $\boldsymbol{l}_n \in \{0,1\}^K$ is the associated *ground-truth* image label for K semantic categories. As shown in Fig. 3.2(a), image pairs, i. e., $(\boldsymbol{I}_m, \boldsymbol{I}_n)$, are sampled from $\mathcal{I}$ for training the classifier. After feeding $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ into the convolutional embedding part of the classifier, corresponding feature maps, $\boldsymbol{F}_m \in \mathbb{R}^{C\times H\times W}$ and $\boldsymbol{F}_n \in \mathbb{R}^{C\times H\times W}$, are obtained, each with $H \times W$ spatial dimension and C channels.

As in [30, 31, 148], we can first separately pass $\boldsymbol{F}_m$ and $\boldsymbol{F}_n$ to a *class-aware fully convolutional layer* $\varphi(\cdot)$ to generate *class-aware activation maps*, i. e.,

$\boldsymbol{S}_m = \varphi(\boldsymbol{F}_m) \in \mathbb{R}^{K\times H\times W}$ and $\boldsymbol{S}_n = \varphi(\boldsymbol{F}_n) \in \mathbb{R}^{K\times H\times W}$, respectively. Then, we apply *global average pooling* (GAP) over $\boldsymbol{S}_m$ and $\boldsymbol{S}_n$ to obtain class score vectors $\boldsymbol{s}_m \in \mathbb{R}^K$ and $\boldsymbol{s}_n \in \mathbb{R}^K$ for $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, respectively. Finally, the *sigmoid cross entropy* (CE) loss is used for supervision:

$$\begin{aligned}\mathcal{L}^{mn}_{\text{basic}}\big((\boldsymbol{I}_m, \boldsymbol{I}_n), (\boldsymbol{l}_m, \boldsymbol{l}_n)\big) &= \mathcal{L}_{\text{CE}}(\boldsymbol{s}_m, \boldsymbol{l}_m) + \mathcal{L}_{\text{CE}}(\boldsymbol{s}_n, \boldsymbol{l}_n), \\ &= \mathcal{L}_{\text{CE}}\big(\text{GAP}(\varphi(\boldsymbol{F}_m)), \boldsymbol{l}_m\big) + \\ &\quad \mathcal{L}_{\text{CE}}\big(\text{GAP}(\varphi(\boldsymbol{F}_n)), \boldsymbol{l}_n\big).\end{aligned} \tag{3.1}$$

So far the classifier is learned in a standard manner, i. e., only individual-image information is used for semantic learning. One can directly use the activation maps to supervise next-stage semantic segmentation learning, as done in [142, 148]. Differently, our classifier additionally utilizes a co-attention mechanism for further mining cross-image semantics and eventually better localizing objects.

Co-Attention for Cross-Image Common Semantics Mining. Our co-attention attends to the two images, i. e., $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, simultaneously, and captures their correlations. We first compute the affinity matrix $\boldsymbol{P}$ between $\boldsymbol{F}_m$ and $\boldsymbol{F}_n$:

$$\boldsymbol{P} = \boldsymbol{F}_m^{\top} \boldsymbol{W}_{\boldsymbol{P}} \boldsymbol{F}_n \in \mathbb{R}^{HW\times HW}, \tag{3.2}$$

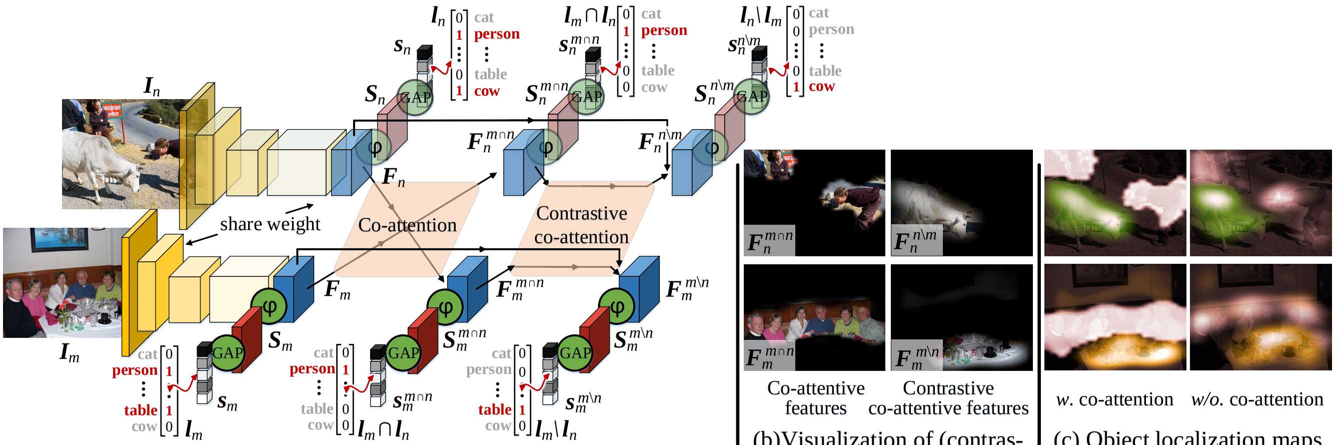
where $\boldsymbol{F}_m \in \mathbb{R}^{C\times HW}$ and $\boldsymbol{F}_n \in \mathbb{R}^{C\times HW}$ are flattened into matrix formats, and $\boldsymbol{W}_{\boldsymbol{P}} \in \mathbb{R}^{C\times C}$ is a learnable matrix. The affinity matrix $\boldsymbol{P}$ stores similarity scores corresponding to all pairs of positions in $\boldsymbol{F}_m$ and $\boldsymbol{F}_n$, i. e., the $(i, j)^{th}$ element of $\boldsymbol{P}$ gives the similarity between i^{th} location in $\boldsymbol{F}_m$ and j^{th} location in $\boldsymbol{F}_n$.

Then $\boldsymbol{P}$ is normalized column-wise to derive attention maps across $\boldsymbol{F}_m$ for each position in $\boldsymbol{F}_n$, and row-wise to derive attention maps across $\boldsymbol{F}_n$ for each position in $\boldsymbol{F}_m$:

$$\begin{aligned}\boldsymbol{A}_m &= \text{softmax}(\boldsymbol{P}) \in [0, 1]^{HW\times HW}, \\ \boldsymbol{A}_n &= \text{softmax}(\boldsymbol{P}^{\top}) \in [0, 1]^{HW\times HW},\end{aligned} \tag{3.3}$$

where softmax is performed column-wise. In this way, $\boldsymbol{A}_n$ and $\boldsymbol{A}_m$ store the co-attention maps in their columns. Next, we can compute attention summaries of $\boldsymbol{F}_m$ ($\boldsymbol{F}_n$) in light of each position of $\boldsymbol{F}_n$ ($\boldsymbol{F}_m$):

$$\begin{aligned}\boldsymbol{F}_m^{m\cap n} &= \boldsymbol{F}_n \boldsymbol{A}_n \in \mathbb{R}^{C\times H\times W}, \\ \boldsymbol{F}_n^{m\cap n} &= \boldsymbol{F}_m \boldsymbol{A}_m \in \mathbb{R}^{C\times H\times W},\end{aligned} \tag{3.4}$$



(a) Overview of our co-attention classifier during training phase

(b)Visualization of (contrastive) co-attentive features

(c) Object localization maps w. and *w/o*. co-attention

FIGURE 3.2: **(a)** In addition to mining object semantics from single-image labels, semantic similarities and differences between paired training images are both leveraged for supervising object pattern learning. **(b)** Co-attentive and contrastive co-attentive features complimentarily capture the shared and unshared objects. **(c)** Our co-attention classifier is able to learn object patterns more comprehensively. *Zoom-in for details*.

where $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$ are reshaped into $\mathbb{R}^{C\times W\times H}$. Co-attentive feature $\boldsymbol{F}_m^{m\cap n}$, derived from $\boldsymbol{F}_n$, preserves the common semantics between $\boldsymbol{F}_m$ and $\boldsymbol{F}_n$ and locate the common objects in $\boldsymbol{F}_m$. Thus we can expect only the common semantics $\boldsymbol{l}_m \cap \boldsymbol{l}_n$[1] can be safely derived from $\boldsymbol{F}_m^{m\cap n}$, and the same goes for $\boldsymbol{F}_n^{m\cap n}$. Such co-attention based common semantic classification can let the classifier understand the object patterns more completely and precisely.

To make things intuitive, consider the example in Fig. 3.2, where $\boldsymbol{I}_m$ contains **Table** and **Person**, and $\boldsymbol{I}_n$ has **Cow** and **Person**. As the co-attention is essentially the affinity computation between all the position pairs between $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, only the semantics of the common objects, **Person**, will be preserved in the co-attentive features, i. e., $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$ (see Fig. 3.2(b)). If we feed $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$ into the class-aware fully convolutional layer φ, the generated class-aware activation maps, i. e., $\boldsymbol{S}_m^{m\cap n} = \varphi(\boldsymbol{F}_m^{m\cap n}) \in \mathbb{R}^{K\times H\times W}$ and $\boldsymbol{S}_n^{m\cap n} = \varphi(\boldsymbol{F}_n^{m\cap n}) \in \mathbb{R}^{K\times H\times W}$, are able to locate the common object **Person** in $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, respectively. After GAP, the predicted semantic classes (scores) $\boldsymbol{s}_m^{m\cap n} \in \mathbb{R}^K$ and $\boldsymbol{s}_n^{m\cap n} \in \mathbb{R}^K$ should be the common semantic labels $\boldsymbol{l}_m \cap \boldsymbol{l}_n$ of $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, i. e., **Person**.

Through co-attention computation, not only the human face, the most discriminative part of **Person**, but also other parts, such as legs and arms, are highlighted in $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$ (see Fig. 3.2(b)). When we set the common class labels, i. e., **Person**, as the supervision signal, the classifier would realize that the semantics preserved in $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$ are related and can be used to recognize **Person**. Therefore, the co-attention, computed across two related images, *explicitly* helps the classifier associate semantic labels and corresponding object regions and better understand the relations between different object parts. It essentially makes full use of the context across training data.

Intuitively, for the co-attention based common semantic classification, the labels $\boldsymbol{l}_m \cap \boldsymbol{l}_n$ shared between $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are used to supervise learning:

$$\begin{aligned}\mathcal{L}_{\text{co-att}}^{mn}\big((\boldsymbol{I}_m, \boldsymbol{I}_n), (\boldsymbol{l}_m, \boldsymbol{l}_n)\big) &= \mathcal{L}_{\text{CE}}(\boldsymbol{s}_m^{m\cap n}, \boldsymbol{l}_m \cap \boldsymbol{l}_n) + \mathcal{L}_{\text{CE}}(\boldsymbol{s}_n^{m\cap n}, \boldsymbol{l}_m \cap \boldsymbol{l}_n), \\ &= \mathcal{L}_{\text{CE}}\big(\text{GAP}(\varphi(\boldsymbol{F}_m^{m\cap n})), \boldsymbol{l}_m \cap \boldsymbol{l}_n\big) + \\ &\quad \mathcal{L}_{\text{CE}}\big(\text{GAP}(\varphi(\boldsymbol{F}_n^{m\cap n})), \boldsymbol{l}_m \cap \boldsymbol{l}_n\big).\end{aligned} \tag{3.5}$$

Contrastive Co-Attention for Cross-Image Exclusive Semantics Mining. Aside from the co-attention described above that explores cross-image common semantics, we propose a contrastive co-attention that mines semantic

---

[1] The set operation '∩' is slightly extended here to represent bitwise-and.

differences between paired images. The co-attention and contrastive co-attention complementarily help the classifier better understand the concept of the objects.

As shown in Fig. 3.2(a), for $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, we first derive *class-agnostic co-attentions* from their co-attentive features, i. e., $\boldsymbol{F}_m^{m\cap n}$ and $\boldsymbol{F}_n^{m\cap n}$, respectively:

$$\begin{aligned} \boldsymbol{B}_m^{m\cap n} &= \sigma(\boldsymbol{W}_{\boldsymbol{B}}\boldsymbol{F}_m^{m\cap n}) \in [0,1]^{H\times W}, \\ \boldsymbol{B}_n^{m\cap n} &= \sigma(\boldsymbol{W}_{\boldsymbol{B}}\boldsymbol{F}_n^{m\cap n}) \in [0,1]^{H\times W}, \end{aligned} \tag{3.6}$$

where $\sigma(\cdot)$ is the *sigmoid* activation function, and the parameter matrix $\boldsymbol{W}_{\boldsymbol{B}} \in \mathbb{R}^{1\times C}$ learns for common semantics collection and is implemented by a convolutional layer with 1×1 kernel. $\boldsymbol{B}_m^{m\cap n}$ and $\boldsymbol{B}_n^{m\cap n}$ are class-agnostic and highlight all the common object regions in $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$, respectively, based on which we derive contrastive co-attentions:

$$\begin{aligned} \boldsymbol{A}_m^{m\backslash n} &= \boldsymbol{1} - \boldsymbol{B}_m^{m\cap n} \in [0,1]^{H\times W}, \\ \boldsymbol{A}_n^{n\backslash m} &= \boldsymbol{1} - \boldsymbol{B}_n^{m\cap n} \in [0,1]^{H\times W}. \end{aligned} \tag{3.7}$$

The contrastive co-attention $\boldsymbol{A}_m^{m\backslash n}$ of $\boldsymbol{I}_m$, as its superscript suggests, addresses those *unshared* object regions that are only of $\boldsymbol{I}_m$, but not of $\boldsymbol{I}_n$, and the same goes for $\boldsymbol{A}_n^{n\backslash m}$. Then we get *contrastive co-attentive features*, i. e., unshared semantics in each images:

$$\begin{aligned} \boldsymbol{F}_m^{m\backslash n} &= \boldsymbol{F}_m \otimes \boldsymbol{A}_m^{m\backslash n} \in \mathbb{R}^{C\times H\times W}, \\ \boldsymbol{F}_n^{n\backslash m} &= \boldsymbol{F}_n \otimes \boldsymbol{A}_n^{n\backslash m} \in \mathbb{R}^{C\times H\times W}. \end{aligned} \tag{3.8}$$

'$\otimes$' denotes element-wise multiplication, where the attention values are copied along the channel dimension. Next, we can sequentially get class-aware activation maps, i. e., $\boldsymbol{S}_m^{m\backslash n} = \varphi(\boldsymbol{F}_m^{m\backslash n}) \in \mathbb{R}^{K\times H\times W}$ and $\boldsymbol{S}_n^{n\backslash m} = \varphi(\boldsymbol{F}_n^{n\backslash m}) \in \mathbb{R}^{K\times H\times W}$, and semantic scores, i. e., $\boldsymbol{s}_m^{m\backslash n} = \mathrm{GAP}(\boldsymbol{S}_m^{m\backslash n}) \in \mathbb{R}^K$ and $\boldsymbol{s}_n^{n\backslash m} = \mathrm{GAP}(\boldsymbol{S}_n^{n\backslash m}) \in \mathbb{R}^K$. For $\boldsymbol{s}_m^{m\backslash n}$ and $\boldsymbol{s}_n^{n\backslash m}$, they are expected to identify the categories of the unshared objects, i. e., $\boldsymbol{l}_m \backslash \boldsymbol{l}_n$ and $\boldsymbol{l}_n \backslash \boldsymbol{l}_m$[2].

Compared with the co-attention that investigates common semantics as informative cues for boosting object patterns mining, the contrastive co-attention addresses complementary knowledge from the semantic differences between paired images. Fig. 3.2(b) gives an intuitive example. After computing the contrastive co-attentions between $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ (Eq. 3.7), **Table**

---

[2] The set operation '$\backslash$' is slightly extend here, i. e., $\boldsymbol{l}_n \backslash \boldsymbol{l}_m = \boldsymbol{l}_n - \boldsymbol{l}_n \cap \boldsymbol{l}_m$.

and **Cow**, which are unique in their original images, are highlighted. Based on the contrastive co-attentive features, i. e., $\boldsymbol{F}_m^{m\backslash n}$ and $\boldsymbol{F}_n^{n\backslash m}$, the classifier is required to accurately recognize **Table** and **Cow** classes, respectively. When the common objects are filtered out by the contrastive co-attentions, the classifier has a chance to focus more on the rest image regions and mine the unshared semantics more consciously. This also helps the classifier better discriminate the semantics of different objects, as the semantics of common objects and unshared ones are disentangled by the contrastive co-attention. For example, if some parts of **Cow** are wrongly recognized as **Person**-related, the contrastive co-attention will discard these parts in $\boldsymbol{F}_n^{n\backslash m}$. However, the rest semantics in $\boldsymbol{F}_n^{n\backslash m}$ may be not sufficient enough for recognizing **Cow**. This will enforce the classifier to better discriminate different objects.

For the contrastive co-attention based unshared semantic classification, the supervision loss is designed as:

$$\begin{aligned}\mathcal{L}_{\text{co-att}}^{mn}\big((\boldsymbol{I}_m,\boldsymbol{I}_n),(\boldsymbol{l}_m,\boldsymbol{l}_n)\big) =& \mathcal{L}_{\text{CE}}(\boldsymbol{s}_m^{m\backslash n},\boldsymbol{l}_m\backslash\boldsymbol{l}_n)+\mathcal{L}_{\text{CE}}(\boldsymbol{s}_n^{n\backslash m},\boldsymbol{l}_n\backslash\boldsymbol{l}_m),\\ =& \mathcal{L}_{\text{CE}}\big(\text{GAP}\big(\varphi(\boldsymbol{F}_m^{m\backslash n})\big),\boldsymbol{l}_m\backslash\boldsymbol{l}_n\big)+\\ & \mathcal{L}_{\text{CE}}\big(\text{GAP}\big(\varphi(\boldsymbol{F}_n^{n\backslash m}\big),\boldsymbol{l}_n\backslash\boldsymbol{l}_m\big).\end{aligned} \tag{3.9}$$

More In-Depth Discussion. One can interpret our co-attention classifier from a view of *auxiliary-task learning* [176, 177], which is investigated in self-supervised learning field to improve data efficiency and robustness, by exploring auxiliary tasks from inherent data structures. In our case, rather than the task of single-image semantic recognition which has been extensively studied in conventional WSSS methods, we explore two auxiliary tasks, i. e., predicting the common and uncommon semantics from image pairs, for fully mining supervision signals from weak supervision. The classifier is driven to better understand the cross-image semantics by attending to (contrastive) co-attentive features, instead of only relying on intra-image information (see Fig. 3.2(c)). In addition, such strategy shares a spirit of *image co-segmentation* [175, 178]. Since the image-level semantics of training set are given, the knowledge about some images share or unshare certain semantics should be used as a cue, or supervision signal, to better locate corresponding objects. Our co-attention based learning pipeline also provides an *efficient data augmentation* strategy, due to the use of paired samples, whose amount is near the square of the number of single training images.

3.3.2 *Co-Attention Classifier Guided WSSS Learning*

Training Co-Attention Classifier. The overall training loss for our co-attention classifier ensembles the three terms defined in Eq. 3.1, 3.5, and 3.9:

$$\mathcal{L}=\textstyle\sum_{m,n}\mathcal{L}^{mn}_{\text{basic}}+\mathcal{L}^{mn}_{\text{co-att}}+\mathcal{L}^{mn}_{\overline{\text{co-att}}}. \tag{3.10}$$

The coefficients of different loss terms are set as 1 in our all experiments. During training, to fully leverage the co-attention to mine the common semantics, we sample two images $(\boldsymbol{I}_m, \boldsymbol{I}_n)$ with at least one common class, i. e., $\boldsymbol{l}_m \cap \boldsymbol{l}_n \neq \boldsymbol{0}$.

Generating Object Localization Maps. Once our image classifier is trained, we apply it over the training data $\mathcal{I}=\{(\boldsymbol{I}_n, \boldsymbol{l}_n)\}_n$ to produce corresponding object localization maps, which are essential for semantic segmentation network training. We explore two different strategies to generate localization maps.

- *Single-round feed-forward prediction*, made over each training image individually. For each training image $\boldsymbol{I}_n$, running the classifier and directly using its class-aware activation map (i. e., $\boldsymbol{S}_n \in \mathbb{R}^{K\times H\times W}$) as the object localization map $\boldsymbol{L}_n$, as most previous network visualization based methods [31, 137, 148] done.
- *Multi-round co-attentive prediction with extra reference information*, which is achieved by considering extra information from other related training images (see Fig. 3.1(c)). Specifically, given a training image $\boldsymbol{I}_n$ and its associated label vector $\boldsymbol{l}_n$, we generate its localization map $\boldsymbol{L}_n$ in a *class-wise* manner. For each semantic class $k \in \{1, \cdots, K\}$ labeled for $\boldsymbol{I}_n$, i. e., $l_{n,k} = 1$ and $l_{n,k}$ is the k^{th} element of $\boldsymbol{l}_n$, we sample a set of related images $\mathcal{R} = \{\boldsymbol{I}_r\}_r$ from $\mathcal{I}$, which are also annotated with label k, i. e., $l_{r,k}=1$. Then we compute the co-attentive feature $\boldsymbol{F}_n^{m\cap r}$ from each related image $\boldsymbol{I}_r \in \mathcal{R}$ to $\boldsymbol{I}_n$, and get the co-attention based class-aware activation map $\boldsymbol{S}_n^{m\cap r}$. Given all the class-aware activation maps $\{\boldsymbol{S}_n^{m\cap r}\}_r$ from $\mathcal{R}$, they are integrated to infer the localization map *only* for class k, i. e., $L_{n,k}=\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} S_{n,k}^{m\cap r}$. Here $L_{n,k}\in\mathbb{R}^{H\times W}$ and $S_{n,k}^{(\cdot)}\in\mathbb{R}^{H\times W}$ indicate the feature map at k^{th} channel of $\boldsymbol{L}_n\in\mathbb{R}^{K\times H\times W}$ and $\boldsymbol{S}_n^{(\cdot)}\in\mathbb{R}^{K\times H\times W}$, respectively. '$|\cdot|$' numerates the elements. After inferring the localization maps for all the annotated semantic classes of $\boldsymbol{I}_n$, we can get $\boldsymbol{L}_n$.

These two localization map generation strategies are studied in our experiments (§3.4.5), and the last one is more favored, as it uses both intra- and inter-image semantics for object inference, and shares a similar data distribution of the training phase. One may notice that the contrastive co-attention is not used here. This is because contrastive co-attentive feature (Eq. 3.8) is from its original image, which is effective for boosting feature representation learning during classifier training, while contributes little for localization maps inference (with limited cross-image information). Related experiments can be found at §3.4.5.

Learning Semantic Segmentation Network. After obtaining high-quality localization maps, we generate pseudo pixel-wise labels for all the training samples $\mathcal{I}$, which can be used to train arbitrary semantic segmentation network. For pseudo groundtruth generation, we follow current popular pipeline [30, 31, 142, 148, 179, 180], that uses localization maps to extract class-specific object cues and adopts saliency maps [181, 182] to get background cues. For the semantic segmentation network, as in [30, 31, 148, 179], we choose DeepLab-LargeFOV [3].

Learning with Extra Simple Single-Label Images. Some recent efforts [133, 134] are made towards exploring extra simple single-label images from other existing datasets [146, 147] for further boosting WSSS. Though impressive, specific network designs are desired, due to the issue of domain gap between additionally used data and the target complex multi-label dataset, i. e., PASCAL VOC 2012 [9]. Interestingly, our co-attention based WSSS algorithm provides an alternate that addresses the challenge of domain gap naturally. Here we revisit the computation of co-attention in Eq. 3.2. When $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are from different domains, the parameter matrix $\boldsymbol{W_P}$, in essence, learns to map them into a unified *common semantic space* [183] and the co-attentive features can capture domain-shared semantics. Therefore, for such setting, we learn three different parameter matrixes for $\boldsymbol{W_P}$, for the cases where $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are from (1) the target semantic segmentation domain, (2) the one-label image domain, and (3) two different domains, respectively. Thus the domain adaption is efficiently achieved as a part of co-attention learning. We conduct related experiments in §3.4.2.

Learning with Extra Web Images. Another trend of methods [136–138, 149] address webly supervised semantic segmentation, i. e., leveraging web images as extra training samples. Though cheaper, web data are typically noisy. To handle this, previous arts propose diverse effective yet sophisticated solutions, such as multi-stage training [149] and self-paced learning [138]. Our co-attention based WSSS algorithm can be easily extended to this setting and solve data noise elegantly. As our co-attention classifier is trained with paired images, instead of previous methods only relying on each image individually, our model provides a more robust training paradigm.

Methods	Publication	Val	Test
Using PASCAL VOC data only			
DCSM [184]	ECCV16	44.1	45.1
SEC [129]	ECCV16	50.7	51.7
AFF [185]	ECCV16	54.3	55.5
DCSP [186]	BMVC17	60.8	61.9
CBTS [187]	CVPR17	52.8	53.7
AE-PSL [128]	CVPR17	55.0	55.7
Oh *et al.* [136]	CVPR17	55.7	56.7
TPL [188]	ICCV17	53.1	53.8
MEFF [189]	CVPR18	-	55.6
GAIN [141]	CVPR18	55.3	56.8
MDC [131]	CVPR18	60.4	60.8
MCOF [130]	CVPR18	60.3	61.2
DSRG [142]	CVPR18	61.4	63.2
PSA [143]	CVPR18	61.7	63.7
SeeNet [179]	NIPS18	63.1	62.8
IRN [190]	CVPR19	63.5	64.8
FickleNet [30]	CVPR19	64.9	65.3
SSDD [144]	ICCV19	64.9	65.5
OAA+ [31]	ICCV19	65.2	66.4
Ours	-	**66.2**	**66.9**

(a)

Methods	Publication	Val	Test
Using extra simple single-label images			
MCNN [139]	ICCV15	-	36.9
MIL-ILP [133]	CVPR15	32.6	-
MIL-sppxl [133]	CVPR15	36.6	35.8
MIL-bb [133]	CVPR15	37.8	37.0
MIL-seg [133]	CVPR15	42.0	40.6
AttnBN [134]	ICCV19	62.1	63.0
Ours	-	**67.1**	**67.2**

(b)

Methods	Publication	Val	Test
Using extra noisy web images/videos			
MCNN [139]	ICCV15	38.1	39.8
Shen *et al.* [135]	BMVC17	56.4	56.9
STC [138]	PAMI17	49.8	51.2
Hong *et al.* [136]	CVPR17	58.1	58.7
WebS-i1 [149]	CVPR17	51.6	-
WebS-i2 [149]	CVPR17	53.4	55.3
Shen *et al.* [137]	CVPR18	63.0	63.9
Ours	-	**67.7**	**67.5**

(c)

TABLE 3.1: Experimental results for WSSS under three different settings. **(a)** Standard setting where only PASCAL VOC 2012 images are used (§3.4.1). **(b)** Additional single-label images are used (§3.4.2). **(c)** Additional web-crawled images are used (§3.4.3).

In addition, during localization map inference, a set of extra related images are considered, which provides more comprehensive and accurate cues, and further improves the robustness. We experimentally demonstrate the effectiveness of our method in such a setting in §3.4.3.

3.3.3 *Detailed Network Architecture*

Network Configuration. In line with conventions [31, 131, 191], our image classifier is based on ImageNet [192] pre-trained VGG-16 [6]. For VGG-

16 network, the last three fully-connected layers are replaced with three convolutional layers with 512 channels and kernel size 3×3, as done in [31, 191]. For the semantic segmentation network, for fair comparison with current top-leading methods [30, 31, 143, 144], we adopt the ResNet-101 [5] version Deeplab-LargeFOV architecture.

Training Phases of the Co-Attention Classifier and Semantic Segmentation Network. Our co-attention classifier is fully end-to-end trained by minimizing the loss defined in Eq. 3.10. The training parameters are set as: initial learning rate (0.001) which is reduced by 0.1 after every 5 epochs, batch size (5), weight decay (0.0002), and momentum (0.9). Once the classifier is trained, we generate localization maps and pseudo segmentation masks over all the training samples (see §3.3.2). Then, with the masks, the semantic segmentation network is trained in a standard way [31] using the hyper-parameter setting in [3].

Inference Phase of the Semantic Segmentation Network. Given an *unseen* test image, our segmentation network works in the *standard* semantic segmentation pipeline [3], i. e., directly generating segments without using any other images. Then CRF [193] post-processing is performed to refine predicted masks.

3.4 EXPERIMENT

Overview. Experiments are first conducted over *three* different WSSS settings: **(1)** The most standard paradigm [31, 128, 142, 144] that only allows image-level supervision from PASCAL VOC 2012 [9] (see §3.4.1). **(2)** Following [133, 134], additional single-label images can be used, yet bringing the challenge of domain gap (see §3.4.2). **(3)** Webly supervised semantic segmentation paradigm [137, 148, 149], where extra web data can be accessed (see §3.4.3). Then, in §3.4.4, we show the results in WSSS track of LID_{20}, where our method achieves the champion. Finally, in §3.4.5, ablation studies are made to assess the effectiveness of essential parts of our algorithm.

Evaluation Metric. In our experiments, the standard intersection over union (IoU) criterion is reported on the val and test sets of PASCAL VOC 2012 [9]. The scores on test set are obtained from official PASCAL VOC evaluation server.

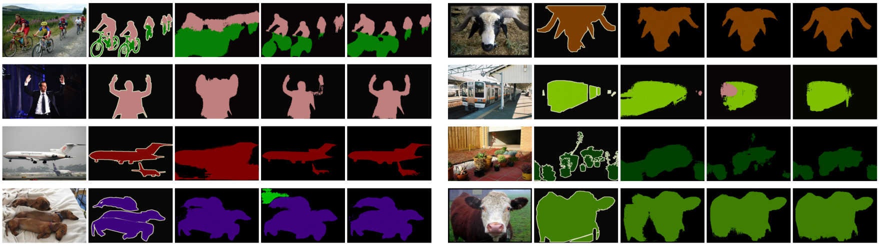


FIGURE 3.3: Visual comparison results on PASCAL VOC12 val set. From *left* to *right*: input image, ground truth, results for PSA [143], OAA+ [31], and our method.

3.4.1 *Experiment 1: Learn WSSS only from PASCAL VOC Data*

Experimental Setup: We first conduct experiment following the most standard setting that learns WSSS with only image-level labels [31, 128, 142, 144], i. e., only image-level supervision from PASCAL VOC 2012 [9] is accessible. PASCAL VOC 2012 contains a total of 20 object categories. As in [3, 128], augmented training data from [194] are also used. Finally, our model is trained on totally 10,582 samples with only image-level annotations. Evaluations are conducted on the val and test sets, which have 1,449 and 1,456 images, respectively.

Experimental Results: Table 3.1a compares our approach and current top-leading WSSS methods with image-level supervision, on both PASCAL VOC12 val and test sets. Additionally, we show some segmentation results in Fig. 3.3. We can observe that our method achieves mIoU scores of 66.2 and 66.9 on val and test sets respectively, outperforming all the competitors. The performance of our method is 87% of the DeepLab-LargeFOV [3] trained with fully annotated data, which achieved an mIoU of 76.3 on val set. When compared to OAA+ [31], current best-performing method, our approach obtains the improvement of 1.0% on val set. This verifies that the localization maps produced by our co-attention classifier effectively detect more complete semantic regions towards the whole target objects. Note that our network is elegantly trained end-to-end in a single phase. In contrast, many other recent approaches use extra networks [31, 143, 144] to learn auxiliary information (e. g., integral attention [31], pixel-wise semantic affinity [144], etc.), or adopt multi-step training [128, 131, 190].

Method	Inference Mode	Input Image(s)	Val
Basic Classifier	Single-round feed-forward	Test image *only*	61.7
Our Variant	Single-round feed-forward	Test image *only*	64.7
	Multi-round co-attention and contrastive co-attention	Test image and other related images	66.2
Full Model	Multi-round co-attention	Test image and other related images	**66.2**

TABLE 3.2: Ablation study for different object localization map generate strategies, reported on PASCAL VOC12 val set. See §3.4.5 for details.

3.4.2 *Experiment 2: Learn WSSS with Extra Simple Single-Label Data*

Experimental Setup: Following [133, 134], we train our co-attention classifier and segmentation network with PASCAL images and extra single-label images. The extra single-label images are borrowed from the subsets of Caltech-256 [147] and ImageNet CLS-LOC [146], and whose annotations are within 20 VOC object categories. There are a total of 20,057 extra single-label images.

Experimental Results: The comparisons are shown in Table 3.1b. Our method significantly improves the most recent method (i. e., AttnBN [134]) in this setting by 5.0% and 4.2% in val and test sets, respectively. With the fact that objects of the same category but from different domains share similar visual patterns [134], our co-attention provides an end-to-end strategy that efficiently captures the common, cross-domain semantics, and learns domain adaption naturally. Even AttnBN is specifically designed for addressing such setting by knowledge transfer, our method still suppresses it by a large margin. Compared with the setting in §3.4.1 where only PASCAL images are used for training, our method obtains improvements on both val and test sets, verifying that it successfully mines knowledge from extra simple single-label data and copes with domain gap well.

3.4.3 *Experiment 3: Learn WSSS with Extra Web-Sourced Data*

Experimental Setup: We also conduct experiments using both PASCAL VOC images and webly craweled images as training data. We use the web data provided by [137], which are retrieved from Bing based on class names. The final dataset contains 76,683 images across 20 PASCAL VOC classes.

Experimental Results: Table 3.1c gives performance comparisons with previous webly supervised segmentation methods. As seen our method outperforms all other approaches and sets new state-of-the-arts with mIoU score of 67.7 and 67.5 on PASCAL VOC 2012 val and test sets, respectively. Among the compared methods, Hong *et al.* [136] utilize richer information of the temporal dynamics provided by additional large-scale videos. In contrast, although only using static data, our method still outperforms it on the val and test sets by 9.6% and 8.8%, respectively. Compared with Shen *et al.* [137] using the same web data as ours, our method substantially improves it by a clear margin of 3.6% on the test set.

3.4.4 *Experiment 4: Performance on WSSS Track of* LID_{20} *Challenge*

Experimental Setup: The challenge dataset [195] is built upon ImageNet [146]. It contains 349,319 images with image-level labels from 200 classes. Evaluations are conducted on the val and test sets, which have 4,690 and 10,000 images, respectively. In this challenge, our co-attention image classifier is built upon ResNet-38 [196], as the dataset has 200 classes and a stronger backbone can better learn subtle semantics between classes. The training parameters are set as: initial learning rate (0.005) and the poly policy based training schedule: $lr = lr_{init} \times (1 - \frac{iter}{max\_iter})^{\gamma}$ with γ(0.9), batch size (8), weight decay (0.0005), and max epoch (15). During training, the equivariant attention [197] is also adopted. Once our image classifier is trained, we run the classifier and directly use its class-aware activation map (i. e., $\boldsymbol{S}_n$) as the object localization map $\boldsymbol{L}_n$. Then we generate pseudo pixel-wise labels for all the training samples $\mathcal{I}$. Since only image tags can be used, we follow [143]: localization maps are first used to train an AffinityNet model, which is then used to generate pseudo ground truth masks and background threshold is set as 0.2. For better segmentation results, we choose ResNet-101 based DeepLab-V3. The parameters are set as below: initial learning rate (0.007) with poly schedule, batch size (48), max epoch (100), and weight decay (0.0001). The segmentation model is trained on 4 Tesla V100 GPUs. During testing, results from multiple scales are averaged, with CRF refinement.

Experimental Results: The final results with the standard mean intersection over union (mIoU) criterion for WSSS track of both LID_{19} and LID_{20} challenges are shown in Table 3.3. Both LID_{19} and LID_{20} challenge use the same data. In LID_{19}, competitors can use extra saliency annotations to learn saliency models and refine pseudo ground truths. However, in LID_{20}, only

Year	Team	Extra Saliency Annotation	Val	Test
LID_{19}	T.T (T.T)	✓	-	8.1
	LEAP_DEXIN	✓	20.7	19.6
	MVN	✓	41.0	40.0
LID_{20}	play-njupt	✗	22.1	31.9
	IOnlyHaveSevenDays	✗	39.0	36.2
	UCU & SoftServe	✗	39.7	37.3
	VL-task1	✗	40.1	37.7
	CVL (**ours**)	✗	**46.2**	**45.1**

TABLE 3.3: Results on *val* and *test* sets of both LID_{19} and LID_{20} WSSS track.

image tags can be accessed. For methods shown in the table, top performing methods are included. As can be seen from Table 3.3, our approach not only outperforms the champion team in LID_{19}, which can use deep learning based saliency models, but also achieves the best performance in LID_{20} and sets a new state-of-the-art (i. e., mIoU of 46.2 and 45.1 in val and test sets, respectively).

3.4.5 *Ablation Studies*

Inference Strategies. Table 3.2 shows mIoU scores on PASCAL VOC 2012 val set with respect to different inference modes (see §3.3.2). When using the traditional inference mode “single-round feed-forward", our method substantially suppresses basic classifier, by improving mIoU score from 61.7 to 64.7. This evidences that co-attention mechanism (trained in an end-to-end manner) in our classifier improves the underlying feature representations and more object regions are identified by the network. We can observe that by using more images to generate localization maps, our method obtains consistent improvement from “Test image *only*” (64.7), to “Test images and other related images” (66.2). This is because more semantic context are exploited during localization map inference. In addition, using contrastive co-attention for localization map inference doesn’t boost performance (66.2). This is because the contrastive co-attentive features for one image are derived from the image itself. In contrast, co-attentive features are from the other related image, thus can be effective in the inference stage.

(Contrastive) Co-Attention. As seen in Table 3.4, by only using co-attention (Eq. 3.5), we already largely suppress the basic classifier (Eq. 3.1) by 3.8%.

Method	(Contrastive) Co-Attention	Training Loss	Val
Basic Classifier	-	$\mathcal{L}_{\text{basic}}$ (Eq. 3.1)	61.7
Our Variant	co-attention *only*	$\mathcal{L}_{\text{basic}}$ (Eq. 3.1)+$\mathcal{L}_{\text{co-att}}$ (Eq. 3.5)	65.5
Full Model	co-attention +contrastive co-attention	$\mathcal{L}_{\text{basic}}$ (Eq. 3.1)+$\mathcal{L}_{\text{co-att}}$ (Eq. 3.5)+$\mathcal{L}_{\overline{\text{co-att}}}$ (Eq. 3.9) = $\mathcal{L}$ (Eq. 3.10)	**66.2**

TABLE 3.4: Ablation study for our co-attention and contrastive co-attention mechanisms for training, reported on PASCAL VOC12 val set. See §3.4.5 for details.

Method	Extra Related Images (#)	Val
Our Variant	0	64.7
	1	65.9
	2	66.0
	4	66.1
	5	66.0
Full Model	3	**66.2**

TABLE 3.5: Ablation study for using different numbers of related images during object localization map generation, reported on PASCAL VOC12 val set (see §3.4.5).

When adding additional contrastive co-attention (Eq. 3.9), we obtain mIoU improvement of 0.7%. Above analysis verify our two co-attentions indeed boost performance.

Number of Related Images for Localization Map Inference. For localization map generation, we use 3 extra related images (§3.3.2). Here, we study how the number of reference images affect the performance. From Table 3.5, it is easily observed that when increasing the number of related images from 0 to 3, the performance gets boosted consistently. However, when further using more images, the performance degrades. This can be attributed to the trade-off between useful semantic information and noise brought by related images. From 0 to 3 reference images, more semantic information is used and more integral regions for objects are mined. When further using more related images, useful information reaches its bottleneck and noise, caused by imperfect localization of the classifier, takes over, decreasing performance.

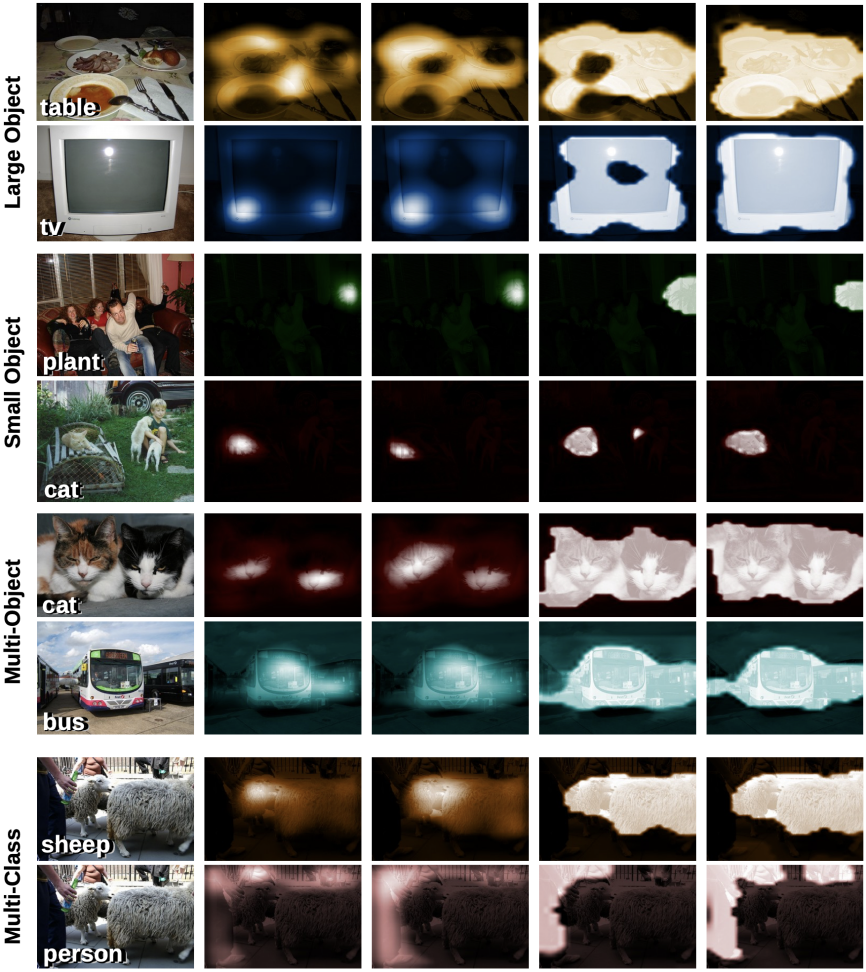


Figure 3.4: Localization maps for different methods. From *left* to *right*: input image, localization maps from basic classifier, OAA+, basic classifier+co-attention, and our full model (basic classifier+co-attention+contrastive co-attention)

3.4.6 *Additional Visual Results*

The quality of localization maps determines the performance of final models. Here, we visually compare the localization maps produced by different

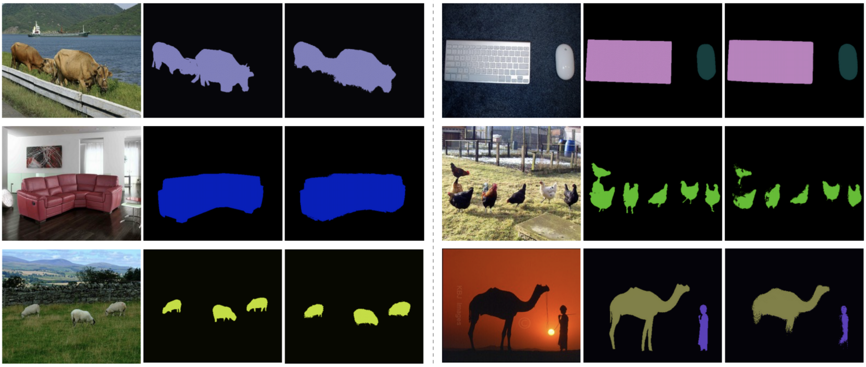


Figure 3.5: Visual results on WSSS track of LID_{20} Challenge. From *left* to *right*: input image, ground-truth mask, and our prediction.

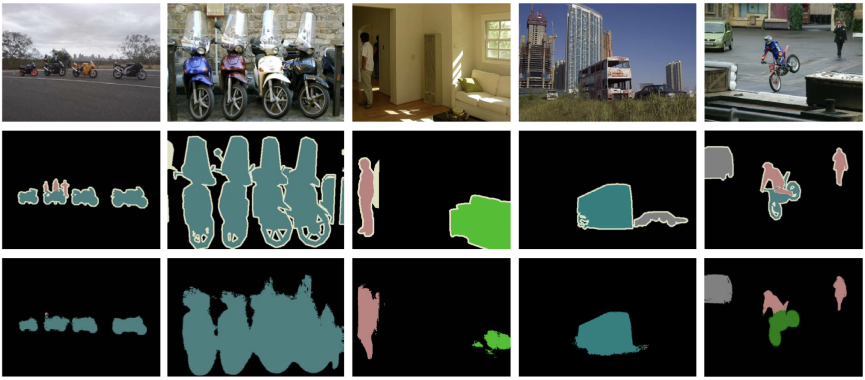


Figure 3.6: Failure cases. For *top* to *bottom*: input image, ground-truth mask, and predicted mask

methods in Fig. 3.4. It shows that our final model generates excellent localization maps which cover more complete object regions.

We show additional visual results on LID_{20} Challenge dataset in Fig. 3.5. For these cases, the predicted masks are of high quality and very close to the ground truth, even though the proposed method is only trained with image-level labels.

Though our proposed WSSS framework gains improved performance over previous methods, it still faces difficulties in some challenging scenes. We show a few representative failure cases of the proposed method in

Fig. 3.6. First, our model may fail to capture poorly visible objects. In the 1_{st} column of Fig. 3.6, the humans are too small to be recognized. In the 3_{rd} and 4_{th} columns of Fig. 3.6, the poor visibility is caused by low contrast to the background (i.e., a large portion of the white sofa is mistakenly merged into the background) and significant occlusion (i.e., the black car is totally missing). Second, our method does not work well with transparent objects. An example is the motorcycles in the 2_{nd} column of Fig. 3.6, whose windshields are hard to be recognized. In addition, our method sometimes cannot accurately predict object semantics, even when it has already generated precise segmentation masks. As shown in the last column of Fig. 3.6, the motorcycle, though being successfully highlighted, is wrongly recognized as a bike.

3.5 CONCLUSION

This work proposes a co-attention classification network to discover integral object regions by addressing cross-image semantics. With this regard, a co-attention is exploited to mine the common semantics within paired samples, while a contrastive co-attention is utilized to focus on the exclusive and unshared ones for capturing complimentary supervision cues. Additionally, by leveraging extra context from other related images, the co-attention boosts localization map inference. Further, by exploiting additional single-label images and web images, our approach is proven to generalize well under domain gap and data noise. Experiments over three WSSS settings consistently show promising results.

4

MINING RELATIONS AMONG CROSS-FRAME AFFINITIES FOR VIDEO SEMANTIC SEGMENTATION

4.1 INTRODUCTION

Semantic segmentation aims at assigning a semantic label to each pixel in a natural image, which is a fundamental and hot topic in the computer vision community. It has wide range of applications in both academic and industrial fields. Thanks to the powerful representation capability of deep neural networks [5–7, 192] and large-scale image datasets [9, 11, 198, 199], tremendous achievements have been seen for image semantic segmentation. However, video semantic segmentation has not been witnessed such tremendous progress [46–49] due to the lack of large-scale datasets. For example, Cityscapes [198] and NYUDv2 [200] datasets only annotate one or several nonadjacent frames in a video clip. CamVid [201] only has a small scale and a low frame rate. The real world is actually dynamic rather than static, so the research on video semantic segmentation (VSS) is necessary. Fortunately, the recent establishment of the large-scale video segmentation dataset, VSPW [202], solves the problem of video data scarcity. This inspires us to denoting our efforts to VSS.

As widely accepted, the contextual information plays a central role in image semantic segmentation [3, 4, 32–45]. When considering videos, the contextual information is twofold: *static contexts* and *motional contexts*, as shown in Fig. 4.1. The former refers to the contexts within the same video frame or the contexts of unchanged content across different frames. Image semantic segmentation has exploited such contexts (for images) a lot, mainly accounting for multi-scale [3, 39, 40, 42] and global/long-range information [4, 32, 43, 44]. Such information is essential not only for understanding the static scene but also for perceiving the holistic environment of videos. The latter, also known as temporal information, is responsible for better parsing moving object/stuff and capturing more effective scene representations with the help of motions. The motional context learning has been widely studied in video semantic segmentation [46–58], which usually relies on optical flows [61] to model *motional contexts*, ignoring the *static contexts*. Although each single aspect, i. e., *static* or *motional contexts*, has

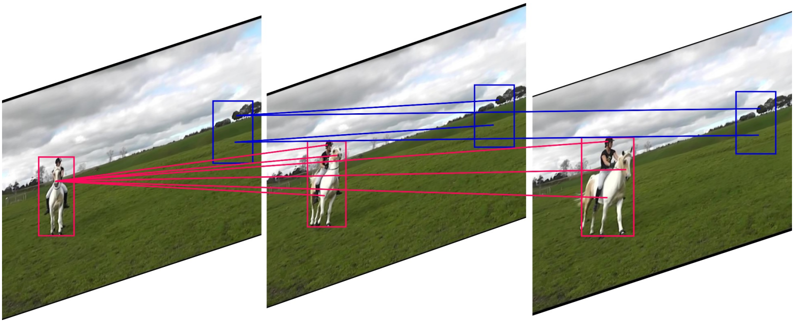


Figure 4.1: Illustration of *static contexts* (in blue) and *motional contexts* (in red) across neighbouring video frames. The human and horse are moving objects, while the grassland and sky are static background. Note that the static stuff is helpful for the recognition of moving objects, i. e., a human is riding a horse on the grassland.

been well studied, how to learn static and motional contexts simultaneously deserves more attention, which is important for VSS.

Furthermore, *static contexts* and *motional contexts* are highly correlated, not isolated, because both contexts are complementary to each other to represent a video clip. Therefore, the ideal solution for VSS is to jointly learn *static* and *motional contexts*, i. e., generating a unified representation of *static* and *motional contexts*. A naïve solution is to apply recent popular self-attention [85, 116, 157] by taking feature vectors at all pixels in neighboring frames as tokens. This can directly model global relationships of all tokens, of course including both static and motional contexts. However, this naïve solution has some obvious drawbacks. For example, it is super inefficient due to the large number of tokens/pixels in a video clip, making this naïve solution unrealistic. It also contains too much redundant computation because most content in a video clip usually does not change much and it is unnecessary to compute attention for the repeated content. Moreover, the too long length of tokens would affect the performance of self-attention, as shown in [16, 203–205] where the reduction of the token length through downsampling leads to better performance. More discussion about why traditional self-attention is inappropriate for video context learning can be found in §4.3.1.

In this chapter, we propose a new Coarse-to-Fine Feature Mining (CFFM) technique, which consists of two parts: coarse-to-fine feature assembling and cross-frame feature mining. Specifically, we first apply a lightweight deep network [15] to extract features from each frame. Then, we assemble the extracted features from neighbouring frames in a coarse-to-fine manner. Here, we use a larger receptive field and a more coarse pooling if the frame is more distant from the target. This feature assembling operation has two meanings. On one hand, it organizes the features in a multi-scale way, and the farthest frame would have the largest receptive field and the most coarse pooling. Since the content in a few sequential frames usually does not change suddenly and most content may only have a little temporal inconsistency, this operation is expected to prepare data for learning *static contexts*. On the other hand, this feature assembling operation enables a large perception region for remote frames because the moving objects may appear in a large region for remote frames. This makes it suitable for learning *motional contexts*. At last, with the assembled features, we use the cross-frame feature mining technique to iteratively mine useful information from neighbouring frames for the target frame. This mining technique is a specially-designed non-self attention mechanism that has two different inputs, unlike commonly-used self-attention that only has one input [85, 116]. The output features enhanced by the CFFM can be directly used for the final prediction. We describe the technical motivations for CFFM in detail in §4.3.1.

The advantages of this new video context learning mechanism are fourfold. **(1)** The proposed CFFM technique can learn a unified representation of *static contexts* and *motional contexts*, both of which are of vital importance for VSS. **(2)** The CFFM technique can be added on top of frame feature extraction backbones to generate powerful video contextual features, with low complexity and limited computational cost. **(3)** Without bells and whistles, we achieve state-of-the-art results for VSS on standard benchmarks by using the CFFM module. **(4)** The CFFM technique has the potential to be extended to improve other video recognition tasks that need powerful video contexts.

4.2 RELATED WORK

4.2.1 *Image Semantic Segmentation*

Image semantic segmentation has always been a key topic in the vision community, mainly because of its wide applications in real-world scenarios. Since the pioneer work of FCN [206] which adopts fully convolution networks to make densely pixel-wise predictions, a number of segmentation methods have been proposed with different motivations or techniques [207–213]. For example, some works try to design effective encoder-decoder network architectures to exploit multi-level features from different network layers [40, 206, 214–217]. Some works impose extra boundary supervision to improve the prediction accuracy of details [45, 218–221]. Some works utilize the attention mechanism to enhance the semantic representations [43, 44, 155, 222–224]. Besides these talent works, we want to emphasize that most research aims at learning powerful contextual information [32–38, 41, 45], including multi-scale [3, 37, 39, 40, 42, 225] and global/long-range information [4, 32, 43, 44]. The contextual information is also essential for VSS, but the video contexts are different from the image contexts, as discussed above.

4.2.2 *Video Semantic Segmentation*

Since the real world is dynamic rather than static, VSS is necessary for pushing semantic segmentation into more practical deployments. Previous research on VSS was limited by the available datasets [202]. Specifically, three datasets were available: Cityscapes [198], NYUDv2 [200], and CamVid [201]. They either only annotate several nonadjacent frames in a video clip or have a small scale, a low frame rate and low resolution. In fact, these datasets are usually used for image segmentation. Fortunately, the recent establishment of the VSPW dataset [202] which is large-scale and fully-annotated solves this problem.

Most of the existing VSS methods utilize the optical flow to capture temporal relations [46–48, 50, 51, 53, 54, 56, 58, 226, 227]. These methods usually adopt different smart strategies to balance the trade-off between accuracy and efficiency [226, 227]. Among them, some works aim at improving the segmentation accuracy by exploiting the temporal relations using the optical flow for feature warping [46–48] or the GAN-like architecture [228] for predictive feature learning [49]. The other works aim at improving the segmentation efficiency by using temporal consistency for feature propagation and reuse [53, 54, 56, 57], or directly reusing high-level features [53, 55], or adaptively selecting the key frame [50], or propagating segmentation results to neighbouring frames [58], or extracting features from different frames with different sub-networks [52], or considering the temporal consistency as extra training constraints [51]. Zhu *et al.* [229] utilized video prediction models to predict future frames as well as future segmentation labels, which are used as augmented data for training better image semantic segmentation models, not for VSS. Different from the above approaches, STT [60] and LMANet [59] directly models the interactions between the target and reference features to exploit the temporal information.

The above VSS approaches explore the temporal relation, here denoted as *motional contexts*. However, video contexts include two highly-correlated aspects: *static* and *motional contexts*. Those methods ignore the *static contexts* that are important for segmenting complicated scenes. This chapter addresses this problem by proposing a new video context learning mechanism, capable of joint learning a unified representation of *static* and *motional contexts*.

4.2.3 *Transformer*

Vision transformer, a strong competitor of convolutional neural networks (CNNs), has been widely adopted in various vision tasks [14, 16, 17, 85, 230–234], due to its powerful ability of modeling global connection within all the input tokens. Specifically, ViT [85] splits an image into patches to construct tokens and processes tokens using typical transformer layers. Swin Transformer [16] improves ViT by introducing shifted windows when computing self-attention. Focal Transformer [230] introduces both fine-grained and coarse-grained attention in architecture design. The effectiveness of transformers has been validated in tracking [235, 236], crowd counting [14, 237], multi-label classification [238] and so on. In the following, we specifically discuss the transformer-based segmentation methods.

To improve segmentation using transformers, some methods [15, 233, 239–241] have been developed. SETR [233] is one of the first transformer-based models for image semantic segmentation. Generally, these works use transformers to generate global-context-aware features. Differently, a new trend of works such as MaskFormer [240] and Mask2Former [241] use transformer decoders to get rid of the conventional per-pixel classification for segmentation. Despite the success of transformers in segmentation, the use of transformer layers in VSS is non-trivial due to the large number of tokens from video frames. Here, we propose an effective and efficient way to model the temporal contextual information for VSS.

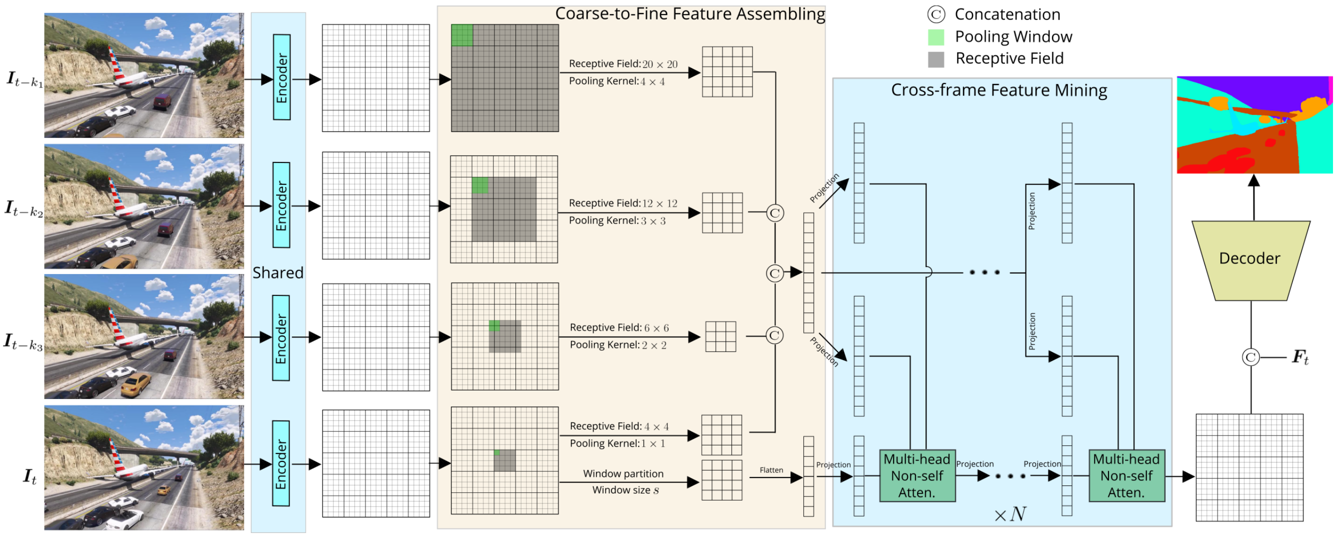

I_{t-k_1}
I_{t-k_2}
I_{t-k_3}
I_t
Encoder
Shared
Coarse-to-Fine Feature Assembling
Receptive Field: 20×20
Pooling Kernel: 4×4
Receptive Field: 12×12
Pooling Kernel: 3×3
Receptive Field: 6×6
Pooling Kernel: 2×2
Receptive Field: 4×4
Pooling Kernel: 1×1
Window partition
Window size s
Flatten
Projection
Ⓒ Concatenation
Pooling Window
Receptive Field
Cross-frame Feature Mining
Multi-head Non-self Atten.
$\times N$
Decoder
F_t

FIGURE 4.2: Overview of the proposed Coarse-to-Fine Feature Mining. All frames are first input to an encoder to extract features, which then go through the coarse-to-fine feature assembling module (CFFA). Features for different frames are processed by different pooling strategies to generate the context tokens. The principle is that for more distant frames, the bigger receptive field and more coarse pooling are used. The shown feature size (20×20), receptive field and pooling kernel are for simple explanation. The context tokens from all frames are concatenated and then processed by cross-frame feature mining (CFM) module. The context tokens are exploited to update the target features by several multi-head non-self attention layers. Finally, we use the enhanced target features to make segmentation prediction for the target frame. *Best viewed with zooming.*

4.3 METHODOLOGY

4.3.1 *Technical Motivation*

Before introducing our method, we discuss our technical motivation to help readers better understand the proposed technique. As discussed above, video contexts include *static contexts* and *motional contexts*. The former is well exploited in image semantic segmentation [3, 4, 32–45, 225], while the latter is studied in video semantic segmentation [46–48, 50, 51, 53–58, 226, 227]. However, there is no research touching the joint learning of both *static* and *motional contexts* which are both essential for VSS.

To address this problem, a naïve solution is to simply apply the recently popular self-attention mechanism [85, 116, 157] to the video sequence by viewing the feature vector at each pixel of each frame as a token. In this way, we can model global relationships by connecting each pixel with all others, so all video contexts can of course be constructed. However, this naïve solution has *three obvious drawbacks*. First, a video sequence has l times more tokens than a single image, where l is the length of the video sequence. This would lead to l^2 times more computational cost than a single image because the complexity of the self-attention mechanism is $\mathcal{O}(N^2C)$, where N is the number of tokens and C is the feature dimension [16, 85, 116]. Such high complexity is unaffordable, especially for VSS that needs on-time processing as video data stream comes in sequence. Second, such direct global modeling would be redundant. Despite that there are some motions in a video clip, the overall semantics/environment would not change suddenly and most video content is repeated. Hence, most of connections built by the direct global modeling are unnecessary, i. e., self-to-self connections. Last but not least, although self-attention can technically model global relationships, a too long sequence length would limit its performance, as demonstrated in [16, 203–205, 242] where downsampling features into small scales leads to better performance than the original long sequence length.

Instead of directly modeling global relationships, we propose to model relationships only among necessary tokens for the joint learning of static and motional contexts. Our CFFM technique consists of two steps. The first step, Coarse-to-Fine Feature Assembling (CFFA), assembles the features extracted from neighbouring frames in a temporally coarse-to-fine manner based on *three observations*. First, the moving objects/stuff can only move gradually across frames in practice, and the objects/stuff cannot move from one position to another far position suddenly. Thus, the region of the possible positions of (an) moving object/stuff in a frame gradually gets larger for farther frames. In other words, for one pixel in a frame, the farther the frames, the larger the correlated regions. Second, although some content may change across frames, the overall semantics and environment would not change much, which means that most video content may only have a little temporal inconsistency. For statistical evidence, we compute the mIoU between the ground-truth masks of consecutive video frames on the VSPW val set [202], to show that the semantic masks for consecutive frames are largely overlapped and the scene changes are thus very small from a frame to its next frame. The obtained mIoU is 89.7%, proving that the objects/background move slowly from frame to frame. Third, the little temporal inconsistency of the "static" content across neighbouring frames can be easily handled by the pooling operation which is scale- and rotation-invariant, as evidenced in previous works [4, 32, 36, 206]. Inspired by the second and third observations, a varied-size region sampling through the pooling operation in neighbouring frames can convey multi-scale contextual information. Therefore, the designed CFFA can perceive multi-scale contextual information (*static contexts*) and *motional contexts*. Specifically, each pixel in the target frame corresponds to a larger receptive field and a more coarse pooling in the farther frame, as depicted in Fig. 4.2. Note that the length of the sampled tokens is much shorter than that in the default self-attention.

The second step of CFFM, Cross-frame Feature Mining (CFM), is designed to mine useful information from the features of neighbouring frames. This is an attention-based process. However, unlike traditional self-attention [85, 116, 157] whose query, key, and value come from the same input, we propose to use a *non-self attention* mechanism, where the query is from the target frame and the key and value are from neighbouring frames. Besides, we only update the query during the iterative running of non-self attention, but we keep the context tokens unchanged. This is intuitive as our goal is to mine information from neighbouring frames and the update of context tokens is thus unnecessary. Compared with self-attention that needs to concatenate and process all assembled features, this non-self attention further reduces the computational cost.

4.3.2 *Coarse-to-Fine Feature Assembling*

Without loss of generalizability, we start our discussion on training data containing video frames $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}, \boldsymbol{I}_t\}$ with ground-truth segmentation of $\{\boldsymbol{S}_{t-k_1}, \cdots, \boldsymbol{S}_{t-k_l}, \boldsymbol{S}_t\}$, and we focus on segmenting $\boldsymbol{I}_t$. Specifically, $\boldsymbol{I}_t$ is the target frame and $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}\}$ are l previous frames which are $\{k_1, \cdots, k_l\}$ frames away from $\boldsymbol{I}_t$, respectively. Let us denote $U = \{t - k_1, \cdots, t - k_l, t\}$ as the set of all frame subscripts. We first process $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}, \boldsymbol{I}_t\}$ using an encoder to extract informative features $\boldsymbol{F} = \{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$, each of which has the size of $\mathbb{R}^{h \times w \times c}$ (h, w, and c represent height, width, and the number of channels, respectively). We aim to exploit $\boldsymbol{F}$ to generate better features for segmenting $\boldsymbol{I}_t$ as relevant and valuable video contexts exist in previous frames.

To efficiently establish long-range interactions between the reference frame features ($\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}\}$) and the target frame features $\boldsymbol{F}_t$, we propose the coarse-to-fine feature assembling module, as showed in Fig. 4.2. Inspired by previous works [16, 230, 242], we split the target frame features $\boldsymbol{F}_t$ into windows and each window attends to a shared set of context tokens. The reason behind this is that attending each location in $\boldsymbol{F}_t$ to a specific set of context tokens requires huge computation and memory cost. When using window size of $s \times s$, $\boldsymbol{F}_t$ is partitioned into $\frac{h}{s} \times \frac{w}{s}$ windows. We obtain the new feature map $\boldsymbol{F}_t^{'}$ as follows:

$$\boldsymbol{F}_t \in \mathbb{R}^{h \times w \times c} \rightarrow \boldsymbol{F}_t' \in \mathbb{R}^{(\frac{h}{s} \times s) \times (\frac{w}{s} \times s) \times c} \rightarrow \boldsymbol{F}_t' \in \mathbb{R}^{\frac{h}{s} \times \frac{w}{s} \times s \times s \times c}. \tag{4.1}$$

Then, we generate context tokens from different frames. The main idea is to see a bigger receptive field and use a more coarse pooling if the frame is more distant from the target, which is why we call this step coarse-to-fine feature assembling. The motivation behind this is described in §4.3.1. Formally, we define two sets of parameters as follows: the receptive fields $r = \{r_{t-k_1}, \cdots, r_{t-k_l}, r_t\}$, and the pooling kernel/window sizes $p = \{p_{t-k_1}, \cdots, p_{t-k_l}, p_t\}$, when generating corresponding context tokens. For $t - k_1 < t - k_2 < \cdots < t - k_l < t$, we have $r_{t-k_1} \geq r_{t-k_2} \geq \cdots \geq r_{t-k_l} \geq r_t$ and $p_{t-k_1} \geq p_{t-k_2} \geq \cdots \geq p_{t-k_l} \geq p_t$. With this definition, we partition $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$ using pooling windows $p = \{p_{t-k_1}, \cdots, p_{t-k_l}, p_t\}$ to pool the features, respectively. The result is processed by a fully connected layer (FC) for dimension reduction. This is formulated as

$$\boldsymbol{F}_j \in \mathbb{R}^{h \times w \times c} \rightarrow \boldsymbol{E}_j \in \mathbb{R}^{\frac{h}{p_j} \times \frac{w}{p_j} \times c \times p_j^2} \xrightarrow{\text{FC}} \boldsymbol{E}_j \in \mathbb{R}^{\frac{h}{p_j} \times \frac{w}{p_j} \times c}, \tag{4.2}$$

where $j \in U$. In Fig. 4.2, we have $r = \{20, 12, 6, 4\}$ and $p = \{4, 3, 2, 1\}$ for all frames (3 reference and 1 target).

For each window partition $\boldsymbol{F}_t'[i] \in \mathbb{R}^{s \times s \times c}$ ($i \in \{1, 2, \cdots, \frac{hw}{s^2}\}$) in the target features, we extract $\frac{r_j}{p_j} \times \frac{r_j}{p_j}$ elements from $\boldsymbol{E}_j$ around the area where the window lies in. This can be easily implemented using the *unfold* function in PyTorch [121]. Let $\boldsymbol{c}_{i,j}$ denote the obtained context tokens from j-th frame and for i-th window partition in the target features. We concatenate $\boldsymbol{c}_{i,j}$ into $\boldsymbol{c}_i$ as follows,

$$\boldsymbol{c}_i = \text{Concat}[\boldsymbol{c}_{i,j}], \tag{4.3}$$

where $j \in U$, $\boldsymbol{c}_i \in \mathbb{R}^{m \times c}$ and $m = \sum_{j \in U} \frac{r_j^2}{p_j^2}$. The context tokens from the target frame are obtained by using parameter set (r_t, p_t) to process the target features. In practice, we additionally use another parameter set ($r_t^{'}$, $p_t^{'}$) to generate more contexts from the target since the target features are more important. For simplicity, we focus our discussion by omitting ($r_t^{'}$, $p_t^{'}$) and using only (r_t, p_t) for the target.

To sum up, $\boldsymbol{c}_i$ contains the context information from all frames, which is used to refine the target frame features. As discussed in §4.3.1, on one hand, $\boldsymbol{c}_i$ covers the tokens at possible positions that moving objects/stuff would appear, so it can be used for learning *motional contexts*. On the other hand, $\boldsymbol{c}_i$ is a multi-scale sampling of neighbouring frames with the temporal inconsistency solved by the pooling operation, so it can be used for learning *static contexts*.

4.3.3 *Cross-frame Feature Mining*

After that we obtain the context tokens $\boldsymbol{c}_i$, for each window partition in the target features, we propose a non-self attention mechanism to mine useful information from neighboring frames. Unlike the traditional self-attention mechanism that computes the query, key, and value from the same input, our non-self attention mechanism utilizes different inputs to calculate the query, key, and value. Since $\boldsymbol{F}_t^{'}$ is the input to the first layer of our cross-frame feature mining module, we re-write it as $\boldsymbol{F}_t^0 = \boldsymbol{F}_t^{'}$. For the i-th window partition in $\boldsymbol{F}_t^0$, the query Q_i, key K_i, and value V_i are computed using three fully connected layers as follows:

$$Q_i = \text{FC}(\boldsymbol{F}_t^0[i]), \quad K_i = \text{FC}(\boldsymbol{c}_i), \quad V_i = \text{FC}(\boldsymbol{c}_i), \tag{4.4}$$

where FC$(\cdot)$ represents a FC layer. Next, we use non-self attention to update the target frame features, given by

$$\boldsymbol{F}_t^1[i] = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{c}} + B)V_i + \boldsymbol{F}_t^0[i], \tag{4.5}$$

where B represents the position bias, following [16]. Note that we omit the formulation of the multi-head attention [85, 116] for simplicity. Eq. (4.4) and Eq. (4.5) are repeated for N steps, and we finally obtain the enhanced feature $\boldsymbol{F}_t^N \in \mathbb{R}^{\frac{h}{s}\times\frac{w}{s}\times s\times s\times c}$ for the target frame. Long-range static and motional contexts from neighbouring frames are continuously exploited to learn better features for segmenting the target frame. Note that in this process, we do not update the context tokens $\boldsymbol{c}_i$ for simplicity/elegance and reducing computation. Since this step is to mine useful information from the reference frames, it is also unnecessary to update $\boldsymbol{c}_i$. This is the advantage of non-self attention.

To generate segmentation predictions, we reshape $\boldsymbol{F}_t^N$ into $\mathbb{R}^{h\times w\times c}$ and concatenate $\boldsymbol{F}_t^N$ with $\boldsymbol{F}_t$. Then, a simple MLP projects the features to segmentation logits. The common cross entropy (CE) is used as the loss function for training. Auxiliary losses on original features are also computed. During inference, our method does not need to extract features for all $l+1$ frames when processing $\boldsymbol{I}_t$. Instead, the features of the reference frames, which are the frames before the target frame, have already been extracted in previous steps. Only the target frame is passed to the encoder to generate $\boldsymbol{F}_t$, and then features $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$ for all frames are passed to the CFFM.

4.3.4 *Complexity Analysis*

Here, we formally analyze the complexity of the proposed CFFM and the recent popular self-attention mechanism [85, 116, 157] when processing video clip features $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$. The coarse-to-fine feature assembling (Eq. (4.2)) has the complexity of $\mathcal{O}((l+1)hwc)$, which is irrespective of p. The cross-frame feature mining has two parts: Eq. (4.4) has the complexity of $\mathcal{O}(hwc^2) + \mathcal{O}(mc^2)$, and Eq. (4.5) is with the complexity of $\mathcal{O}(hwmc)$. As mentioned early, $m = \sum_{j\in U}\frac{r_j^2}{p_j^2}$. To sum over, the complexity of our method is given by

$$\begin{aligned}\mathcal{O}(\text{CFFM}) &= \mathcal{O}(hwmc) + \mathcal{O}(hwc^2) + \mathcal{O}(mc^2) + \mathcal{O}((l+1)hwc)\\ &= \mathcal{O}(hwmc) + \mathcal{O}(hwc^2),\end{aligned} \tag{4.6}$$

where the derivation is conducted by removing less significant terms. For the self-attention mechanism [85, 116, 157], the complexity is $\mathcal{O}((l+1)^2h^2w^2c) + \mathcal{O}((l+1)hwc^2)$. Since $m \ll (l+1)^2hw$, the complexity of the proposed approach is much less than the self-attention mechanism. Take the example in Fig. 4.2, $m = 66$ while $(l+1)^2hw = 6400$.

4.3.5 *Difference with STT*

We notice that a concurrent work STT [60] also utilizes bigger searching regions for more distant frames and self-attention mechanisms to establish connections across frames. While two works share these similarities, there are key differences between them. *First*, two methods have different motivations. We target at exploiting both *static* and *motional* contexts, while STT focuses on capturing the temporal relations among complex regions. Note that the concept of static/motional contexts is similar to the concept of simple/complex regions in STT. As a result, STT models only the motional contexts, while our method models both *static* and *motional* contexts. *Second*, the designs are different. For query selection, STT selects 50% of query locations in order to reduce the computation. However, our method splits the query features into windows and the query features in each window share the same contexts to reduce the computation. For key/value selection, STT operates in the same granularity, while our method processes the selected key/value into different granularity, which reduces the number of tokens and models the multi-scale information for static contexts. *Third*, our cross-frame feature mining can exploit multiple transformer layers to deeply mine the contextual information from the reference frames, but STT only uses one layer. The reason may be that STT only updates the query features of the selected locations and using multiple STT layers could lead to inconsistency in the query features in un-selected and selected locations.

4.4 EXPERIMENTS

4.4.1 *Experimental Setup*

Implementation details. We implement our approach based on the public *mmsegmentation* [243] toolbox and conduct all experiments on 4 NVIDIA GPUs. The backbones are the same as SegFormer [15], which are all pre-trained on ImageNet [192]. For other parts of our model, we adopt random initialization. Our model uses 3 reference frames unless otherwise specified, and $\{k_1, k_2, k_3\} = \{9, 6, 3\}$, following [202]. We found that this selection of reference frames is enough to include rich context and achieve impressive performance. For the receptive field, pooling kernel and window size, we set $r = \{49, 20, 6, 7\}$, $p = \{7, 4, 2, 1\}$ and $s = 7$. For the target frame, we additionally have $r_t^{'} = 35$ and $p_t^{'} = 5$. During training, we adopt augmentations including random resizing, flipping, cropping, and photometric distortion. We use the crop size of 480×480 for the VSPW dataset [202] and 512×1024 for Cityscapes [198]. For optimizing parameters, we use the AdamW and "poly" learning rate schedule, with an initial learning rate of 6*e*-5. During testing, we conduct single-scale test and resize all images on VSPW to the size of 480×853 and 512×1024 for Cityscapes. Note that for efficiency and simplicity, the predicted mask is obtained by feeding the whole image to the network, rather than using sliding window as in [233]. We do *not* use any post-processing such as CRF [193].

Datasets. Our experiments are mainly conducted on the VSPW dataset [202], which is the largest video semantic segmentation benchmark. Its training, validation and test sets have 2,806 clips (198,244 frames), 343 clips (24,502 frames), and 387 clips (28,887 frames), respectively. It contains diverse scenarios including both indoor and outdoor scenes, annotated for 124 categories. More importantly, VSPW has dense annotations with a high frame rate of 15fps, making itself the best benchmark for video semantic segmentation till now. In contrast, previous datasets used for video semantic segmentation only have very sparse annotation, i. e., only one frame out of many consecutive frames is annotated. In addition, we also evaluate the proposed method on the Cityscapes dataset [198], which annotates one frame out of every 30 frames.

Evaluation metrics. Following previous works [206], we use mean IoU (mIoU), and weigheted IoU to evaluate the segmentation performance. In addition, we also adopt video consistency (VC) [202] to evaluate the smoothness of the predicted segmentation maps across the temporal domain. Formally, for a video clips $\{\boldsymbol{I}_c\}_{c=1}^{C}$ with ground truth masks $\{\boldsymbol{S}_c\}_{c=1}^{C}$ and predicted masks $\{\boldsymbol{S}_c^{'}\}_{c=1}^{C}$, VC_n is computed as follows,

$$\mathrm{VC}_n = \frac{1}{C - n + 1} \sum_{i=1}^{C-n+1} \frac{(\cap_i^{i+n-1} \boldsymbol{S}_i) \cap (\cap_i^{i+n-1} \boldsymbol{S}_i^{'})}{\cap_i^{i+n-1} \boldsymbol{S}_i}, \tag{4.7}$$

where $C \geq n$. After computing VC_n for every video, we obtain the mean of VC_n for all videos as mVC_n. The purpose of this metric is to evaluate the level of consistency in the predicted masks among those common areas (pixels' semantic labels don't change) across long-range frames. For more details, please refer to [202]. Note that to compute VC metric, the ground-truth masks for all frames are needed.

The details of computing FPS are as follows. The FPS is measured in mini-batches with the batch size set to 2. We keep note of the computation time $\mathcal{T}$ for processing $\mathcal{K}$ mini-batches. The FPS can be calculated by $2\mathcal{K}/\mathcal{T}$. We set the batch size to 2 because this leads to high usage (>95%) of GPU, which is common in this community. We computed the FPS for all methods in the same way for fair comparisons.

4.4.2 *Comparison with State-of-the-art Methods*

We compare the proposed method with state-of-the-art algorithms on VSPW [202] in Tab. 4.1. The results are analyzed from different aspects. For small models (# of parameters $<$ 20M), our method outperforms corresponding baseline with a clear margin, while introducing limited model complexity. For example, using the backbone MiT-B0, we obtain 2.5% mIoU gain over the strong baseline of SegFormer [15], with the cost of increasing the parameters from 13.8M to 15.5M and reducing the FPS from 73.4 to 43.1. Our method also provides much more consistent predictions for the videos, outperforming the baseline with 5.0% and 5.6% in mVC_8 and mVC_{16}, respectively.

For large models, our approach achieves the new state-of-the-art performance in this challenging dataset and also generates visually consistent results. Specifically, our model with 26.5M parameters (slightly larger than SegFormer [15] with MiT-B2) achieves 44.9% mIoU at the frame rate of 23.8fps. Our large model (based on MiT-B5) achieves mIoU of 49.3% and performs best in terms of visual consistency, with mVC_8 and mVC_{16} of 90.8% and 87.1%, respectively. For all backbones (MiT-B0, MiT-B1, MiT-B2 and MiT-B5), CFFM clearly outperforms the corresponding baselines, showing that the proposed modules are stable. The results validate the effectiveness of the proposed coarse-to-fine feature assembling and cross-frame feature mining in mining informative contexts from all frames.

For Cityscapes [198] dataset, our method is compared with recent efficient segmentation methods. Only using 4.6M parameters, our model obtains 74.0% mIoU with frame rate of 34.2fps, achieving an excellent balance on model size, performance and speed. When using deeper backbone, we achieve 75.1% mIoU with the frame rate of 23.6fps. Note that this dataset has sparse annotations, the excellent performance demonstrates that our method works well for both fully supervised and semi-supervised settings.

Methods	Backbone	Params (M) ↓	mIoU ↑	Weighted IoU ↑	mVC_8 ↑	mVC_{16} ↑	FPS (f/s) ↑
SegFormer [15]	MiT-B0	3.8	32.9	56.8	82.7	77.3	73.4
SegFormer [15]	MiT-B1	13.8	36.5	58.8	84.7	79.9	58.7
CFFM (Ours)	MiT-B0	4.7	35.4	58.5	87.7	82.9	43.1
CFFM (Ours)	MiT-B1	15.5	**38.5**	**60.0**	**88.6**	**84.1**	29.8
DeepLabv3+ [39]	ResNet-101	62.7	34.7	58.8	83.2	78.2	-
UperNet [244]	ResNet-101	83.2	36.5	58.6	82.6	76.1	-
PSPNet [4]	ResNet-101	70.5	36.5	58.1	84.2	79.6	13.9
OCRNet [35]	ResNet-101	58.1	36.7	59.2	84.0	79.0	14.3
ETC [51]	PSPNet	89.4	36.6	58.3	84.1	79.2	-
NetWarp [244]	PSPNet	89.4	37.0	57.9	84.4	79.4	-
ETC [51]	OCRNet	58.1	37.5	59.1	84.1	79.1	-
NetWarp [244]	OCRNet	58.1	37.5	58.9	84.0	79.0	-
$TCB_{st\text{-}ppm}$ [202]	ResNet-101	70.5	37.5	58.6	87.0	82.1	10.0
$TCB_{st\text{-}ocr}$ [202]	ResNet-101	58.1	37.4	59.3	86.9	82.0	5.5
$TCB_{st\text{-}ocr\text{-}mem}$ [202]	ResNet-101	58.1	37.8	59.5	87.9	84.0	5.5
SegFormer [15]	MiT-B2	24.8	43.9	63.7	86.0	81.2	39.2
SegFormer [15]	MiT-B5	82.1	48.2	65.1	87.8	83.7	17.2
CFFM (Ours)	MiT-B2	26.5	44.9	64.9	89.8	85.8	23.8
CFFM (Ours)	MiT-B5	85.5	**49.3**	**65.8**	**90.8**	**87.1**	11.3

TABLE 4.1: Comparison with state-of-the-art methods on the VSPW [202] validation set. Our model outperforms the compared methods, with better balance in terms of model size, performance and speed.

Methods	Backbone	Params (M)	mIoU	FPS (f/s)
FCN [206]	MobileNetV2	9.8	61.5	14.2
CC [53]	VGG-16	-	67.7	16.5
DFF [56]	ResNet-101	-	68.7	9.7
GRFP [47]	ResNet-101	-	69.4	3.2
PSPNet [4]	MobileNetV2	13.7	70.2	11.2
DVSN [50]	ResNet-101	-	70.3	19.8
Accel [54]	ResNet-101	-	72.1	3.6
ETC [51]	ResNet-18	13.2	71.1	9.5
SegFormer [15]	MiT-B0	3.7	71.9	58.5
CFFM (Ours)	MiT-B0	4.6	74.0	34.2
SegFormer [15]	MiT-B1	13.8	74.1	46.8
CFFM (Ours)	MiT-B1	15.4	**75.1**	23.6

Table 4.2: Comparison with recent efficient video semantic segmentation methods on the Cityscapes [198] dataset.

Methods	Backbone	N	mIoU	mVC_8	mVC_{16}	Params (M)
SegFormer [15]	MiT-B0	-	32.9	82.7	77.3	3.8
CFFM (Ours)	MiT-B0	1	35.4	**87.7**	82.9	4.7
	MiT-B0	2	**35.7**	**87.7**	**83.0**	5.5
SegFormer [15]	MiT-B1	-	36.5	84.7	79.9	13.8
CFFM (Ours)	MiT-B1	1	37.8	88.3	83.6	14.6
	MiT-B1	2	38.5	**88.6**	**84.1**	15.5
	MiT-B1	3	38.7	**88.6**	**84.1**	16.3
	MiT-B1	4	**38.8**	88.5	83.9	17.2

Table 4.3: Ablation study on the number of attention layers in cross-frame feature mining module.

The qualitative results are shown in Fig. 4.3. For the given example, our method resolves the inconsistency existing in the predictions of the baseline, due to the use of rich contextual information from the all frames.

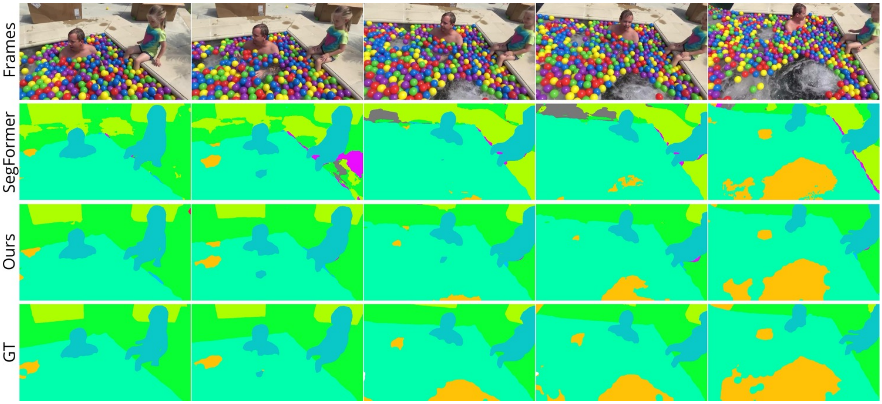


FIGURE 4.3: Qualitative results. We compare the proposed method with the baseline (SegFormer [15]) visually. From *top* to *down*: the input video frames, the predictions of SegFormer [15], our predictions, and the ground truth. It shows that our model produces more accurate and consistent results, compared to the strong baseline. *Best viewed in color.*

Methods	k1	k2	k3	mIoU	mVC_8	mVC_{16}
SegFormer	-	-	-	36.5	84.7	79.9
CFFM (Ours)	-	-	3	37.4	87.4	82.4
	-	-	6	37.7	88.0	83.3
	-	-	9	37.9	88.4	83.9
	3	2	1	37.7	88.3	83.6
	9	6	3	**38.5**	**88.6**	**84.1**

TABLE 4.4: Ablation study on the selection of the reference frames. We use MiT-B1 as the backbone.

4.4.3 *Ablation Study*

All ablation studies are conducted on the large-scale VSPW [202] dataset and follows the same training strategies as described above, for fair comparison.

Influence of the number of attention layers. Tab. 4.3 shows the performance of our method with respect to the number of non-self attention layers in cross-frame feature mining module. For two backbones of MiT-B0 [15] and MiT-B1 [15], our method clearly outperforms the corresponding baseline (SegFormer) when only using a single attention layer and introducing

Methods	mIoU	mVC_8	mVC_{16}
Baseline	36.5	84.7	79.9
Baseline +static contexts	37.7	84.4	79.4
Baseline +static/motional contexts	**38.5**	**88.6**	**84.1**

Table 4.5: Ablation study on static and motional contexts.

a small amount of additional parameters. It demonstrates the effectiveness of the proposed coarse-to-fine feature assembling module and the attention layer. The former efficiently extracts the context information from the frames and the latter effectively mine the information to refine target features. In addition, we observe there is a trade-off between performance and the model complexity on MiT-B1 backbone. When using more attention layers, better mIoU is obtained while the model size linearly increases. For our method on MiT-B1, we choose $N = 2$ since better trade-off is obtained.

Impact of selection of reference frames. We study the impact of the selection of reference frames in Tab. 4.4. We start by using a single reference frame. There seems to be a trend that when increasing the distance between the reference frame and the target frame, better performance is obtained. The possible reason for this is that the far-away reference frame may contain richer and different context which complements the one of the target frame. When using more reference frames ($k_1 = 9, k_2 = 6, k_3 = 3$), the best performance is achieved. It is worthy noting that the reference frames combination of $k_1 = 3, k_2 = 2$, and $k_3 = 1$, performs similarly as the cases when using single reference frame, possibly due to the fact that the close reference frames don't give much new information for segmenting the target frame.

Ablation on static and motional contexts. In this experiment, we study the impact of static and motional contexts on performance. Different from previous methods, the proposed CFFM can learn both static and motional contexts in a unified model. When CFFM predicts the segmentation mask for the current frame, it uses three previous frames as reference frames. For this ablation study, we simulate a case where only static contexts can be used, by replicating the current frame three times and using them as the reference frames. In this way, only static contexts could be used since all the reference frames are the same as the current one. We denote this experiment as “baseline+static contexts". Following our setting in ablation studies, we use the backbone MiT-B1 and the VSPW val dataset. The comparisons are shown in Tab. 4.5.

Impact of CFFA and CFM Starting from SegFormer [15], we first add CFFA by using MLP to process the context tokens and then merge them with the target features. We obtain mIoU of 37.6. Then we add both CFFA and CFM on the baseline, which is our final model. The segmentation mIoU is 38.5. It shows that both CFFA and CFM are valuable for the proposed CFFM mechanism.

4.5 CONCLUSION

The video contexts include *static contexts* and *motional contexts*, both of which are essential for video semantic segmentation. Previous methods pay much attention to *motional contexts* but ignore the *static contexts*. To this end, this chapter proposes a Coarse-to-Fine Feature Mining (CFFM) technique to jointly learn a unified presentation of static and motional contexts, for precise and efficient VSS. CFFM contains two parts: coarse-to-fine feature assembling and cross-frame feature mining. The former summarizes contextual information with different granularity for different frames, according to their distance to the target frame. The latter efficiently mines the contexts from neighbouring frames to enhance the feature of the target frame. While adding limited computational resources, CFFM boosts segmentation performance in a clear margin on datasets with full or partial annotations.

5

MINING RELATIONS AMONG CROSS-FRAME AFFINITIES FOR VIDEO SEMANTIC SEGMENTATION

5.1 INTRODUCTION

Image semantic segmentation aims at classifying each pixel of the input image to one of the predefined class labels, which is one of the most fundamental tasks in visual intelligence. Deep neural networks have made tremendous progresses in this field [3, 4, 33–35, 37, 43, 44, 206, 225], benefiting from the availability of large-scale image datasets [11, 198, 199] for semantic segmentation. However, in real life, we usually confront more complex scenarios in which a series of successive video frames need to be segmented. Thus, it is desirable to explore video semantic segmentation (VSS) by exploiting the temporal information.

The core of VSS is how to leverage temporal information. Most of the existing VSS works rely on the optical flow to model the temporal information. Specifically, they first compute the optical flow [61] that is further used to warp the features from neighboring video frames for feature alignment [46, 47, 50, 51, 54, 56, 58]. Then, the warped features can be simply aggregated. Although workable in certain scenarios, those methods are still unsatisfactory because i) the optical flow is error-prone and thus the error could be accumulated; ii) directly warping features may yield inevitable loss on the spatial correlations [52, 57]. Hence, other approaches [59, 60] directly aggregate the temporal information in the feature level using attention techniques, as shown in Fig. 5.1. Since they are conceptually simple and avoid the problems incurred by optical flow, we follow this way to exploit temporal information. In general, those methods first calculate the attentions/affinities between the target and the references, which are then used to generate the refined features. Though promising, they only consider the single-scale attention. What's more, they do not mine the relations within the affinities.

In this chapter, we propose a novel approach MRCFA by Mining Relations among Cross-Frame Affinities for VSS. Specifically, we compute the **Cross-Frame Affinities (CFA)** between the features of the *target* frame and the *reference* frame. Hence, CFA is expected to have large activation for informative features and small activation for useless features. When aggregating the CFA-based temporal features, the informative features are highlighted and useless features are suppressed. As a result, the segmentation of the target frame would be improved by embedding temporal contexts. With the above analysis, the main focus of this chapter is mining relations among CFA to improve the representation capability of CFA. Since deep neural networks usually generate multi-scale features and CFA can be calculated at different scales, we can obtain multi-scale CFA accordingly. Intuitively, the relations among CFA are twofold: single-scale intrinsic correlations and multi-scale relations.

For the *single-scale intrinsic correlations*, each feature token in a reference frame (i. e., reference token) corresponds to a CFA map for the target frame. Intuitively, we have the observation that the CFA map of each reference token should be locally correlated as the feature map of the target frame is locally correlated, which is also the basis of CNNs. It is interesting to note that the traditional 2D convolution can be adopted to model such single-scale intrinsic correlations of CFA. Generally, convolution is used for processing features. In contrast, we use convolution to refine the affinities of features for improving the quality of affinities. We call this step Single-scale Affinity Refinement (SAR).

For the *multi-scale relations*, we propose to exploit the relations among multi-scale CFA maps. The CFA maps generated from high-level features have a small scale and a coarse representation, while the CFA maps generated from low-level features have a large scale and a fine representation. It is natural to aggregate multi-scale CFA maps using a high-to-low decoder structure so that the resulting CFA would contain both coarse and fine affinities. Generally, the decoder structure is usually used for fusing multi-scale features. In contrast, we build a decoder to aggregate the multi-scale affinities of features. We call this step Multi-scale Affinity Aggregation (MAA).

When we revisit the above MAA, one requirement arises: the reference tokens at different scales should have the same number and corresponding semantics; otherwise, it is impossible to connect a decoder. As discussed above, each reference token corresponds to a CFA map for the target frame. Only when two reference tokens have the same semantics, their CFA maps can be merged. For this goal, a simple solution is to downsample reference tokens at different scales into the same size. This also saves the computation due to the reduction of reference tokens. It inspires us to further reduce the computation by sampling reference tokens. To this end, we propose a **Selective Token Masking** strategy to select S most important reference

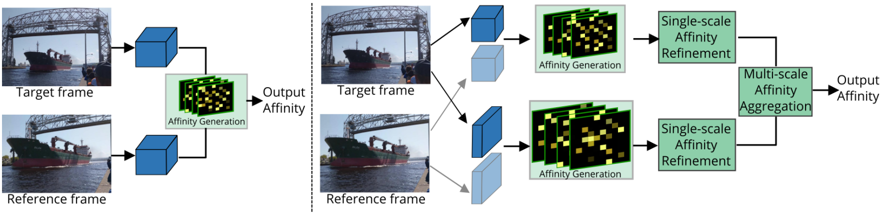


FIGURE 5.1: *Left*: recent VSS methods [59, 60] for which the affinity is directly forwarded to the next step (feature retrieval). The affinity is shown in a series of 2D maps. *Right*: We propose to mine the relations within the affinities before outputting the affinity, by Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA).

tokens and abandon less important ones. Then, the relation mining among CFA is executed based on the selected tokens.

In summary, there are three aspects for mining relations among CFA: i) We propose Single-scale Affinity Refinement for refining the affinities among features, based on single-scale intrinsic correlations; 2) We further introduce Multi-scale Affinity Aggregation by using an affinity decoder for aggregating the multi-scale affinities among features; 3) To make it feasible to execute MAA and improve efficiency, we propose Selective Token Masking (STM) to generate a subset of consistent reference tokens for each scale. After strengthened with single-scale and multi-scale relations, the final CFA can be directly used for embedding reference features into the target frame. Extensive experiments show the superiority of our method over previous VSS methods. Besides, our exploration of affinities among features would provide a new perspective on VSS.

5.2 RELATED WORKS

5.2.1 *Image Semantic Segmentation*

Image semantic segmentation has always been a hot topic in image understanding since it plays an important role in many real applications such as autonomous driving, robotic perception, augmented reality, aerial image analysis, and medical image analysis. In the era of deep learning, various algorithms have been proposed to improve semantic segmentation. Those related works can be divided into two groups: CNN-based methods [3, 42, 190, 206, 224, 245] and transformer-based methods [15, 233]. Among

CNN-based methods, FCN [206] is a pioneer work, which adopts fully convolutional networks and pixel-to-pixel classification. Since then, other methods [3, 4, 38, 43, 44, 155] have been proposed to increase the receptive fields or representation ability of the network. Another group of works [15, 233] is based on the transformer which is first proposed in natural language processing [116] and has the ability to capture global context [85]. Though tremendous progress has been achieved in image segmentation, researchers have paid more and more attention to VSS since video streams are a more realistic data modality.

5.2.2 *Video Semantic Segmentation*

Video semantic segmentation (VSS), aiming at classifying each pixel in each frame of a video into a predefined category, can be tackled by applying single image semantic segmentation algorithms [3, 4, 15, 39, 40] on each video frame. Though simple, this approach serves as an important baseline in VSS. One obvious drawback of this method is that the temporal information between consecutive frames is discarded and unexploited. Hence, dedicated VSS approaches [46–54, 57–59, 202, 226, 229, 244, 246] are proposed to make use of the temporal dimension to segment videos.

Most of the current VSS approaches can be divided into two groups. The first group of approaches focuses on using temporal information to reduce computation. Specifically, LLVS [57], Accel [54], GSVNET [58] and EVS [246] conserve computation by propagating the features from the key frames to non-key frames. Similarly, DVSNet [225] divides the current frame into different regions and the regions which do not differ much from previous frames do not traverse the slow segmentation network, but a fast flow network. However, due to the fact that they save computation on some frames or regions, their performance is usually inferior to the single frame baseline. The second group of methods focuses on exploring temporal information to improve segmentation performance and prediction consistency across frames. Specifically, NetWarp [244] wraps the features of the reference frames for temporal aggregation. TDNet [52] aggregates the features of sequential frames with an attention propagation module. ETC [51] uses motion information to impose temporal consistency among predictions between sequential frames. STT [60], LMANet [59] and CFFM [247] exploit the features from reference frames to help segment the target frame by the attention mechanism. Despite the promising results, those methods do

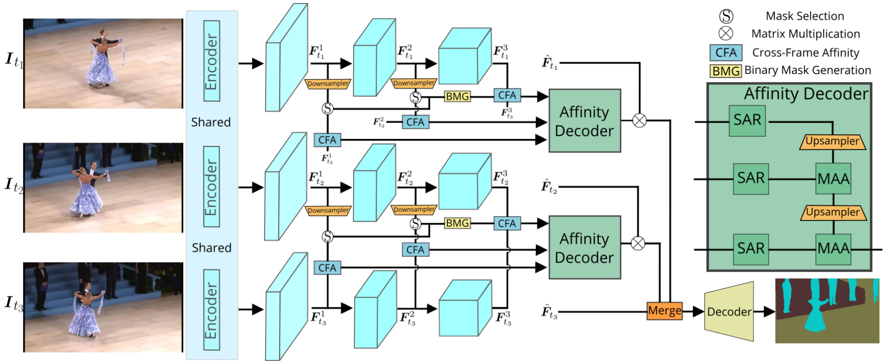


Figure 5.2: Network overview of MRCFA. Our method is illustrated when the clip contains three frames ($T = 3$). The first two frames are reference frames while the last one is the target frame. All frames first go through the encoder to extract the multi-scale features ($L = 3$) from the intermediate layers. For each reference frame, we compute the Cross-Frame Affinities (CFA) across different scales of features. To save computation, Selective Token Masking is proposed. Then, the multi-scale affinities are input to an affinity decoder to learn a unified and informative affinity, through the Single-scale Affinity Refinement (SAR) module and Multi-scale Affinity Aggregation (MAA). The new representation of the target frame using the reference is obtained by exploiting the refined affinity to retrieve the corresponding reference features. Finally, all the new representations of the target are merged to segment the target. *Best viewed in color.*

not consider correlation mining among cross-frame affinities. This chapter provides a new perspective on VSS by mining the relations among affinities.

5.3 METHODOLOGY

In this section, we target VSS and present a novel approach MRCFA through Mining Relations among Cross-Frame Affinities. The main idea of MRCFA is to mine the relations among multi-scale affinities computed from multi-scale intermediate features between the target frame and the reference frames, as illustrated in Fig. 5.2. We first provide the preliminaries in §5.3.1. Next, we introduce Single-scale Affinity Refinement (SAR) which independently refines each single-scale affinity in §5.3.2. After that, Multi-scare Affinities Aggregation (MAA) which merges affinities across various scales is presented in §5.3.3. Finally, we explain the Selective Token Masking mechanism (§5.3.4) to reduce the computation.

5.3.1 *Preliminaries*

Given a video clip $\{\boldsymbol{I}_{t_i} \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{T}$ containing T video frames and corresponding ground-truth masks $\{\boldsymbol{M}_{t_i} \in \mathbb{R}^{H\times W}\}_{i=1}^{T}$, our objective is to learn a VSS model. Without loss of generalizability, we focus on segmenting the last frame $\boldsymbol{I}_{t_T}$, which is referred as the target frame. All the previous frames $\{\boldsymbol{I}_{t_i}\}_{i=1}^{T-1}$ are referred as the reference frames. Each frame $\boldsymbol{I}_{t_i}$ is first input into an encoder to extract intermediate features $\{\boldsymbol{F}_{t_i}^{l} \in \mathbb{R}^{H_l W_l \times C_l}\}_{l=1}^{L}$ in various scales from L intermediate layers of the deep encoder, where H_l, W_l, C_l correspond to the height, width, number of channels of the feature map, respectively. For simplicity, multi-scale features $\{\boldsymbol{F}_{t_i}^{l}\}_{l=1}^{L}$ are in the order that shallow features are followed by deep features. We have $H_{l_1} \geq H_{l_2}$ and $W_{l_1} \geq W_{l_2}$, if $l_1 < l_2$. In this chapter, we aim to exploit the contextual information in the reference frames to refine the features of the target frame and thus improve the target's segmentation. Instead of simply modeling the affinities among frames for feature aggregation, we devote our efforts to mine relations among cross-frame affinities.

5.3.2 *Single-scale Affinity Refinement*

We start with introducing the process of generating multi-scale affinities between the target frame and each reference frame. We first map the features $\{\boldsymbol{F}_{t_T}^{l}\}_{l=1}^{L}$ of the target frames into the queries $\{\boldsymbol{Q}^{l}\}_{l=1}^{L}$ by a linear layer, as:

$$\boldsymbol{Q}^{l} = f(\boldsymbol{F}_{t_T}^{l}; \boldsymbol{W}_{query}^{l}), \tag{5.1}$$

where $\boldsymbol{W}_{query}^{l} \in \mathbb{R}^{C_l\times C_l}$ is the weight matrix of the linear layer f and $\boldsymbol{Q}^{l} \in \mathbb{R}^{H_l W_l\times C_l}$. Similarly, the multi-scale features $\{\boldsymbol{F}_{t_i}^{l}\}_{l=1}^{L}$ of the reference frame ($i \in [1, T-1]$) are also processed to generate the keys $\{\boldsymbol{K}_{t_i}^{l}\}_{l=1}^{L}$, as follows:

$$\boldsymbol{K}_{t_i}^{l} = f(\boldsymbol{F}_{t_i}^{l}; \boldsymbol{W}_{key}^{l}), \tag{5.2}$$

where $\boldsymbol{W}_{key}^{l} \in \mathbb{R}^{C_l\times C_l}$ is the corresponding weight matrix and $\boldsymbol{K}_{t_i}^{l} \in \mathbb{R}^{H_l W_l\times C_l}$. After obtaining the queries and the keys, we are ready to generate the affinities between the target frame $\boldsymbol{I}_{t_T}$ and each reference frame

$\boldsymbol{I}_{t_i}$ ($i \in [1, T-1]$) across all scales. Then, **Cross-Frame Affinities (CFA)** are computed as:

$$\boldsymbol{A}^l_{t_i} = \boldsymbol{Q}^l \times \boldsymbol{K}^{l\top}_{t_i}, \tag{5.3}$$

where we have $\boldsymbol{A}^l_{t_i} \in \mathbb{R}^{H_l W_l \times H_l W_l}$, $l \in [1, L]$ and $i \in [1, T-1]$. It means that, at each scale, the target frame has an affinity map with each reference frame.

Based on the affinities $\{\boldsymbol{A}^l_{t_i}\}^L_{l=1}$, our affinity decoder is designed to mine the correlations between them to learn a better affinity between the target and the reference frame. As shown in Fig. 5.2, it is comprised of two modules: Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). Please refer to §5.1 for our motivations. In order to reduce computation and prepare the affinities for MAA module which requires the same number and corresponding semantics (see §5.1), our affinity decoder operates on $\{\tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{H_l W_l \times S}\}^L_{l=1}$, rather than $\{\boldsymbol{A}^l_{t_i} \in \mathbb{R}^{H_l W_l \times H_l W_l}\}^L_{l=1}$. The affinities $\tilde{\boldsymbol{A}}^l_{t_i}$ is a downsampled version of $\boldsymbol{A}^l_{t_i}$ along the second dimension, which will be explained in §5.3.4.

Single-scale Affinity Refinement (SAR). For the affinity matrix $\tilde{\boldsymbol{A}}^l_{t_i}$, each of its elements corresponds to a similarity between a token in the query and a token in the key. We reshape $\tilde{\boldsymbol{A}}^l_{t_i}$ from $\mathbb{R}^{H_l W_l \times S}$ to $\mathbb{R}^{H_l \times W_l \times S}$. In order to learn the correlation within the single-scale affinity $\tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{H_l \times W_l \times S}$, a straightforward way is to exploit 3D convolution. However, this approach suffers from two weaknesses. First, it requires a large amount of computational cost. Second, not all the activations within the 3D window are meaningful. Considering a 3D convolution with a kernel $\mathcal{K} \in \mathbb{R}^{k \times k \times k}$, the normal 3D convolution at the location $x = (x_1, x_2, x_3)$ is formulated as:

$$(\tilde{\boldsymbol{A}}^l_{t_i} * \mathcal{K})_x = \sum_{(o_1, o_2, o_3) \in \mathcal{N}(x)} \tilde{\boldsymbol{A}}^l_{t_i}(o_1, o_2, o_3) \mathcal{K}(o_1 - x_1, o_2 - x_2, o_3 - x_3), \tag{5.4}$$

where $\mathcal{N}(x)$ is the set of locations in the 3D window ($k \times k \times k$) centered at x, and $|\mathcal{N}(x)| = k^3$. As seen in Eq. (5.4), all the neighbors along three dimensions are used to conduct the 3D convolution. However, the last dimension of $\tilde{\boldsymbol{A}}^l_{t_i}$ is the sparse selection in the key (§5.3.4) and thus does not contain spatial information. Including the neighbors along the last dimension could introduce noise and bring more complexity. Thus, we propose to refine the affinities across the first two dimension. For affinity $\tilde{\boldsymbol{A}}^l_{t_i}$ of each scale, we first permute it to $\mathbb{R}^{S \times H_l \times W_l}$ and then use 2D convolutions to learn the relations within the affinity. The refined affinity is denoted as $\bar{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{S \times H_l \times W_l}$. This process can be formulated as:

$$\begin{aligned} &\tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{H_l \times W_l \times S} \rightarrow \tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{S \times H_l \times W_l}, \\ &\bar{\boldsymbol{A}}^l_{t_i} = G(\tilde{\boldsymbol{A}}^l_{t_i}), \end{aligned} \tag{5.5}$$

where G represents a few connvolutional layers. Due to the use of 2D convolution and the token reduction mentioned in §5.3.4, the refinement of affinities is fast. After refining affinity for each scale, we collect the refined affinities $\{\bar{\boldsymbol{A}}^l_{t_i}\}_{l=1}^{L}$ for all scales. Next, we present Multi-scale Affinity Aggregation (MAA) module.

5.3.3 *Multi-scale Affinity Aggregation*

Multi-scale Affinity Aggregation (MAA). The affinity from the deep features contains more semantic but more coarse information, while the affinity from the shallow features contains more fine-grained but less semantic information. Thus, we propose a **Multi-scale Affinity Aggregation** module to aggregate the information from small-scale affinities to large-scale affinities, as:

$$\begin{aligned} \boldsymbol{B}^L_{t_i} &= \bar{\boldsymbol{A}}^L_{t_i}, \\ \boldsymbol{B}^l_{t_i} &= G(\Gamma(\boldsymbol{B}^{l+1}_{t_i}) + \bar{\boldsymbol{A}}^l_{t_i}), \quad l = L-1, ..., 1, \end{aligned} \tag{5.6}$$

where Γ denotes upsampling operation to match the spatial size when necessary. By Eq. (5.6), we generate the final refined affinity $\boldsymbol{B}^1_{t_i}$ between the target frame $\boldsymbol{I}_{t_T}$ and each reference frame $\boldsymbol{I}_{t_i}$ ($i \in [1, L-1]$).

Feature Retrieval. For single-frame semantic segmentation, SegFormer [15] generates the final feature $\hat{\boldsymbol{F}}_{t_i} \in \mathbb{R}^{\hat{H}\hat{W} \times \hat{C}}$ by merging multiple intermediate features. The final features are informative and directly used to predict the segmentation mask [15]. Using the refined affinity $\boldsymbol{B}^1_{t_i}$ and the informative features $\hat{\boldsymbol{F}}_{t_i}$, we compute the new refined feature representations for the target frame. Specifically, the feature $\hat{\boldsymbol{F}}_{t_i}$ is first downsampled to the size of $\mathbb{R}^{H_L W_L \times \hat{C}}$. To correspond the refined affinity and the informative feature, we sample feature $\hat{\boldsymbol{F}}_{t_i}$ using the token selection mask $\tilde{\boldsymbol{M}}_{t_i}$ (§5.3.4) and obtain $\tilde{\boldsymbol{F}}_{t_i} \in \mathbb{R}^{S \times \hat{C}}$. The new feature representation for the target frame using the reference is obtained as:

$$\begin{aligned} \boldsymbol{B}^1_{t_i} \in \mathbb{R}^{S \times H_1 \times W_1} &\rightarrow \boldsymbol{B}^1_{t_i} \in \mathbb{R}^{H_1 W_1 \times S}, \\ \boldsymbol{O}_{t_i} &= \boldsymbol{B}^1_{t_i} \times \tilde{\boldsymbol{F}}_{t_i}. \end{aligned} \tag{5.7}$$

Intuitively, this step is to retrieve the informative features from the reference frame to the target frame using affinity. Computing Eq. (5.7) for all reference frames, we obtain the new representations of the target frame as $\{\boldsymbol{O}_{t_i}\}_{i=0}^{T-1}$.

The final feature used to segment the target frame is merged from $\{\boldsymbol{O}_{t_i}\}_{i=0}^{T-1}$ and $\hat{\boldsymbol{F}}_{t_L}$ as follows:

$$\boldsymbol{O}_{t_L} = \frac{1}{T-1}\Gamma(\sum_{i=1}^{T-1}\boldsymbol{O}_{\boldsymbol{t_i}}) + \hat{\boldsymbol{F}}_{t_L}. \tag{5.8}$$

Finally, a simple MLP decoder projects $\boldsymbol{O}_{t_L}$ to the segmentation logits, and typical cross-entropy loss is used for training. In the test period, when segmenting the target frame I_{t_T}, the encoder only needs to generate the features for the current target while the reference frames are already processed in previous steps and the corresponding features can be directly used.

5.3.4 *Selective Token Masking*

As discussed in §5.1, there should be the same number of reference tokens with corresponding semantics across scales. Besides, computing cross-frame affinities requires a lot of computation. Thus, our affinity decoder does not process $\{\boldsymbol{A}_{t_i}^l \in \mathbb{R}^{H_l W_l \times H_l W_l}\}_{l=1}^{L}$, but rather its downsampled version $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^{L}$. Here, we explain how to generate $\{\tilde{\boldsymbol{A}}_{t_i}^l\}_{l=1}^{L}$, by reducing the number of tokens in the multi-scale keys $\{\boldsymbol{K}_{t_i}^l\}_{l=1}^{L}$ before computing Eq. (5.3).

We exploit convolutional layers to downsample the multi-scale keys to the spatial size of $H_L \times W_L$. Specifically, for the key $\boldsymbol{K}_{t_i}^l$ ($l \in [1, L-1]$), we process it by a convolutional layer with both kernel and stride size of $(\frac{H_l}{H_L}, \frac{W_l}{W_L})$. As a result, we obtain new keys $\hat{\boldsymbol{K}}_{t_i}^l$ with smaller spatial size, which is given by

$$\begin{aligned} \boldsymbol{K}_{t_i}^l \in \mathbb{R}^{H_l W_l \times C_l} &\rightarrow \boldsymbol{K}_{t_i}^l \in \mathbb{R}^{C_l \times H_l \times W_l}, \\ \hat{\boldsymbol{K}}_{t_i}^l &= g(\boldsymbol{K}_{t_i}^l; (\frac{H_l}{H_L}, \frac{W_l}{W_L}); (\frac{H_l}{H_L}, \frac{W_l}{W_L})), \\ \hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{C_l \times H_L \times W_L} &\rightarrow \hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{H_L W_L \times C_l}. \end{aligned} \tag{5.9}$$

where $g(\cdot; (k_h, k_w); (s_h, s_w))$ represents a convolutional layer with the kernel size (k_h, k_w) and the stride (s_h, s_w). After this step, we obtain the downsampled keys $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$, where $\hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{H_L W_L \times C_l}$, $l \in [1, L-1]$ and $i \in [1, T-1]$.

To further reduce the number of tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$, we propose to select important tokens and discard less important ones. The idea is to first compute the affinity for the deepest query/key pair ($\boldsymbol{Q}^L$ and $\boldsymbol{K}_{t_i}^L$), then generate a binary mask of important token locations, and finally select tokens in keys using the mask. The process of **Binary Mask Generation (BMG)** is in the following. The affinity between the deepest query and key is given by $\boldsymbol{A}_{t_i}^L \in \mathbb{R}^{H_L W_L \times H_L W_L}$, following Eq. (5.3). Next, we choose the top-n maximum elements across each column of $\boldsymbol{A}_{t_i}^L$, given by

$$\begin{aligned} \hat{\boldsymbol{A}}_{t_i}^L[:, j] &= \arg\max_n(\boldsymbol{A}_{t_i}^L[:, j]), \\ j &\in [1, H_L W_L], \end{aligned} \tag{5.10}$$

where $\arg\max_n$ means to take the top-n elements, and $\hat{\boldsymbol{A}}_{t_i}^L \in \mathbb{R}^{n \times H_L W_L}$. Then, we sum over the top-n elements and generate a token importance map $\boldsymbol{M}_{t_i}$ as

$$\boldsymbol{M}_{t_i} = \sum_{j=1}^{n} (\hat{\boldsymbol{A}}_{t_i}^L[j, :]), \tag{5.11}$$

in which we have $\boldsymbol{M}_{t_i} \in \mathbb{R}^{H_L W_L}$. We recover the spatial size of $\boldsymbol{M}_{t_i}$ by reshaping it to $\mathbb{R}^{H_L \times W_L}$. The token importance map $\boldsymbol{M}_{t_i}$ shows the importance level of every location in the key feature map. Since $\boldsymbol{M}_{t_i}$ is derived from the deepest/highest level of features, the token importance information it contains is semantic-oriented and can be shared in other shallow levels. We use it to sample the tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$. Specifically, we sample p percent of the locations with the top-p highest importance scores in $\boldsymbol{M}_{t_i}$, where p is referred as the token selection ratio. The binary token selection mask with p percent of the locations highlighted is denoted as $\tilde{\boldsymbol{M}}_{t_i}$. The location with the value 1 in $\tilde{\boldsymbol{M}}_{t_i}$ means the token importance is within the top-p percent and the corresponding token will be selected. The location with the value 0 in $\tilde{\boldsymbol{M}}_{t_i}$ means the token in that location is less important and will thus be discarded. The total number of locations with the value 1 in $\tilde{\boldsymbol{M}}_{t_i}$ is denoted by $S = pH_L W_L$.

Using mask $\tilde{\boldsymbol{M}}_{t_i}$, we select p percent of tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$. The keys after selection are denoted as $\{\tilde{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{S \times C_l}\}_{l=1}^{L-1}$. With $\boldsymbol{Q}^l$ and $\tilde{\boldsymbol{K}}_{t_i}^l$, we compute the affinities $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^{L-1}$ using Eq. (5.3). For $\boldsymbol{A}_{t_i}^L$, we also conduct sampling using $\tilde{\boldsymbol{M}}_{t_i}$ and obtain $\tilde{\boldsymbol{A}}_{t_i}^L \in \mathbb{R}^{H_L W_L \times S}$. Merging the affinities from all L scales gives final affinities of $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^{L}$. After computing the affinities for all reference frames, we have the downsampled affinities $\{\{\tilde{\boldsymbol{A}}_{t_i}^l\}_{l=1}^{L}\}_{i=1}^{T-1}$.

5.4 EXPERIMENTS

5.4.1 *Experimental Setup*

Datasets. Densely annotating video frames requires intensive manual labeling efforts. The widely used datasets for VSS are Cityscapes [198] and CamVid [248] datasets. However, these datasets only contain sparse annotations, which limits the exploration of temporal information. Fortunately, the Video Scene Parsing in the Wild (VSPW) dataset [202] is proposed to facilitate the progress of this field. It is currently the largest-scale VSS dataset with 198,244 training frames, 24,502 validation frames and 28,887 test frames. For each video, 15 frames per second are densely annotated for 124 categories. These aspects make VSPW the most challenging benchmark for VSS up till now. Hence, most of our experiments are conducted on VSPW. To further demonstrate the effectiveness of MRCFA, we also show results on Cityscapes, for which only one out of 30 frames is densely annotated.

Implementation details. For the encoder, we use the MiT backbones as in Segformer [15], which have been pretrained on ImageNet-1K [146]. For VSPW dataset, three reference frames are used, which are 9, 6 and 3 frames ahead of the target, following [202]. Three-scale features from the last three transformer blocks are used to compute the cross-frame affinities and mine their correlations. For the Mask-based Token Selection (MTS), we set p=80% for MiT-B0 and p=50% for other backbones unless otherwise specified. For training augmentations, we use random resizing, horizontal flipping, and photometric distortion to process the original images. Then, the images are randomly cropped to the size of 480×480 to train the network. We set the batch size as 8 during training. The models are all trained with AdamW optimizer for a maximum of 160k iterations and "poly" learning rate schedule. The initial learning rate is 6e-5. For simplicity, we perform the single-scale test on the whole image, rather than the sliding window test or multi-scale test. The input images are resized to 480×853 for VSPW. We also do not perform any post-processing such as CRF [193]. For Cityscape, the input image is cropped to 512×1024 during training and resized to the same resolution during inference. And we use two reference frames and four-scale features. The number of frames being processed per second (FPS) is computed on a single Quadro RTX 6000 GPU (24G memory).

Evaluation metrics. To evaluate the segmentation results, we adopt the commonly used metrics of Mean IoU (mIoU) and Weighted IoU (WIoU),

Methods	T	t_1	t_2	t_3	mIoU ↑	mVC_8 ↑	mVC_{16} ↑
SegFormer [15]	-	-	-	-	36.5	84.7	79.9
MRCFA (Ours)	2	-1	-	-	38.0	85.9	81.2
	2	-3	-	-	38.1	85.5	80.7
	2	-6	-	-	38.2	85.1	80.3
	2	-9	-	-	37.4	85.5	81.2
	3	-6	-3	-	38.4	87.0	82.1
	3	-9	-6	-	38.4	86.9	82.0
	4	-9	-6	-3	**38.9**	**88.8**	**84.4**

Table 5.1: The impact of the selection of reference frames. The best results are shown in **bold**.

p	mIoU ↑	mVC_8 ↑	mVC_{16} ↑	Memory (M) ↓	FPS (f/s) ↑
100%	39.4	89.2	84.9	1068	32.9
90%	39.1	89.1	84.8	1035	34.2
70%	39.1	88.2	83.9	969	36.8
50%	38.9	88.8	84.4	903 (15.4%)	40.1 (21.9%)
30%	38.5	86.7	81.9	838	43.5
10%	35.9	86.2	81.7	773	47.2

Table 5.2: The impact of token selection ratio p. The row which best deals with the trade-off between performance and computation resources is shown in red.

following [206]. We also use Video Consistency (VC) [202] to evaluate the category consistency among the adjacent frames in the video, following [202]. Formally, video consistency VC_n for n consecutive frames for a video clips $\{\boldsymbol{I}_c\}_{c=1}^{C}$, is computed by: $\text{VC}_n = \frac{1}{C-n+1}\sum_{i=1}^{C-n+1}\frac{(\cap_i^{i+n-1}\boldsymbol{S}_i)\cap(\cap_i^{i+n-1}\boldsymbol{S}_i^{'})}{\cap_i^{i+n-1}\boldsymbol{S}_i}$, where $C \geq n$. $\boldsymbol{S}_i$ and $\boldsymbol{S}_i^{'}$ are the ground-truth mask and predicted mask for i^{th} frame, respectively. We compute the mean of video consistency VC_n for all videos in the dataset as mVC_n. Following [202], we compute mVC_8 and mVC_{16} to evaluate the visual consistency of the predicted masks. Please refer to [202] for more details about VC.

5.4.2 *Ablation Studies*

We conduct ablation studies on the large-scale VSPW dataset [202] to validate the key designs of MRCFA. For fairness, we adopt the same settings as in §5.4.1 unless otherwise specified. The ablation studies are conducted on MiT-B1 backbone.

Influence of the reference frames. We study the performance of our method with respect to different choices of reference frames in Tab. 5.1. We have the following observations. First, using a single reference frame largely improves the segmentation performance (mIoU). For example, when using a single reference frame which is 3 frames ahead of the target one, the mIoU improvement over the baseline (SegFormer) is 1.6%, i.e., 38.1 over 36.5. Further adding more reference frames, better segmentation performance is observed. The best mIoU of 38.9 is obtained when using reference frames of 9, 6, and 3 frames ahead of the target. Second, for the prediction consistency metrics (mVC$_8$ and mVC$_{16}$), the advantage of exploiting more reference frames is more obvious. For example, using one reference frame ($t_1 = -6$) gives mVC$_8$ and mVC$_{16}$ of 85.1 and 80.3, improving the baseline by 0.4% and 0.4%, respectively. However, when using three reference frames ($t_1 = -9$, $t_2 = -6$, $t_3 = -3$), the achieved mVC$_8$ and mVC$_{16}$ are much more superior to the baseline, improving by 4.1% and 4.5%. The results are reasonable because using more reference frames gives the model a bigger view of the previously predicted features and thus generates more consistent predictions.

Influence of token selection ratio p. We study the influence of the token selection ratio p in terms of performance and computational resources in Tab. 5.2. Smaller p represents that less number of tokens in the key features are selected and thus less computation resource is required. Hence, there is a trade-off between the segmentation performance and the required resources (GPU memory and additional latency). In the experiments, when reducing $p = 100\%$ to 50%, the performance reduces slightly (0.5 in mIoU) while the GPU memory reduces by 15.4% and FPS increases by 21.9%. When further reducing p to 10%, the performance largely decreases in terms of mIoU, mVC$_8$ and mVC$_{16}$. The reason is that too many tokens are discarded in the reference frames and the remained tokens are not informative enough to provide the required contexts for segmenting the target frame. To sum up, the best trade-off is achieved when $p = 50\%$.

Influence of the feature scales. For VSPW dataset, we use three-scale features output from the last three transformer blocks. Here, we conduct an

L	mIoU ↑	mVC$_8$ ↑	mVC$_{16}$ ↑	Params (M) ↓	FPS (f/s) ↑
1	37.5	87.7	83.1	14.8	44.3
2	38.1	87.5	82.5	15.3	43.8
3	**38.9**	**88.8**	**84.4**	16.2	40.1

Table 5.3: Ablation study on the number of feature scales (L). Using more scales of features for our method progressively increases the performance.

Methods	SAR	MAA	mIoU ↑	mVC$_8$ ↑	mVC$_{16}$ ↑	Params (M) ↓
SegFormer	-	-	36.5	84.7	79.9	13.8
Feature Pyramid	-	-	37.8	87.0	82.0	16.2
Affinity Decoder	✓	✗	37.8	87.1	82.6	16.2
	✗	✓	37.4	88.3	83.6	16.2
	✓	✓	**38.9**	**88.8**	**84.4**	16.2

Table 5.4: Ablation study on the affinity decoder. Within our design, SAR and MAA are essential parts which contribute to the refinement of the affinity.

ablation study on the impact of the used feature scales. The results are shown in Tab. 5.3. It can be observed that using the features from the last stage ($L = 1$) or the last two stages ($L = 2$) gives inferior performance while consuming less computational resources and achieving faster running speed. When using three-scale features, the best results are achieved in terms of mIoU, mVC$_8$, and mVC$_{16}$. This is due to the fact that the features in different scales contain complementary information, and the proposed affinity decoder successfully mines this information through learning correlations between multi-scale affinities.

Binary mask generation. In **Selective Token Masking**, the number of tokens in $\{\hat{\boldsymbol{K}}_{t_i}^{l}\}_{l=1}^{L-1}$ is reduced by selecting important tokens and discarding less important ones. We compute the importance of the tokens in the key using the affinity $\boldsymbol{A}_{t_i}^{L} \in \mathbb{R}^{H_L W_L \times H_L W_L}$ from the deepest features. The first dimension of $\boldsymbol{A}_{t_i}^{L}$ corresponds to the query while its second dimension corresponds to the key. The importance of each token in the key is determined by its top-n similarities with the query tokens. If $n = 1$, then the importance of a key token is only related to the maximum similarity between that token and the query tokens. An alternative is to decide the importance of a key

Means	n	mIoU ↑	mVC_8 ↑	mVC_16 ↑
Top-n	1	37.8	87.6	83.0
	5	38.9	88.8	**84.4**
	10	**39.3**	88.7	84.2
	50	39.2	**88.9**	**84.4**
	100	38.9	88.0	83.5
Average	-	38.7	87.1	82.2

TABLE 5.5: Ablation study on the means of **Binary Mask Generation (BMG)**.

token by taking an average of the similarities between that token and the query tokens, instead of using top-n similarities. However, this design can be largely affected by class imbalance. For example, if most tokens in the query belongs to the same class, then it is likely that the key tokens which have the same class labels will be selected as important tokens and tokens of other classes may be discarded.

We compare "top-n selection" with "average selection" in Tab. 5.5. It shows that "top-n selection" (for most choices of n) is better than the "average selection", which supports our analysis. We also conduct an ablation study on the number of maximum similarities being chosen. It shows that best mIoU of 39.3 is obtained when using $n = 10$, and good performance is achieved when setting n to be a reasonable number.

Methods	Backbone	mIoU ↑	Weighted IoU ↑	mVC_8 ↑	mVC_{16} ↑	Params (M) ↓	FPS (f/s) ↑
SegFormer [15]	MiT-B0	32.9	56.8	82.7	77.3	3.8	73.4
SegFormer [15]	MiT-B1	36.5	58.8	84.7	79.9	13.8	58.7
MRCFA (Ours)	MiT-B0	35.2	57.9	88.0	83.2	5.2	50.0
MRCFA (Ours)	MiT-B1	**38.9**	**60.0**	**88.8**	**84.4**	16.2	40.1
DeepLabv3+ [39]	ResNet-101	34.7	58.8	83.2	78.2	62.7	-
UperNet [244]	ResNet-101	36.5	58.6	82.6	76.1	83.2	-
PSPNet [4]	ResNet-101	36.5	58.1	84.2	79.6	70.5	13.9
OCRNet [35]	ResNet-101	36.7	59.2	84.0	79.0	58.1	14.3
ETC [51]	PSPNet	36.6	58.3	84.1	79.2	89.4	-
NetWarp [244]	PSPNet	37.0	57.9	84.4	79.4	89.4	-
ETC [51]	OCRNet	37.5	59.1	84.1	79.1	58.1	-
NetWarp [244]	OCRNet	37.5	58.9	84.0	79.0	58.1	-
$\mathrm{TCB}_{\text{st-ppm}}$ [202]	ResNet-101	37.5	58.6	87.0	82.1	70.5	10.0
$\mathrm{TCB}_{\text{st-ocr}}$ [202]	ResNet-101	37.4	59.3	86.9	82.0	58.1	5.5
$\mathrm{TCB}_{\text{st-ocr-mem}}$ [202]	ResNet-101	37.8	59.5	87.9	84.0	58.1	5.5
SegFormer [15]	MiT-B2	43.9	63.7	86.0	81.2	24.8	39.2
SegFormer [15]	MiT-B5	48.2	65.1	87.8	83.7	82.1	17.2
MRCFA (Ours)	MiT-B2	45.3	64.7	90.3	86.2	27.3	32.1
MRCFA (Ours)	MiT-B5	**49.9**	**66.0**	**90.9**	**87.4**	84.5	15.7

Table 5.6: State-of-the-art comparison on the VSPW [202] validation set. MRCFA outperforms the compared methods on both accuracy (mIoU) and prediction consistency.

Ablation study on affinity decoder. We conduct ablation studies on the proposed affinity decoder. The results are shown in Tab. 5.4. Our affinity decoder processes the multi-scale affinities and generates a refined affinity matrix for each pair of the target and reference frames. It is reasonable to ask whether this design is better than the feature pyramid baseline. For this baseline (Feature Pyramid), we first compute the features for the target frame using the reference frame features at each scale and then merge those multi-scale features. For fair comparisons, we use a similar number of parameters for this baseline and other settings are also the same as ours. The result shows that while Feature Pyramid performs favorably over the single-frame baseline, our approach clearly surpasses it. It validates the effectiveness of the proposed affinity decoder.

As presented in §5.3.2, our affinity decoder has two modules: Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). The ablation study of two modules is provided in Tab. 5.4. Only using SAR, our method obtains the mIoU of 37.8, while only using MAA gives the mIoU of 37.4. Both variants are clearly better than the baseline, validating their effectiveness. Combining both modules, the proposed approach achieves the best mIoU, mVC_8, and mVC_{16}. It shows that both SAR and MAA are essential parts of the affinity decoder to learn better affinities to help segment the target frame.

5.4.3 *Segmentation Results*

The state-of-the-art comparisons on VSPW [202] dataset are shown in Tab. 5.6. Besides segmentation performance and visual consistency of the predicted masks, we also report the model complexity and FPS. According to the model size, the methods are divided into two groups: small models and large models. Among all methods, our MRCFA achieves state-of-the-art performance and produces the most consistent segmentation masks across video frames. For small models, our method on MiT-B1 clearly outperforms the strong baseline SegFormer [15] by 2.4% in mIoU and 1.2% in weighted IoU. In terms of the visual consistency in the predicted masks, our approach is superior to other methods, surpassing the second best method with 4.1% and 4.5% in mVC_8 and mVC_{16}, respectively. For large models, MRCFA shows similar behavior. The results indicate that our method is effective in mining the relations between the target and reference frames through the designed modules: SAR and MAA.

Methods	Backbone	mIoU↑	Params (M) ↓	FPS (f/s) ↑
FCN [206]	MobileNetV2	61.5	9.8	14.2
CC [53]	VGG-16	67.7	-	16.5
DFF [56]	ResNet-101	68.7	-	9.7
GRFP [47]	ResNet-101	69.4	-	3.2
PSPNet [4]	MobileNetV2	70.2	13.7	11.2
DVSN [50]	ResNet-101	70.3	-	19.8
Accel [54]	ResNet-101	72.1	-	3.6
ETC [51]	ResNet-18	71.1	13.2	9.5
SegFormer [15]	MiT-B0	71.9	3.7	58.5
MRCFA (Ours)	MiT-B0	72.8	4.2	33.3
SegFormer [15]	MiT-B1	74.1	13.8	46.8
MRCFA (Ours)	MiT-B1	**75.1**	14.9	21.5

TABLE 5.7: State-of-the-art comparison on the Cityscapes [198] val set.

Despite that our approach achieves impressive performance, it adds limited model complexity and latency. Specifically, compared to SegFormer (MiT-B2), MRCFA slightly increases the number of parameters from 24.8M to 27.3M and reduces the FPS from 39.2 to 32.1. The efficiency of our method benefits from the proposed STM mechanism for which we abandon unimportant tokens.

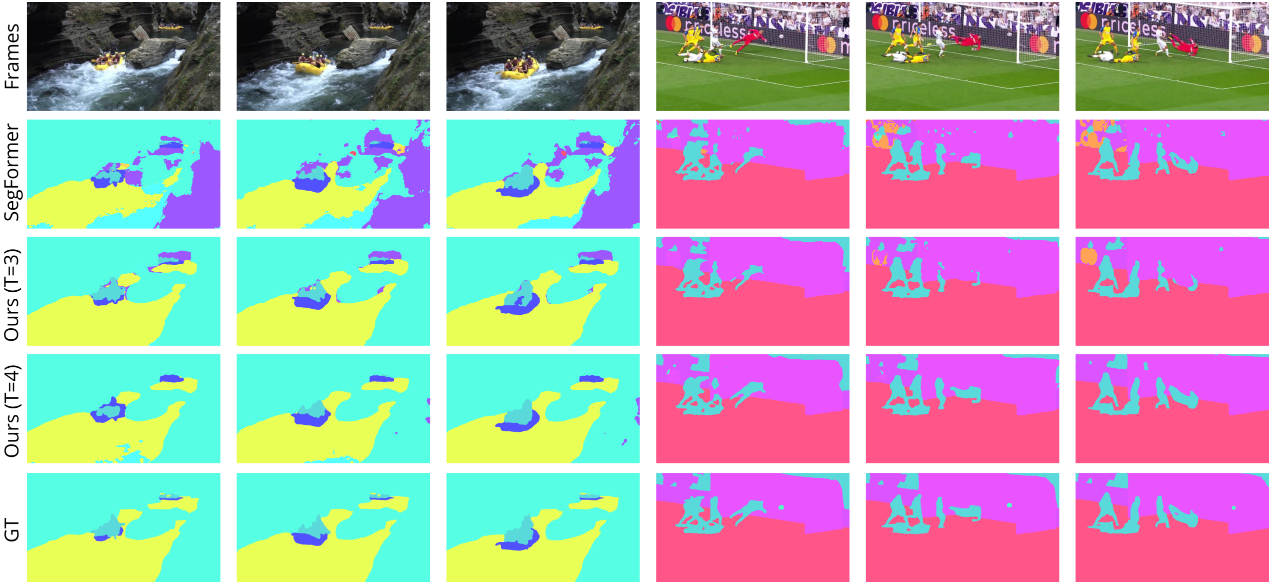


FIGURE 5.3: Qualitative results. From *top* to *bottom*: the input frames, the predicted masks of SegFormer [15], the predictions of ours ($T = 3, t_1 = -3, t_2 = -6$), the predictions of ours ($T = 4, t_1 = -3, t_2 = -6, t_3 = -9$) and the ground-truth masks. Our model generates better results than the baseline in terms of accuracy and VC.

Classes	wall	ceiling	door	stair	ladder	escalator	Playground slide	handrail fence	window	rail
IoU	**59.32**	**62.82**	**12.89**	**3.35**	**3.84**	**67.34**	**65.14**	35.64	34.36	**69.73**
Acc.	86.86	70.05	15.84	4.09	3.87	71.62	68.86	48.12	46.54	83.85
Classes	goal	pillar	pole	floor	ground	grass	sand	athletic field	road	path
IoU	**53.56**	27.05	**14.58**	**71.64**	**53.95**	**75.40**	47.18	**82.54**	**50.5**	28.07
Acc.	78.62	34.18	25.10	88.82	82.29	92.00	49.30	85.97	57.39	33.37
Classes	crosswalk	building	house	bridge	tower	windmill	well well lid	other construction	sky	mountain
IoU	38.64	40.29	**60.25**	**78.43**	28.76	**15.58**	**63.15**	32.20	**94.67**	**55.62**
Acc.	42.09	48.20	75.81	91.34	47.56	15.77	71.90	50.59	98.36	75.97
Classes	stone	wood	ice	snowfield	grandstand	sea	river	lake	waterfall	water
IoU	36.87	24.40	**78.97**	47.23	**5.00**	**67.27**	38.51	**19.23**	**51.65**	23.62
Acc.	49.67	37.21	88.34	48.14	5.05	86.88	51.71	31.46	61.98	31.72
Classes	billboard Bulletin Board	sculpture	pipeline	flag	parasol umbrella	cushion carpet	tent	roadblock	car	bus
IoU	36.11	**4.26**	**2.73**	**2.85**	**11.81**	25.91	**78.16**	**54.43**	**60.90**	**72.89**
Acc.	47.51	4.62	3.56	3.31	15.15	30.21	82.39	62.46	80.17	83.7
Classes	truck	bicycle	motorcycle	wheeled machine	ship boat	raft	airplane	tyre	traffic light	lamp
IoU	**14.13**	32.18	**6.09**	38.67	34.68	46.28	**81.12**	**51.41**	24.43	26.44
Acc.	17.82	41.35	6.11	63.87	83.47	49.73	92.74	52.13	25.45	48.74
Classes	person	cat	dog	horse	cattle	other animal	tree	flower	other plant	toy
IoU	**79.20**	**51.83**	32.20	43.75	20.82	43.48	**73.77**	**17.69**	41.87	35.06
Acc.	96.77	65.49	37.90	62.28	21.65	46.66	84.96	18.55	58.24	44.77
Classes	ball net	backboard	skateboard	bat	ball	cupboard showcase storage rack	box	traveling case trolley case	basket	bag package
IoU	37.49	**58.73**	**14.17**	34.62	**74.03**	40.98	**7.57**	**61.92**	**0.00**	**8.54**
Acc.	46.63	71.09	17.52	40.13	96.70	54.81	14.03	77.58	0.00	11.03
Classes	trash can	cage	plate	tub bowl pot	bottle cup	barrel	fishbowl	bed	pillow	table desk
IoU	**9.59**	48.52	25.61	**5.14**	29.78	48.13	**54.31**	**19.88**	**14.37**	47.72
Acc.	12.73	48.80	37.19	5.52	43.89	55.53	84.78	23.53	20.92	63.62
Classes	chair seat	bench	sofa	shelf	bathtub	gun	commode	roaster	other machine	refrigerator
IoU	33.21	**0.00**	**53.14**	**8.46**	28.39	25.96	**65.58**	29.03	**56.76**	**76.33**
Acc.	48.09	0.01	65.73	9.81	33.24	29.85	93.57	36.30	70.60	86.40
Classes	washing machine	Microwave oven	fan	curtain	textiles	clothes	painting poster	mirror	flower pot vase	clock
IoU	**58.01**	37.50	34.70	33.89	48.25	**71.18**	39.86	**1.80**	**17.58**	29.00
Acc.	69.20	44.33	40.42	45.54	52.69	74.61	53.70	1.82	25.52	29.10
Classes	book	tool	blackboard	tissue	screen television	computer	printer	mobile phone	keyboard	other electronic product
IoU	**13.01**	**5.79**	**0.00**	**16.09**	**59.86**	**12.69**	**16.68**	**53.16**	**74.51**	**6.29**
Acc.	13.86	9.83	0.00	17.57	76.80	25.42	17.20	60.87	79.53	9.89
Classes	fruit	food	instrument	train	-	-	-	-	-	-
IoU	**91.78**	**2.03**	**73.14**	48.92	-	-	-	-	-	-
Acc.	99.14	16.07	90.87	59.86	-	-	-	-	-	-

TABLE 5.8: Per-class results. The good performance (IoU $>$ 50%) is shown in **red** while unfavorable performance (IoU $<$ 20%) is shown in **blue**. We can observe that there is a big gap among classes in terms of mIoU and accuracy: some classes have very good performance while the results for some other classes are not good. This may inspire future research in VSS.

We conduct additional experiments on the semi-supervised Cityscapes [198] dataset, for which only one frame in each video clip is pixel-wise annotated. Tab. 5.7 shows the results. Similar to VSPW, MRCFA also achieves state-of-the-art results among the compared approaches under the semi-supervised setting and has a fast running speed. Besides the quantitative comparisons analyzed above, we also qualitatively compare the proposed method with the baseline on the sampled video clips in Fig. 5.3. For the two samples, our method generates more accurate segmentation masks, which are also more visually consistent.

Methods	# of layers	mIoU	mVC_8	mVC_{16}	Params (M)
VSwin [249]	1	37.3	86.4	81.3	14.6
VSwin [249]	2	37.1	86.9	82.0	15.4
VSwin [249]	3	36.7	87.6	82.6	16.2
MRCFA (Ours)	-	**38.9**	**88.8**	**84.4**	16.2

TABLE 5.9: Comparison with the Video Swin Transformer [249] (VSwin). For VSwin, we experiment different number of transformer layers. It shows that the proposed method clearly outperforms VSwin by a large margin.

Comparison with video swin transformer. We compare the proposed method with Video Swin Transformer [249] (VSwin), which is the extension of Swin Transformer [16] in videos. Since VSwin is not designed for video semantic segmentation, we explain how to adopt VSwin in this task. As introduced in the method, we extract informative features $\hat{\boldsymbol{F}}_{t_i} \in \mathbb{R}^{\hat{H}\hat{W}\times\hat{C}}$ for each frame in the video clip $\{\boldsymbol{I}_{t_i}\}_{i=1}^{T}$. All the features are concatenated as $\hat{\boldsymbol{F}} \in \mathbb{R}^{T\times\hat{H}\hat{W}\times\hat{C}}$. To prepare the feature for VSwin method, we process it as follows:

$$\begin{aligned}\hat{\boldsymbol{F}} \in \mathbb{R}^{T\times\hat{H}\hat{W}\times\hat{C}} &\rightarrow \hat{\boldsymbol{F}} \in \mathbb{R}^{(\frac{T}{p}\times p)\times(\frac{\hat{H}}{m}\times m)\times(\frac{\hat{W}}{m}\times m)\times\hat{C}} \\ &\rightarrow \hat{\boldsymbol{F}} \in \mathbb{R}^{(\frac{T}{p}\times\frac{\hat{H}}{m}\times\frac{\hat{W}}{m})\times(p\times m\times m)\times\hat{C}},\end{aligned} \tag{5.12}$$

where $p \times m \times m$ is the shape of the 3D window patch. The multi-head self-attention is computed within each window patch instead of the whole feature map, which is the core design of VSwin. After several video swin tranformer layers, the obtained features are used for predicting segmentation masks. The results are shown in Tab. 5.9. We tune the number of transformer layers for VSwin. Our approach largely surpasses the VSwin. Specifically, MRCFA has an advantage of 1.8%, 1.9%, and 2.4% over VSwin using two layers in terms of mIoU, mVC_8, and mVC_{16}, respectively. The possible reasons are in two-fold. First, the global interactions are only established within the 3D window patch while our method models the global connections between features of the target frame and all reference frames. Second, VSwin is mainly designed for video classification and thus may have difficulty in mining the contextual information among the frames for semantic segmentation.

Per-class Result. For the new VSPW [202] dataset, it is also interesting to examine the performance of each class to have a sense of easy or difficult classes. We show the per-class segmentation results in Tab. 5.8. As can be observed, different classes may have very different results. Some classes (e. g., *sky*, *person*, *fruit*, *ice*, *clothes*, *airplane*, *tent*) have very good results while some other categories (e. g., *basket*, *bench*, *pipeline*, *mirror*) have poor performance. There are several possible reasons for this: (1) the class imbalance problem existing in the VSPW dataset; (2) the difficulty to differentiate similar classes; and (3) the difficulty to segment small objects/stuff. We believe that the observation can inspire future work to design specific techniques to improve performance on those difficult classes.

5.4.4 *Additional Visual Results*

We provide more qualitative results for *indoor* scenes in Fig. 5.4. For all examples, our model generates better predictions in terms of segmentation accuracy and mask consistency across frames. We also observe that when using more reference frames by changing $T = 3$ to $T = 4$, better segmentation performance is obtained. Those visual results further demonstrate the effectiveness of the proposed modules including Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA) in aggregating the temporal information to help segment the target frame. The underlying reason for MRCFA's superiority lies in the mining of hyper relations among cross-frame attentions.

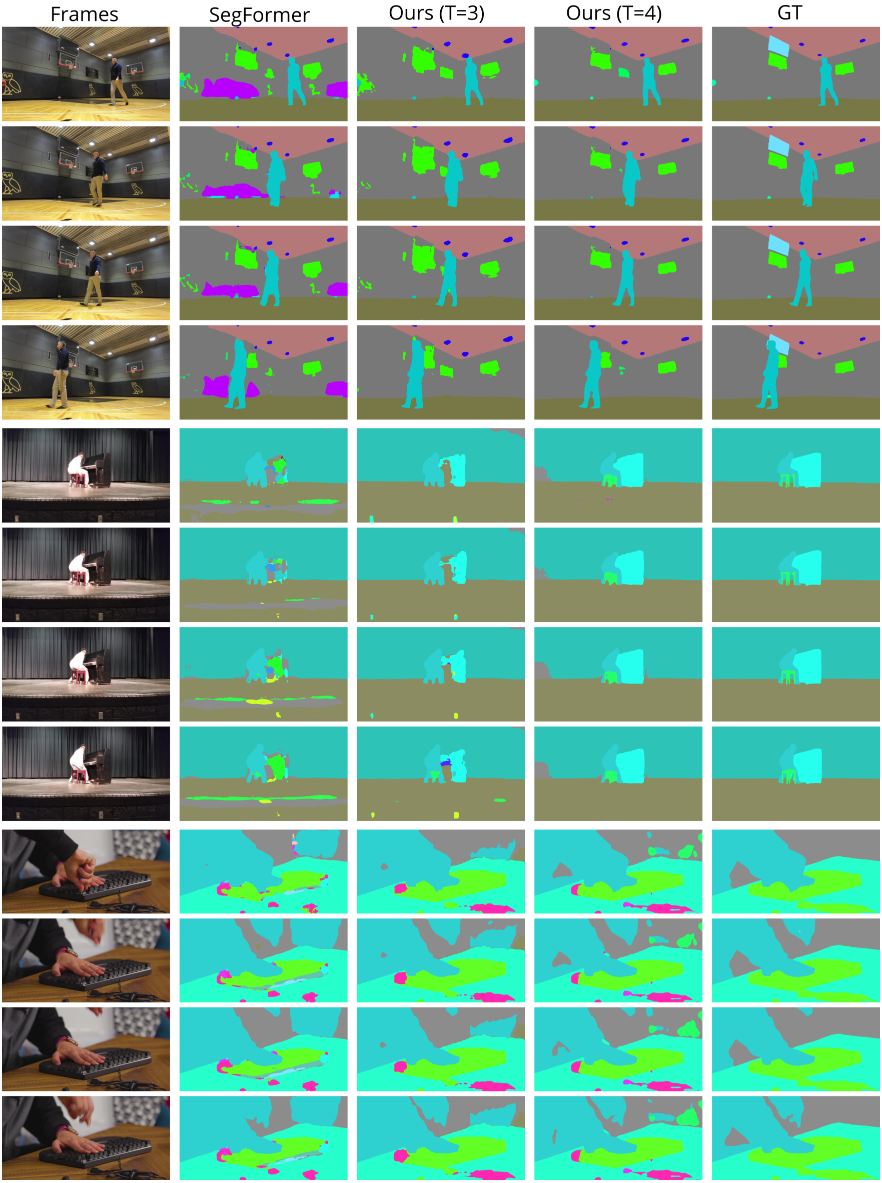


FIGURE 5.4: Additional qualitative results for *indoor scenes*. From *left* to *right*: the input frames, the predicted masks of SegFormer [15], the predictions of ours (T=3, $t_1 = -3$, $t_2 = -6$), the predictions of ours (T=4, $t_1 = -3$, $t_2 = -6$, $t_3 = -9$) and the ground-truth masks. *Best viewed in color*.

5.5 CONCLUSION

This chapter presents a novel framework MRCFA for semantic segmentation under dynamic scenes. Different from previous methods, we aim at mining the relations among multi-scale Cross-Frame Affinities (CFA) in two aspects: single-scale intrinsic correlations and multi-scale relations. Accordingly, Single-scale Affinity Refinement (SAR) is proposed to independently refine the affinity of each scale, while Multi-scale Affinity Aggregation (MAA) is designed to merge the refined affinities across various scales. To reduce computation and facilitate MAA, Selective Token Masking (STM) is adopted to sample important tokens in keys for the reference frames. Combining all the novelties, MRCFA generates better affinity relations between the target and the reference frames without largely adding computational resources. Extensive experiments on large-scale public datasets (VSPW and Cityscapes) demonstrate the effectiveness and efficiency of MRCFA, by setting new state-of-the-art performance and keeping low latency. The key components are validated to be essential for our method by extensive ablation studies. Overall, our exploration of mining the relations among cross-frame affinities could provide a new perspective for this task.

6

CONCLUSION AND OUTLOOK

6.1 CONTRIBUTIONS

In this thesis, we explore dense prediction problem under challenging scenarios using advanced attention mechanisms: **(1)** learning under camouflaged scenes using *single-image attention*; **(2)** learning from weak supervisions using *cross-image attention*; **(3)** densely understanding dynamic scenes using *video attention*. The studies on three challenging situations and advanced attention mechanisms are crucial for further promoting the development of dense prediction task and deepening our understanding of attention techniques. This thesis is divided into three parts, each focusing on one challenging situation.

In Chapter 2, we present a systematic study of object counting under camouflaged/indiscernible scenes where the objects are blended with respect to their surroundings. A new task, indiscernible object counting (IOC), is proposed. Due to the unavailability of large-scale dataset for IOC, we present a high-quality dataset with point annotations. For benchmarking purposes, we selected a number of mainstream methods and evaluated them on our dataset. We found that the top-performing methods for conventional crowd counting do not necessarily maintain their superiority for IOC. Therefore, we propose IOCFormer which combines global and local attention in a unified approach. The strengthened features after the attention modules help distinguish the foreground from the background. Experiments show IOCFormer achieves state-of-the-art performance on the challenging dataset. Our study further advances the frontier of dense prediction under camouflaged scenes.

In Chapter 3, we discuss semantic segmentation with image-level labels. The problem is ill-posed and challenging, since only image-level labels are available for training a model which can generate per-pixel predictions. To make full use of weak annotations, we propose a siamese network which takes a pair of images as input. Two advanced attention modules, co-attention and contrastive co-attention, are introduced to mine the semantic similarities and differences between two images. By conducting two auxiliary tasks, our model generates better object localization maps which identify more complete object regions. The generated pseudo ground-truth masks help train a better semantic segmentation model. As a result, the proposed approach achieves state-of-the-art performance under various settings and wins the first prize in the challenge. Our study represents a significant contribution to dense prediction under weak annotations.

In Chapter 4, we explore video semantic segmentation, which aims to densely segment the dynamic and complex scenes into different categories. We analyze the static and motional contexts in videos and propose a novel approach, CFFM, which learns both contexts in a unified framework. To incorporate the information from the mined contexts, we use non-self attention to refine the features of the target frame which is the focus of our method. The refined feature maps exploit more contextual information from prevision video frames, thus generating better mask predictions. Experiments show that CFFM achieves promising results on existing large-scale datasets in terms of segmentation accuracy and temporal consistency among predicted masks.

In Chapter 5, we study video semantic segmentation from a new perspective. While CFFM and previous methods directly use the affinity between the target and the contexts. We analyze that there exists useful local information within the affinity map and an affinity value in a specific location has relations with the affinity values in its neighbors. However, this information is discarded for those models. Therefore, we propose a new method, MRCFA, which further mines the relation among cross-frame affinities. In addition, MRCFA exploits multi-scale features to generate multi-scale affinities which include more information. The refined affinity maps help generate better feature maps for the target frame and thus yield better predictions. Experiments show that MRCFA achieves state-of-the-art performance, without significantly increasing processing latency due to efficient module designs. Our study finds that mining hyper-relation among attention maps benefits video semantic segmentation. The two methods proposed in Chapter 4 and Chapter 5 contribute to the development of dense prediction for dynamic and complex scenes (videos).

In the era of deep learning, dense prediction task for general and common settings has been significantly improved. However, the challenging scenarios are far from being solved and hinder the usage of algorithms in real-life applications. To bridge this gap, this thesis specifically studies dense prediction tasks under these difficult situations. For each case, we analyze the difficulties/challenges, and propose effective algorithms to address them. From a technical perspective, the proposed approaches focus on using attention mechanisms, including single-image attention, cross-image attention, and video attention, to refine per-pixel feature representation. The refined feature maps contain more accurate semantic information, thus generating better density maps for object counting and segmentation masks for semantic segmentation. In conclusion, this thesis contributes to further advancing dense prediction tasks and pave the way for real applications such as autonomous driving, AR/VR, and medical image/video analysis. We hope that the discussed methods inspire future research and are used by others.

6.2 DISCUSSION AND FUTURE RESEARCH

Although great progresses have been made, there are limitations about the proposed methods. Here, we would like to discuss them and provide the possible future research directions.

Indiscernible object counting. In this thesis, we make solid contributions to dense prediction under indiscernible scenes by proposing a new task, a new large-scale dataset, and a novel approach. Even though the proposed model achieves state-of-the-art performance, it has certain drawbacks.

- It is specifically designed to make indiscernible objects stand out and therefore might not show significant advantages on common crowd counting datasets.
- it focuses on counting a single class (foreground) due to the limitation of existing datasets while models are required to count various classes with large count variations in real world.

To address the above limitations, there are several promising directions. First, it is necessary to design a unified model which can achieve state-of-the-art performance for both general and indiscernible scenes. Second, it is interesting to study a more realistic object counting scenario where various classes exist and the counts for different classes have large variation. Third, a more challenging direction is to design a open-vocabulary object counting approach which can deal with large count variations and switch the counting target.

Semantic segmentation with image-level labels. Even though the proposed method achieves promising results on three different settings and inspires further research [250, 251], there are some limitations.

- Like other semantic segmentation models using only image-level labels, our method has unsatisfactory performance on small objects.

- Since we follow a two-step pipeline for this task: first generate pseudo mask and then train fully supervised segmentation model, our approach is not end-to-end trainable.

To address the above limitations, there are several promising directions. First, a mixed of weak annotations can be explored to improve performance for small objects. For large objects, image-level labels generally work well. For small objects, a better but still cheap supervision such as point or scribble may be used. Semantic segmentation using mixed forms of weak supervisions could be a solution to achieve good performance for all objects while largely reducing annotation cost. Second, modifying the proposed method to be an end-to-end solution is interesting but non-trivial. Besides the proposed co-attention modules, other designs and techniques might be added to achieve the goal.

Semantic segmentation under dynamic scenes. In this thesis, we discuss algorithms for video semantic segmentation. The first work, CFFM, proposes to mine static and motional contexts in a unified framework, which are then exploited to refine the features for the target frame through non-self attention. The second work, MRCFA, improves CFFM by additionally mining the hyper-relations among the cross-frame attentions between the target and the references. Therefore, it generates better affinity maps, which are used to provide better segmentation masks. Nevertheless, there are certain drawbacks.

- Due to the memory limitations of the hardware, our method only uses the short-term temporal information, i.e., a few reference frames before the target frame. The long-term temporal information may also be helpful but is not studied in our research.
- Benefiting from the advanced attention mechanisms, MRCFA has the ability to focus on important contexts/tokens while discarding non-important ones. However, each frame of the videos still need to go through the image encoder. The property of large redundancy existing in videos is not explicitly explored from the input side. For example, two consecutive frames of a video have large overlapping for most times.

To address the above limitations, there are several promising directions. First, incorporating long-term temporal information in algorithm design is interesting. However, there are some challenges: memory limitation of GPUs and unavailability of extremely long video datasets. Brilliant designs in models or training techniques are possible to solve the memory problem, while dataset problem may be more difficult to deal with. Existing video datasets for VSS usually contain 150 frames (5 seconds) per video. There is a necessity for long video datasets which could contain 1000 frames (40 seconds) per video. The difficulties of long video datasets are in both storage and annotation. Second, exploiting the redundancy in videos to largely save computation and memory is worth exploring. Ideally, the computation for repeated regions in the videos could be saved. However, it is not the case for almost all video perception algorithms.

BIBLIOGRAPHY

1. He, K., Gkioxari, G., Dollár, P. & Girshick, R. *Mask r-cnn* in *ICCV* (2017).
2. Cholakkal, H., Sun, G., Khan, F. S. & Shao, L. *Object counting and instance segmentation with image-level supervision* in *CVPR* (2019).
3. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI* **40**, 834 (2018).
4. Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. *Pyramid scene parsing network* in *CVPR* (2017), 2881.
5. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *CVPR* (2016).
6. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* (2014).
7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. *Going deeper with convolutions* in *CVPR* (2015), 1.
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. *Imagenet: A large-scale hierarchical image database* in *CVPR* (2009).
9. Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. The pascal visual object classes challenge: A retrospective. *IJCV* **111**, 98 (2015).
10. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. *Microsoft coco: Common objects in context* in *ECCV* (2014).
11. Caesar, H., Uijlings, J. & Ferrari, V. *COCO-Stuff: Thing and stuff classes in context* in *CVPR* (2018), 1209.
12. Cholakkal, H., Sun, G., Khan, S., Khan, F. S., Shao, L. & Van Gool, L. Towards partial supervision for generic object counting in natural scenes. *IEEE TPAMI* **44**, 1604 (2022).
13. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE TPAMI* **39**, 1137 (2016).
14. Sun, G., Liu, Y., Probst, T., Paudel, D. P., Popovic, N. & Gool, L. V. Rethinking Global Context in Crowd Counting. *Machine Intelligence Research* (2023).
15. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M. & Luo, P. *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers* in *NeurIPS* (2021).
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. *Swin transformer: Hierarchical vision transformer using shifted windows* in *ICCV* (2021), 10012.
17. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L. & Timofte, R. *Swinir: Image restoration using swin transformer* in *ICCV* (2021), 1833.
18. Hu, J., Shen, L. & Sun, G. *Squeeze-and-excitation networks* in *CVPR* (2018).
19. Woo, S., Park, J., Lee, J.-Y. & So Kweon, I. *Cbam: Convolutional block attention module* in *ECCV* (2018).

20. Li, J., Wang, J., Tian, Q., Gao, W. & Zhang, S. *Global-local temporal representations for video person re-identification* in *Proceedings of the IEEE/CVF international conference on computer vision* (2019), 3958.
21. Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J. & Shao, L. *Camouflaged Object Detection* in *CVPR* (2020).
22. Le, T.-N., Cao, Y., Nguyen, T.-C., Le, M.-Q., Nguyen, K.-D., Do, T.-T., Tran, M.-T. & Nguyen, T. V. Camouflaged Instance Segmentation In-the-Wild: Dataset, Method, and Benchmark Suite. *IEEE TIP* **31**, 287 (2021).
23. Lamdouar, H., Yang, C., Xie, W. & Zisserman, A. *Betrayed by motion: Camouflaged object discovery via motion segmentation* in *ACCV* (2020).
24. Cheng, X., Xiong, H., Fan, D.-P., Zhong, Y., Harandi, M., Drummond, T. & Ge, Z. *Implicit motion handling for video camouflaged object detection* in *CVPR* (2022).
25. Lin, D., Dai, J., Jia, J., He, K. & Sun, J. *Scribblesup: Scribble-supervised convolutional networks for semantic segmentation* in *CVPR* (2016).
26. Papandreou, G., Chen, L.-C., Murphy, K. P. & Yuille, A. L. *Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation* in *ICCV* (2015).
27. Dai, J., He, K. & Sun, J. *Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation* in *ICCV* (2015).
28. Bearman, A., Russakovsky, O., Ferrari, V. & Fei-Fei, L. *What's the point: Semantic segmentation with point supervision* in *ECCV* (2016).
29. Pathak, D., Shelhamer, E., Long, J. & Darrell, T. Fully convolutional multi-class multiple instance learning. *arXiv preprint* (2014).
30. Lee, J., Kim, E., Lee, S., Lee, J. & Yoon, S. *Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference* in *CVPR* (2019).
31. Jiang, P.-T., Hou, Q., Cao, Y., Cheng, M.-M., Wei, Y. & Xiong, H.-K. *Integral Object Mining via Online Attention Accumulation* in *ICCV* (2019).
32. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A. & Agrawal, A. *Context encoding for semantic segmentation* in *CVPR* (2018), 7151.
33. Jin, Z., Gong, T., Yu, D., Chu, Q., Wang, J., Wang, C. & Shao, J. *Mining Contextual Information Beyond Image for Semantic Segmentation* in *ICCV* (2021), 7231.
34. Jin, Z., Liu, B., Chu, Q. & Yu, N. *ISNet: Integrate Image-Level and Semantic-Level Context for Semantic Segmentation* in *ICCV* (2021), 7189.
35. Yuan, Y., Chen, X. & Wang, J. *Object-contextual representations for semantic segmentation* in *ECCV* (2020), 173.
36. Liu, J., He, J., Qiao, Y., Ren, J. S. & Li, H. *Learning to predict context-adaptive convolution for semantic segmentation* in *ECCV* (2020), 769.
37. Li, X., Yang, Y., Zhao, Q., Shen, T., Lin, Z. & Liu, H. *Spatial pyramid based graph reasoning for semantic segmentation* in *CVPR* (2020), 8950.
38. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. *Semantic image segmentation with deep convolutional nets and fully connected CRFs* in *ICLR* (2015).
39. Chen, L.-C., Papandreou, G., Schroff, F. & Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

40. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. *Encoder-decoder with atrous separable convolution for semantic image segmentation* in *ECCV* (2018), 801.
41. Yu, F. & Koltun, V. *Multi-scale context aggregation by dilated convolutions* in *ICLR* (2016).
42. Yang, M., Yu, K., Zhang, C., Li, Z. & Yang, K. *DenseASPP for semantic segmentation in street scenes* in *CVPR* (2018), 3684.
43. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y. & Liu, W. *CCNet: Criss-cross attention for semantic segmentation* in *ICCV* (2019), 603.
44. Zhu, Z., Xu, M., Bai, S., Huang, T. & Bai, X. *Asymmetric non-local neural networks for semantic segmentation* in *ICCV* (2019), 593.
45. Zhen, M., Wang, J., Zhou, L., Li, S., Shen, T., Shang, J., Fang, T. & Quan, L. *Joint semantic segmentation and boundary detection using iterative pyramid contexts* in *CVPR* (2020), 13666.
46. Gadde, R., Jampani, V. & Gehler, P. V. *Semantic video CNNs through representation warping* in *ICCV* (2017), 4453.
47. Nilsson, D. & Sminchisescu, C. *Semantic video segmentation by gated recurrent flow propagation* in *CVPR* (2018), 6819.
48. Liu, S., Wang, C., Qian, R., Yu, H., Bao, R. & Sun, Y. *Surveillance video parsing with single frame supervision* in *CVPR* (2017), 413.
49. Jin, X., Li, X., Xiao, H., Shen, X., Lin, Z., Yang, J., Chen, Y., Dong, J., Liu, L., Jie, Z., *et al.* *Video scene parsing with predictive feature learning* in *ICCV* (2017), 5580.
50. Xu, Y.-S., Fu, T.-J., Yang, H.-K. & Lee, C.-Y. *Dynamic video segmentation network* in *CVPR* (2018), 6556.
51. Liu, Y., Shen, C., Yu, C. & Wang, J. *Efficient semantic video segmentation with per-frame inference* in *ECCV* (2020), 352.
52. Hu, P., Caba, F., Wang, O., Lin, Z., Sclaroff, S. & Perazzi, F. *Temporally distributed networks for fast video semantic segmentation* in *CVPR* (2020), 8818.
53. Shelhamer, E., Rakelly, K., Hoffman, J. & Darrell, T. *Clockwork convnets for video semantic segmentation* in *ECCV* (2016), 852.
54. Jain, S., Wang, X. & Gonzalez, J. E. *Accel: A corrective fusion network for efficient semantic segmentation on video* in *CVPR* (2019), 8866.
55. Carreira, J., Patraucean, V., Mazare, L., Zisserman, A. & Osindero, S. *Massively parallel video networks* in *ECCV* (2018), 649.
56. Zhu, X., Xiong, Y., Dai, J., Yuan, L. & Wei, Y. *Deep feature flow for video recognition* in *CVPR* (2017), 2349.
57. Li, Y., Shi, J. & Lin, D. *Low-latency video semantic segmentation* in *CVPR* (2018), 5997.
58. Lee, S.-P., Chen, S.-C. & Peng, W.-H. *GSVNet: Guided Spatially-Varying Convolution for Fast Semantic Segmentation on Video* in *ICME* (2021), 1.
59. Paul, M., Danelljan, M., Van Gool, L. & Timofte, R. *Local memory attention for fast video semantic segmentation* in *IROS* (2021), 1102.
60. Li, J., Wang, W., Chen, J., Niu, L., Si, J., Qian, C. & Zhang, L. *Video Semantic Segmentation via Sparse Temporal Transformer* in *ACM MM* (2021), 59.

61. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D. & Brox, T. *FlowNet: Learning optical flow with convolutional networks* in *ICCV* (2015), 2758.
62. Narayan, S., Cholakkal, H., Khan, F. S. & Shao, L. *3c-net: Category count and center loss for weakly-supervised action localization* in *ICCV* (2019).
63. Xie, J., Cholakkal, H., Muhammad Anwer, R., Shahbaz Khan, F., Pang, Y., Shao, L. & Shah, M. *Count-and similarity-aware R-CNN for pedestrian detection* in *ECCV* (2020).
64. Wang, M. & Wang, X. *Automatic adaptation of a generic pedestrian detector to a specific traffic scene* in *CVPR* (2011).
65. Chan, A. B., Liang, Z.-S. J. & Vasconcelos, N. *Privacy preserving crowd monitoring: Counting people without people models or tracking* in *CVPR* (2008).
66. Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C. & Clune, J. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *PNAS* **115**, E5716 (2018).
67. Nguyen, H.-T., Ngo, C.-W. & Chan, W.-K. SibNet: Food instance counting and segmentation. *PR* **124**, 108470 (2022).
68. Alam, M. M. & Islam, M. T. Machine learning approach of automatic identification and counting of blood cells. *HTL* **6**, 103 (2019).
69. Chattopadhyay, P., Vedantam, R., Selvaraju, R. R., Batra, D. & Parikh, D. *Counting everyday objects in everyday scenes* in *CVPR* (2017).
70. Laradji, I. H., Rostamzadeh, N., Pinheiro, P. O., Vazquez, D. & Schmidt, M. *Where are the blobs: Counting by localization with point supervision* in *ECCV* (2018).
71. Stahl, T., Pintea, S. L. & Van Gemert, J. C. Divide and count: Generic object counting by image divisions. *IEEE TIP* **28**, 1035 (2018).
72. Zhang, Y., Zhou, D., Chen, S., Gao, S. & Ma, Y. *Single-image crowd counting via multi-column convolutional neural network* in *CVPR* (2016).
73. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N. & Shah, M. *Composition loss for counting, density map estimation and localization in dense crowds* in *ECCV* (2018).
74. Li, Y., Zhang, X. & Chen, D. *CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes* in *CVPR* (2018).
75. Sindagi, V., Yasarla, R. & Patel, V. M. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE TPAMI* **44**, 2594 (2022).
76. Onoro-Rubio, D. & López-Sastre, R. J. *Towards perspective-free object counting with deep learning* in *ECCV* (2016).
77. Lu, H., Cao, Z., Xiao, Y., Zhuang, B. & Shen, C. TasselNet: counting maize tassels in the wild via local counts regression network. *PM* **13**, 79 (2017).
78. Song, Q., Wang, C., Wang, Y., Tai, Y., Wang, C., Li, J., Wu, J. & Ma, J. *To choose or to fuse? scale selection for crowd counting* in *AAAI* (2021).
79. Liang, D., Xu, W. & Bai, X. *An end-to-end transformer model for crowd localization* in *ECCV* (2022).
80. Hsieh, M.-R., Lin, Y.-L. & Hsu, W. H. *Drone-based object counting by spatially regularized regional proposal network* in *ICCV* (2017).

81. Wang, Q., Gao, J., Lin, W. & Li, X. NWPU-Crowd: A Large-Scale Benchmark for Crowd Counting and Localization. *IEEE TPAMI* (2020).
82. Sindagi, V. A., Yasarla, R. & Patel, V. M. *Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method* in *ICCV* (2019).
83. Chen, K., Loy, C. C., Gong, S. & Xiang, T. *Feature mining for localised crowd counting.* in *BMVC* (2012).
84. Le, T.-N., Nguyen, T. V., Nie, Z., Tran, M.-T. & Sugimoto, A. Anabranch network for camouflaged object segmentation. *CVIU* **184**, 45 (2019).
85. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. *An image is worth 16x16 words: Transformers for image recognition at scale* in *ICLR* (2021).
86. Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. L. & Fei-Fei, L. *Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation* in *CVPR* (2019).
87. Bolya, D., Zhou, C., Xiao, F. & Lee, Y. J. *YOLACT: Real-time Instance Segmentation* in *ICCV* (2019).
88. Lyu, Y., Zhang, J., Dai, Y., Li, A., Liu, B., Barnes, N. & Fan, D.-P. *Simultaneously Localize, Segment and Rank the Camouflaged Objects* in *CVPR* (2021).
89. Liu, W., Salzmann, M. & Fua, P. *Context-aware crowd counting* in *CVPR* (2019).
90. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W. & Lin, L. *Crowd counting with deep structured scale integration network* in *ICCV* (2019).
91. Ma, Z., Wei, X., Hong, X. & Gong, Y. *Bayesian loss for crowd count estimation with point supervision* in *ICCV* (2019).
92. Wan, J. & Chan, A. *Modeling Noisy Annotations for Crowd Counting* in *NeurIPS* (2020).
93. Wang, B., Liu, H., Samaras, D. & Hoai, M. *Distribution Matching for Crowd Counting* in *NeurIPS* (2020).
94. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F. & Wu, Y. *Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework* in *ICCV* (2021).
95. Zand, M., Damirchi, H., Farley, A., Molahasani, M., Greenspan, M. & Etemad, A. *Multiscale Crowd Counting and Localization By Multitask Point Supervision* in *ICASSP* (2022).
96. Lin, H., Ma, Z., Ji, R., Wang, Y. & Hong, X. *Boosting Crowd Counting via Multifaceted Attention* in *CVPR* (2022).
97. Idrees, H., Saleemi, I., Seibert, C. & Shah, M. *Multi-source multi-scale counting in extremely dense crowd images* in *CVPR* (2013).
98. Zhang, C., Kang, K., Li, H., Wang, X., Xie, R. & Yang, X. Data-driven crowd understanding: A baseline for a large-scale crowd dataset. *IEEE TMM* **18**, 1048 (2016).
99. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S. & Ding, E. *Perspective-guided convolution networks for crowd counting* in *ICCV* (2019).
100. Wang, Q., Gao, J., Lin, W. & Yuan, Y. *Learning from synthetic data for crowd counting in the wild* in *CVPR* (2019).

101. Fan, D.-P., Ji, G.-P., Cheng, M.-M. & Shao, L. Concealed object detection. *IEEE TPAMI* **44**, 6024 (2022).

102. Shu, W., Wan, J., Tan, K. C., Kwong, S. & Chan, A. B. *Crowd Counting in the Frequency Domain* in *CVPR* (2022).

103. Cheng, Z.-Q., Dai, Q., Li, H., Song, J., Wu, X. & Hauptmann, A. G. *Rethinking Spatial Invariance of Convolutional Networks for Object Counting* in *CVPR* (2022).

104. Liu, L., Lu, H., Zou, H., Xiong, H., Cao, Z. & Shen, C. *Weighing counts: Sequential crowd counting by reinforcement learning* in *ECCV* (2020).

105. Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., Yang, X. & Pang, Y. *Attention scaling for crowd counting* in *CVPR* (2020).

106. Arteta, C., Lempitsky, V. & Zisserman, A. *Counting in the wild* in *ECCV* (2016).

107. Ge, W. & Collins, R. T. *Marked point processes for crowd counting* in *CVPR* (2009).

108. Liu, X., Van De Weijer, J. & Bagdanov, A. D. Exploiting unlabeled data in CNNs by self-supervised learning to rank. *IEEE TPAMI* **41**, 1862 (2019).

109. Liang, D., Chen, X., Xu, W., Zhou, Y. & Bai, X. TransCrowd: weakly-supervised crowd counting with transformers. *SCIS* **65**, 1 (2022).

110. Qian, Y., Zhang, L., Hong, X., Donovan, C. R. & Arandjelovic, O. *Segmentation Assisted U-shaped Multi-scale Transformer for Crowd Counting* in *BMVC* (2022).

111. Zhong, Y., Li, B., Tang, L., Kuang, S., Wu, S. & Ding, S. *Detecting Camouflaged Object in Frequency Domain* in *CVPR* (2022).

112. Liu, W., Durasov, N. & Fua, P. *Leveraging Self-Supervision for Cross-Domain Crowd Counting* in *CVPR* (2022).

113. Gong, S., Zhang, S., Yang, J., Dai, D. & Schiele, B. *Bi-level Alignment for Cross-Domain Crowd Counting* in *CVPR* (2022).

114. Chen, B., Yan, Z., Li, K., Li, P., Wang, B., Zuo, W. & Zhang, L. *Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting* in ICCV (2021).

115. He, Y., Ma, Z., Wei, X., Hong, X., Ke, W. & Gong, Y. *Error-aware density isomorphism reconstruction for unsupervised cross-domain crowd counting* in *AAAI* (2021).

116. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. *Attention is all you need* in *NeurIPS* (2017).

117. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. *End-to-end object detection with transformers* in *ECCV* (2020).

118. Wan, J., Liu, Z. & Chan, A. B. *A generalized loss function for crowd counting and localization* in *CVPR* (2021).

119. Wan, J., Wang, Q. & Chan, A. B. Kernel-Based Density Map Generation for Dense Object Counting. *IEEE TPAMI* **44**, 1357 (2022).

120. Kingma, D. P. & Ba, J. *Adam: A method for stochastic optimization* in *ICLR* (2015).

121. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al. PyTorch: An imperative style, high-performance deep learning library* in *NeurIPS* (2019), 8026.

122. Caruana, R. Multitask learning. *ML* **28**, 41 (1997).

123. Weinberger, K., Dasgupta, A., Langford, J., Smola, A. & Attenberg, J. *Feature hashing for large scale multitask learning* in *ICML* (2009).

124. Sun, G., Probst, T., Paudel, D. P., Popović, N., Kanakis, M., Patel, J., Dai, D. & Van Gool, L. *Task switching network for multi-task learning* in *ICCV* (2021).

125. Zeiler, M. D. & Fergus, R. *Visualizing and understanding convolutional networks* in *ECCV* (2014).

126. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. *Learning deep features for discriminative localization* in *CVPR* (2016).

127. Kumar Singh, K. & Jae Lee, Y. *Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization* in *ICCV* (2017).

128. Wei, Y., Feng, J., Liang, X., Cheng, M.-M., Zhao, Y. & Yan, S. *Object region mining with adversarial erasing: A simple classification to semantic segmentation approach* in *CVPR* (2017).

129. Kolesnikov, A. & Lampert, C. H. *Seed, expand and constrain: Three principles for weakly-supervised image segmentation* in *ECCV* (2016).

130. Wang, X., You, S., Li, X. & Ma, H. *Weakly-supervised semantic segmentation by iteratively mining common object features* in *CVPR* (2018).

131. Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J. & Huang, T. S. *Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation* in *CVPR* (2018).

132. Lee, S., Lee, J., Lee, J., Park, C.-K. & Yoon, S. Robust tumor localization with pyramid grad-cam. *arXiv preprint* (2018).

133. Pinheiro, P. O. & Collobert, R. *From image-level to pixel-level labeling with convolutional networks* in *CVPR* (2015).

134. Li, K., Zhang, Y., Li, K., Li, Y. & Fu, Y. *Attention bridging network for knowledge transfer* in *ICCV* (2019).

135. Shen, T., Lin, G., Liu, L., Shen, C. & Reid, I. *Weakly supervised semantic segmentation based on Web image co-segmentation* in *BMVC* (2017).

136. Hong, S., Yeo, D., Kwak, S., Lee, H. & Han, B. *Weakly supervised semantic segmentation using web-crawled videos* in *CVPR* (2017).

137. Shen, T., Lin, G., Shen, C. & Reid, I. *Bootstrapping the performance of webly supervised semantic segmentation* in *CVPR* (2018).

138. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.-M., Feng, J., Zhao, Y. & Yan, S. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *TPAMI* **39**, 2314 (2016).

139. Tokmakov, P., Alahari, K. & Schmid, C. *Weakly-supervised semantic segmentation using motion cues* in *ECCV* (2016).

140. Fang, H., Lu, G., Fang, X., Xie, J., Tai, Y. & Lu, C. *Weakly and Semi Supervised Human Body Part Parsing via Pose-Guided Knowledge Transfer* in *CVPR* (2018).

141. Li, K., Wu, Z., Peng, K.-C., Ernst, J. & Fu, Y. *Tell me where to look: Guided attention inference network* in *CVPR* (2018).

142. Huang, Z., Wang, X., Wang, J., Liu, W. & Wang, J. *Weakly-supervised semantic segmentation network with deep seeded region growing* in *CVPR* (2018).

143. Ahn, J. & Kwak, S. *Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation* in *CVPR* (2018).
144. Shimoda, W. & Yanai, K. *Self-supervised difference detection for weakly-supervised semantic segmentation* in *ICCV* (2019).
145. Fan, J., Zhang, Z. & Tan, T. *Cian: Cross-image affinity net for weakly supervised semantic segmentation* in *AAAI* (2020).
146. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.* ImageNet large scale visual recognition challenge. *IJCV* **115**, 211 (2015).
147. Griffin, G., Holub, A. & Perona, P. Caltech-256 object category dataset (2007).
148. Lee, J., Kim, E., Lee, S., Lee, J. & Yoon, S. *Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation* in *ICCV* (2019).
149. Jin, B., Ortiz Segovia, M. V. & Susstrunk, S. *Webly supervised semantic segmentation* in *ICCV* (2017).
150. Joulin, A., Bach, F. & Ponce, J. *Discriminative clustering for image co-segmentation* in *CVPR* (2010).
151. Luong, M.-T., Pham, H. & Manning, C. D. *Effective approaches to attention-based neural machine translation* in *EMNLP* (2015).
152. Cheng, J., Dong, L. & Lapata, M. *Long short-term memory-networks for machine reading* in *EMNLP* (2016).
153. Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B. & Bengio, Y. *A structured self-attentive sentence embedding* in *ICLR* (2017).
154. Paulus, R., Xiong, C. & Socher, R. *A deep reinforced model for abstractive summarization* in *ICLR* (2018).
155. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z. & Lu, H. *Dual attention network for scene segmentation* in *CVPR* (2019).
156. Sun, M., Yuan, Y., Zhou, F. & Ding, E. *Multi-attention multi-class constraint for fine-grained image recognition* in *ECCV* (2018).
157. Wang, X., Girshick, R., Gupta, A. & He, K. *Non-local neural networks* in *CVPR* (2018).
158. Wang, X., Li, L., Ye, W., Long, M. & Wang, J. *Transferable attention for domain adaptation* in *AAAI* (2019).
159. Zhang, Y., Nie, S., Liu, W., Xu, X., Zhang, D. & Shen, H. T. *Sequence-to-sequence domain adaptation network for robust text image recognition* in *CVPR* (2019).
160. Wang, W., Zhu, H., Dai, J., Pang, Y., Shen, J. & Shao, L. *Hierarchical human parsing with typed part-relation reasoning* in *CVPR* (2020).
161. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A. L. & Wang, X. *Multi-context attention for human pose estimation* in *CVPR* (2017).
162. Ye, Q., Yuan, S. & Kim, T.-K. *Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation* in *ECCV* (2016).
163. Cao, J., Pang, Y. & Li, X. Triply supervised decoder networks for joint detection and segmentation. *CVPR* (2019).

164. Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B. & Bai, X. *Progressive pose attention transfer for person image generation* in *CVPR* (2019).

165. Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. *Self-attention generative adversarial networks* in *ICML* (2019).

166. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X. & He, X. *Attngan: Fine-grained text to image generation with attentional generative adversarial networks* in *CVPR* (2018).

167. Lu, J., Yang, J., Batra, D. & Parikh, D. *Hierarchical question-image co-attention for visual question answering* in *NeurIPS* (2016).

168. Xiong, C., Zhong, V. & Socher, R. *Dynamic coattention networks for question answering* in *ICLR* (2017).

169. Nguyen, D.-K. & Okatani, T. *Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering* in *CVPR* (2018).

170. Yu, Z., Yu, J., Cui, Y., Tao, D. & Tian, Q. *Deep modular co-attention networks for visual question answering* in *CVPR* (2019).

171. Zheng, Z., Wang, W., Qi, S. & Zhu, S.-C. *Reasoning visual dialogs with structural and partial observations* in *CVPR* (2019).

172. Wu, Q., Wang, P., Shen, C., Reid, I. & Van Den Hengel, A. *Are you talking to me? reasoned visual dialog generation through adversarial learning* in *CVPR* (2018).

173. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.-F., Wang, W. Y. & Zhang, L. *Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation* in *CVPR* (2019).

174. Wang, W., Lu, X., Shen, J., Crandall, D. J. & Shao, L. *Zero-shot video object segmentation via attentive graph neural networks* in *ICCV* (2019).

175. Lu, X., Wang, W., Ma, C., Shen, J., Shao, L. & Porikli, F. *See more, know more: Unsupervised video object segmentation with co-attention siamese networks* in *CVPR* (2019).

176. Odena, A., Olah, C. & Shlens, J. *Conditional image synthesis with auxiliary classifier gans* in *ICML* (2017).

177. Gidaris, S., Singh, P. & Komodakis, N. *Unsupervised representation learning by predicting image rotations* in *ICLR* (2018).

178. Wang, W. & Shen, J. Higher-order image co-segmentation. *IEEE TMM* **18**, 1011 (2016).

179. Hou, Q., Jiang, P., Wei, Y. & Cheng, M.-M. *Self-erasing network for integral object attention* in *NeurIPS* (2018).

180. Zeng, Y., Zhuge, Y., Lu, H. & Zhang, L. *Joint learning of saliency detection and weakly supervised semantic segmentation* in *ICCV* (2019).

181. Hou, Q., Cheng, M.-M., Hu, X., Borji, A., Tu, Z. & Torr, P. Deeply Supervised Salient Object Detection with Short Connections. *TPAMI* **41**, 815 (2019).

182. Liu, J.-J., Hou, Q., Cheng, M.-M., Feng, J. & Jiang, J. *A Simple Pooling-Based Design for Real-Time Salient Object Detection* in *CVPR* (2019).

183. Pan, B., Cao, Z., Adeli, E. & Niebles, J. C. *Adversarial Cross-Domain Action Recognition with Co-Attention* in *AAAI* (2020).

184. Wataru, S. & Keiji, Y. *Distinct class saliency maps for weakly supervised semantic segmentation* in *ECCV* (2016).

185. Qi, X., Liu, Z., Shi, J., Zhao, H. & Jia, J. *Augmented feedback in semantic segmentation under image level supervision* in *ECCV* (2016).
186. Chaudhry, A., Dokania, P. K. & Torr, P. H. *Discovering class-specific pixels for weakly-supervised semantic segmentation* in *BMVC* (2017).
187. Roy, A. & Todorovic, S. *Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation* in *CVPR* (2017).
188. Kim, D., Cho, D., Yoo, D. & So Kweon, I. *Two-phase learning for weakly supervised object localization* in *ICCV* (2017).
189. Ge, W., Yang, S. & Yu, Y. *Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning* in *CVPR* (2018).
190. Ahn, J., Cho, S. & Kwak, S. *Weakly supervised learning of instance segmentation with inter-pixel relations* in *CVPR* (2019), 2209.
191. Zhang, X., Wei, Y., Feng, J., Yang, Y. & Huang, T. S. *Adversarial complementary learning for weakly supervised object localization* in *CVPR* (2018).
192. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. **25**, 1097 (2012).
193. Krähenbühl, P. & Koltun, V. *Efficient inference in fully connected crfs with gaussian edge potentials* in *NeurIPS* (2011).
194. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S. & Malik, J. *Semantic contours from inverse detectors* in *ICCV* (2011).
195. Wei, Y., Zheng, S., Cheng, M.-M., Zhao, H., Wang, L., Ding, E., Yang, Y., Torralba, A., Liu, T., Sun, G., *et al.* LID 2020: The Learning from Imperfect Data Challenge Results (2020).
196. Wu, Z., Shen, C. & Van Den Hengel, A. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition* **90**, 119 (2019).
197. Wang, Y., Zhang, J., Kan, M., Shan, S. & Chen, X. *Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation* in *CVPR* (2020).
198. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. *The Cityscapes dataset for semantic urban scene understanding* in *CVPR* (2016), 3213.
199. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A. & Torralba, A. Semantic understanding of scenes through the ADE20K dataset. *IJCV* **127**, 302 (2019).
200. Silberman, N., Hoiem, D., Kohli, P. & Fergus, R. *Indoor segmentation and support inference from RGBD images* in *ECCV* (2012), 746.
201. Brostow, G. J., Shotton, J., Fauqueur, J. & Cipolla, R. *Segmentation and Recognition Using Structure from Motion Point Clouds* in *ECCV* (2008), 44.
202. Miao, J., Wei, Y., Wu, Y., Liang, C., Li, G. & Yang, Y. *VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild* in *CVPR* (2021), 4133.
203. Xu, W., Xu, Y., Chang, T. & Tu, Z. *Co-Scale Conv-Attentional Image Transformers* in *ICCV* (2021), 9981.
204. Heo, B., Yun, S., Han, D., Chun, S., Choe, J. & Oh, S. J. *Rethinking spatial dimensions of vision transformers* in *ICCV* (2021), 11936.

205. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H. & Shen, C. *Twins: Revisiting the design of spatial attention in vision transformers* in *NeurIPS* (2021).

206. Shelhamer, E., Long, J. & Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE TPAMI* **39**, 640 (2017).

207. Zhu, L., Ji, D., Zhu, S., Gan, W., Wu, W. & Yan, J. *Learning Statistical Texture for Semantic Segmentation* in *CVPR* (2021), 12537.

208. Wang, L., Li, D., Zhu, Y., Tian, L. & Shan, Y. *Dual super-resolution learning for semantic segmentation* in *CVPR* (2020), 3774.

209. Pang, Y., Li, Y., Shen, J. & Shao, L. *Towards bridging semantic gap to improve semantic segmentation* in *ICCV* (2019), 4230.

210. Nirkin, Y., Wolf, L. & Hassner, T. *HyperSeg: Patch-wise hypernetwork for real-time semantic segmentation* in *CVPR* (2021), 4061.

211. Hu, H., Ji, D., Gan, W., Bai, S., Wu, W. & Yan, J. *Class-wise dynamic graph convolution for semantic segmentation* in *ECCV* (2020), 1.

212. Zhang, H., Zhang, H., Wang, C. & Xie, J. *Co-occurrent features in semantic segmentation* in *CVPR* (2019), 548.

213. Wei, Z., Zhang, J., Liu, L., Zhu, F., Shen, F., Zhou, Y., Liu, S., Sun, Y. & Shao, L. *Building detail-sensitive semantic segmentation networks with polynomial pooling* in *CVPR* (2019), 7115.

214. Ronneberger, O., Fischer, P. & Brox, T. *U-Net: Convolutional networks for biomedical image segmentation* in *MICCAI* (2015), 234.

215. Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI* **39**, 2481 (2017).

216. Tian, Z., He, T., Shen, C. & Yan, Y. *Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation* in *CVPR* (2019), 3126.

217. Liu, J., He, J., Zhang, J., Ren, J. S. & Li, H. *EfficientFCN: Holistically-guided decoding for semantic segmentation* in *ECCV* (2020), 1.

218. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. & Sang, N. *Learning a discriminative feature network for semantic segmentation* in *CVPR* (2018), 1857.

219. Takikawa, T., Acuna, D., Jampani, V. & Fidler, S. *Gated-SCNN: Gated shape CNNs for semantic segmentation* in *ICCV* (2019), 5229.

220. Li, X., Li, X., Zhang, L., Cheng, G., Shi, J., Lin, Z., Tan, S. & Tong, Y. *Improving semantic segmentation via decoupled body and edge supervision* in *ECCV* (2020), 435.

221. Wang, C., Zhang, Y., Cui, M., Liu, J., Ren, P., Yang, Y., Xie, X., Hua, X., Bao, H. & Xu, W. Active Boundary Loss for Semantic Segmentation. *arXiv preprint arXiv:2102.02696* (2021).

222. Chen, L.-C., Yang, Y., Wang, J., Xu, W. & Yuille, A. L. *Attention to scale: Scale-aware semantic image segmentation* in *CVPR* (2016), 3640.

223. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z. & Liu, H. *Expectation-maximization attention networks for semantic segmentation* in *ICCV* (2019), 9167.

224. Seifi, S. & Tuytelaars, T. *Attend and Segment: Attention Guided Active Semantic Segmentation* in *ECCV* (2020), 305.

225. He, J., Deng, Z. & Qiao, Y. *Dynamic multi-scale filters for semantic segmentation* in *ICCV* (2019), 3562.

226. Kundu, A., Vineet, V. & Koltun, V. *Feature space optimization for semantic video segmentation* in *CVPR* (2016), 3168.

227. Mahasseni, B., Todorovic, S. & Fern, A. *Budget-aware deep semantic video segmentation* in *CVPR* (2017), 1029.

228. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. *Generative adversarial nets* in *NeurIPS* (2014).

229. Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S., Tao, A. & Catanzaro, B. *Improving semantic segmentation via video propagation and label relaxation* in *CVPR* (2019), 8856.

230. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L. & Gao, J. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641* (2021).

231. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J. & Yan, S. *Tokens-to-Token ViT: Training Vision Transformers From Scratch on ImageNet* in *ICCV* (2021), 558.

232. Liu, Y., Wu, Y.-H., Sun, G., Zhang, L., Chhatkuli, A. & Van Gool, L. Vision transformers with hierarchical attention. *arXiv preprint arXiv:2106.03180* (2021).

233. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. & Zhang, L. *Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers* in *CVPR* (2021), 6881.

234. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J. & Zhou, J. *Pointr: Diverse point cloud completion with geometry-aware transformers* in *ICCV* (2021), 12498.

235. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X. & Lu, H. *Transformer tracking* in *CVPR* (2021), 8126.

236. Yan, B., Peng, H., Fu, J., Wang, D. & Lu, H. *Learning spatio-temporal transformer for visual tracking* in *ICCV* (2021), 10448.

237. Liang, D., Chen, X., Xu, W., Zhou, Y. & Bai, X. TransCrowd: Weakly-supervised crowd counting with transformers. *Sci. China Inform. Sci.* **65**, 1 (2022).

238. Lanchantin, J., Wang, T., Ordonez, V. & Qi, Y. *General multi-label image classification with transformers* in *CVPR* (2021), 16478.

239. Wang, H., Zhu, Y., Adam, H., Yuille, A. & Chen, L.-C. *Max-DeepLab: End-to-end panoptic segmentation with mask transformers* in *CVPR* (2021), 5463.

240. Cheng, B., Schwing, A. & Kirillov, A. *Per-pixel classification is not all you need for semantic segmentation* in *NeurIPS* (2021), 17864.

241. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. & Girdhar, R. *Masked-attention mask transformer for universal image segmentation* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), 1290.

242. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P. & Shao, L. *Pyramid vision transformer: A versatile backbone for dense prediction without convolutions* in *ICCV* (2021), 568.

243. Contributors, M. *MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark* `https://github.com/open-mmlab/mmsegmentation`. 2020.

244. Xiao, T., Liu, Y., Zhou, B., Jiang, Y. & Sun, J. *Unified perceptual parsing for scene understanding* in *ECCV* (2018), 418.

245. Sun, G., Wang, W., Dai, J. & Van Gool, L. *Mining cross-image semantics for weakly supervised semantic segmentation* in *European Conference on Computer Vision* (2020), 347.

246. Paul, M., Mayer, C., Gool, L. V. & Timofte, R. *Efficient video semantic segmentation with labels propagation and refinement* in *WACV* (2020), 2873.

247. Sun, G., Liu, Y., Ding, H., Probst, T. & Van Gool, L. *Coarse-to-fine feature mining for video semantic segmentation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 3126.

248. Brostow, G. J., Fauqueur, J. & Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* **30**, 88 (2009).

249. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S. & Hu, H. Video swin transformer. *arXiv preprint arXiv:2106.13230* (2021).

250. Wu, L., Fang, L., He, X., He, M., Ma, J. & Zhong, Z. Querying labeled for unlabeled: Cross-image semantic consistency guided semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

251. Zhou, T., Zhang, M., Zhao, F. & Li, J. *Regional semantic contrast and aggregation for weakly supervised semantic segmentation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 4299.

CURRICULUM VITAE

PERSONAL DATA

Name	Guolei Sun
Place of Birth	Wuhan, China
Citizen of	China

EDUCATION

2019.9 – 2024.1	ETH Zürich, Switzerland Doctoral Studies in Computer Vision Lab
2016.1 – 2018.7	KAUST, Saudi Arabia MSc in Computer Science
2011.9 – 2015.7	Huazhong University of Science and Technology, China BSc in Science

EMPLOYMENT

2023.10 – 2024.1	Research Scientist intern Meta New York, the United States
2023.5 – 2023.8	Research Scientist intern Adobe San Jose, the United States
2018.8 – 2019.9	Research Engineer Inception Institute of Artificial Intelligence Abu Dhabi, the United Emirates

PROFESSIONAL ACTIVITIES

2018 – present	Journal Reviewer IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) International Journal of Computer Vision (IJCV) IEEE Transactions on Image Processing (TIP) IEEE Transactions on Neural Networks and Learning Systems (TNNLS) IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)
2018 – present	Conference Reviewer The IEEE / CVF Computer Vision and Pattern Recognition (CVPR) International Conference on Computer Vision (ICCV) European Conference on Computer Vision (ECCV) International Conference on Learning Representations (ICLR) Conference on Neural Information Processing Systems (NeurIPS)