Multi-Object Multi-Part Scene Parsing

Thesis submitted in partial fulfillment
of the requirements for the degree of

Masters of Science
in
Computer Science and Engineering by Research

by

Pranav Gupta
2021701043
pranav.gu@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
March, 2025

Copyright © Pranav Gupta, 2025
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ‘Multi-Object Multi-Part Scene Parsing‘ by Pranav Gupta, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Ravi Kiran Sarvadevabhatla

To, my family and friends

Acknowledgments

To begin with, I would like to extend my heartfelt gratitude to my advisor, Prof. S. Ravi Kiran, for his unwavering guidance and support throughout this incredible journey. His insightful advice has not only helped me become a better researcher but has also played a significant role in shaping my moral and ethical character. Prof. Ravi Kiran's meticulous attention to the finer details of my presentations has significantly enhanced my presentation skills, making me more confident and articulate in sharing my work.

Additionally, I would like to express my sincere appreciation to my co-advisor, Dr. Pradeep Shenoy, and my co-author, Rishubh Singh, for their invaluable guidance and assistance throughout my research journey. The discussions we had about the project were intellectually stimulating and deeply enjoyable. Furthermore, our conversations about our future plans provided reassurance and a sense of control over our respective paths, fostering a collaborative and supportive environment.

I am also profoundly grateful to my friends Shivanshu Sharma, Ravi Mishra, BVK, Zeeshan Khan, Madhav Agarwal, Neha Sahipjhon and Shubham Dokania. They have made my college life incredibly enjoyable, offering the much-needed balance between work and leisure. Our study sessions together were instrumental in helping me manage my coursework and research effectively. Their impact on my cultural tastes has been immense, and I feel extremely fortunate to have shared wonderful memories with such amazing friends over the years.

Finally, I would like to extend my deepest thanks to my family for their unwavering support and motivation. Their trust in me has allowed me to focus on my research without any worries, and their encouragement during times of distress and failure has been a crucial factor in helping me persevere and succeed.

This thesis would not have been possible without the support of many incredible people. I dedicate this work to all of them and sincerely hope to have their continued support in my future endeavors as well.

Abstract

This thesis presents a comprehensive exploration of Multi-Object Multi-Part Scene Parsing in 2D images, showcasing significant advancements through two novel approaches, FLOAT and OLAF, each tailored to enhance scene parsing performance and scalability. The first paper introduces FLOAT, a factorized label space framework designed to independently predict object categories and part attributes, thereby simplifying the segmentation task and enhancing scalability. Notably, FLOAT incorporates a unique 'zoom' refinement technique at inference time, significantly elevating segmentation accuracy, particularly for smaller objects and parts. Empirical results on the Pascal-Part datasets underscore FLOAT's superior performance, achieving notable improvements in mean Intersection Over Union (mIOU) and segmentation quality IOU (sqIOU), especially on the most comprehensive Pascal-Part-201 dataset, reflecting its effectiveness in handling diverse and complex scenes.

The second paper delves into OLAF, a plug-and-play methodology that augments traditional RGB inputs with object-based structural cues to better capture the complexities of scene structures. This approach leverages a weight adaptation technique, allowing pre-trained RGB models to seamlessly integrate augmented data, thus stabilizing the optimization process. Additionally, the introduction of the LDF encoder module aids in providing low-level dense feature guidance, enhancing the segmentation of smaller parts. OLAF demonstrates its versatility across various architectures and datasets, achieving significant mIOU gains on multiple Pascal-Part benchmarks, highlighting its broad applicability and robust performance enhancements in challenging segmentation scenarios.

Together, these studies contribute to the evolving field of computer vision by offering scalable, efficient, and effective solutions for multi-object multi-part scene parsing, reflecting a significant stride in parsing intricate scenes with high granularity and diversity.

Contents

Chapter Page

1 Introduction . . . 1
1.1 Multi-Object Multi-Part Scene Parsing . . . 2
1.1.1 History and Popular Approaches: . . . 2
1.1.1.1 Some of The Popular Approaches: . . . 2
1.2 Challenges in Multi-Object Multi-Part Scene Parsing . . . 3
1.3 Our Contributions . . . 4
1.4 Thesis Layout . . . 4

2 FLOAT: Factorized Learning of Object Attributes for Improved Multi-object Multi-part Scene Parsing . . . 6
2.1 Introduction . . . 6
2.2 Related Works . . . 7
2.3 Our framework (FLOAT) . . . 9
2.3.1 Relabeling images with factored labels . . . 9
2.3.2 Factorized semantic segmentation architecture . . . 10
2.3.3 Top-Down Merge . . . 10
2.3.4 Inference-time Zoom Refinement (IZR) . . . 11
2.3.5 Optimization . . . 12
2.3.6 Implementation and Training Details . . . 14
2.4 Datasets and Evaluation Metrics . . . 14
2.4.1 Evaluation Metrics . . . 15
2.5 Experimental Results . . . 16
2.5.1 Pascal-Part-201 . . . 16
2.5.2 Pascal-Part-58 and Pascal-Part-108 . . . 18
2.5.3 Ablation Studies . . . 19
2.5.4 Qualitative Analysis . . . 19
2.6 Conclusion . . . 19

3 OLAF: A Plug-and-Play Framework for Enhanced Multi-object Multi-part Scene Parsing . . . 21
3.1 Introduction . . . 21
3.2 Related Works . . . 22
3.3 Our Method (OLAF) . . . 24
3.3.1 Foreground and Edge masks as boundary cues . . . 24
3.3.2 Low-level Dense Feature Extractor (LDF) . . . 25
3.3.2.1 Incorporation of LDF on Different Architecture Types . . . 27

3.3.3 Weight Adaptation . . . 27
3.4 Experiments . . . 28
3.4.1 Datasets . . . 28
3.4.2 Evaluation Metrics . . . 29
3.4.3 Training Details . . . 29
3.5 Experimental Results . . . 29
3.5.1 Pascal-Part-58, 108 and 201 . . . 29
3.5.2 PartImageNet . . . 30
3.5.3 Ablation Studies . . . 30
3.5.3.1 Input channel baselines: . . . 32
3.5.3.2 Weight Adaptation: . . . 33
3.5.3.3 Effect of Mask Noise on Performance: . . . 35
3.5.4 Qualitative Results . . . 35
3.5.5 Computational Metrics . . . 35
3.5.6 Effect of input resolution . . . 35
3.5.7 Gain Analysis . . . 37
3.5.8 Observations and Limitations . . . 39
3.5.8.1 Observations . . . 39
3.5.8.2 Limitations . . . 39
3.6 Conclusion . . . 39

4 Conclusions . . . 40

5 Future Work . . . 43

Bibliography . . . 44

List of Figures

Figure Page

1.1 Figure depicting scene parsing at the object level alongside a visual representation of scene parsing at the part level. 1

2.1 An overview diagram of our FLOAT framework (Sec. 2.3). Given an input image I_{rgb}, an object-level semantic segmentation network M_{obj} (, in blue) generates object prediction map (S_o). Two decoders (in orange) produce object category grouped part-level prediction maps for `animate' ($S_a$) and `inanimate' objects (S_i) in the scene. Another decoder (in red) produces part-attribute grouped prediction maps for `left-right' ($S_{lr}$) and `front-back' (S_{fb}). At inference time (shown by dotted lines), outputs from the decoders are merged in a top-down manner. The resulting prediction is further re ned using the IZR technique (see Fig. 2.2) to obtain the nal segmentation map (S_p). 8

2.2 An overview of Inference-time Zoom Re nement (IZR) - Sec. 2.3.4. During inference, predictions from the object-level network M_{obj} are used to obtain padded bounding boxes for scene objects (B). The corresponding object crops (C) are processed by the factorized network (F, Sec. 2.3). The resulting label maps (D) are composited to generate S_p, the nal re ned part segmentation map (E). Notice the improvement in segmentation quality relative to the part label map without IZR (included for comparison). . . 9

2.3 12

2.4 Toy example comparing mIOU and sqIOU with two images from *toy-person* category containing parts *head* and *torso*. `Red' and 'blue' represent ground-truth, `pink' and 'green' represent prediction overlap areas. mIOU fails to re ect the bad segmentation of head in image I_2 while sqIOU is fairer. 13

2.5 An illustration of labelling granularity in different versions of the Pascal-Part dataset. Pascal-Part-108 adds smaller parts (e.g., *eyes*, *ears*) to Pascal-Part-58. Our newly introduced Pascal-Part-201 further adds directional information to parts as appropriate (e.g., {`left',`right'} to *eyes*, *ears*; {`front',`back'} to *legs*). 13

2.6 Qualitative comparison on Pascal-Part-201. We observe that FLOAT gets small objects parts – *person* in the upper image, *cat* in the middle image. FLOAT also gets the left-right and front-back correct *leg(s)* of *dog* and *cat*, *side* of *car*, *wheel* of *bike*. 17

3.1 The segmentation results for state-of-the-art approach FLOAT [91] and its limitations can be seen in the second column. In the first row, FLOAT completely fails to identify TV Frame and TV Screen. In the second row, FLOAT fails to capture the edge partition between Car-Body, Car-Tire and also between Car-Body, Car-Window. The third column shows results by incorporating our plug-and-play approach OLAF into FLOAT, leading to significantly improved object and part segmentation results. 22
3.2 The recipe for OLAF, our plug-and-play framework for enhanced multi-object multi-part scene parsing. Augment RGB input with object-based channels (fg/bg, boundary edges) obtained from frozen pre-trained models M_O (M_E). Use Low-level Dense Feature guidance from segmentation encoder (LDF, shaded green). Employ targeted weight adaptation for stable optimization with augmented input. We show that following this recipe leads to significant gains (up to 10 mIoU) across multiple architectures and across multiple challenging datasets. 23
3.3 Illustration of OLAF's architectural integration with FLOAT [91]. FLOAT's components are tagged with $\mathcal{F}$. The object masks from output S_o of object segmentation network M_o are merged to obtain the foreground map I_{fg}. The output of edge generation network M_e is thresholded and filtered using I_{fg} to obtain edge map I_{edge}. The obtained maps are stacked with input image I_c to obtain the 5-channel input I_p^0 for the part segmentation network $\mathcal{F}_k$. The interface for LDF with encoder E_{part} and its architecture (top right) are also shown. A similar integration of OLAF also exists for U-Net style and Transformer style architectures. 25
3.4 LDF (Sec. 3.3.2) consistently improves the performance of small/thin parts. As shown in Row I, FLOAT [91] with LDF successfully predicts Aeroplane-Body while FLOAT fails to do so adequately. Similar results can be seen in Row II, where LDF successfully predicts Car-Light which FLOAT completely misses. 26
3.5 Incorporation of "LDF" in U-Net-like architectures 27
3.6 Incorporation of "LDF" in Transformer-like architectures 28
3.7 Qualitative comparison on Pascal-Part-201. OLAF consistently improves the performance of previous methods (BSANet [129], FLOAT [91]). This is especially seen for small parts as shown in Row 1 (eyes, ears, nose and right-front-leg), Row 2 (eye, ears, nose, mouth and tail) and occluded parts as shown in Row 3 (parts of motorbike). 32
3.8 Robustness of our approach to mask noise. The plot depicts mIoU scores as a function of noise added to the ground-truth binary fg/bg and edge masks of different datasets. Noise levels of 0%, 5%, 10%, 20% and 30% were introduced by randomly flipping the corresponding percentage of pixel labels in the masks. Note that 0% noise level corresponds to using ground-truth masks during training and inference. OLAF achieves mIoU scores of 64.3 (part-58), 51.5 (part-108) and 40.9 (part-201), using HED edges which correspond to noise levels of 18.97, 26.51 and 20.44, respectively (represented by x's in the figure). 33
3.9 Qualitative comparison on Pascal-Part-108. OLAF consistently improves the performance of previous methods (BSANet [129], FLOAT [91]). This is especially seen for small parts as shown in Row 1 (eyes and tail), Row 2 (eyes, ears, front-paw and tail) and Row 3 (wheel, headlight and very tiny parts of humans). 34
3.10 Qualitative comparison for images from PartImageNet [41]. OLAF consistently improves the segmentation quality, especially for the harder small parts of objects. 36
3.11 Results depicting the limitations of OLAF 37

3.12 Gain in mIoU vs Classes across all Pascal-Part datasets. Choice of model is FLOAT (current SOTA across all Pascal-Part datasets) 38

List of Tables

Table Page

2.1 Category-wise results for Pascal-Part-201. FLOAT outperforms competing methods by large margins w.r.t mIOU (top) and sqIOU (bottom). 15

2.2 Results on Pascal-Part-58, Pascal-Part-108: FLOAT outperforms the baseline and other existing methods on mIOU and with a signi cant gap on sqIOU. Missing CO-Rank entries are due to incomplete of cial codebase and missing details in the paper. 17

2.3 Ablation study: Starting from baseline with no factorization at all, we see that systematically adding components of FLOAT pipeline noticeably improves segmentation quality. M_{part} is combined decoder for all part-level labels, $FLOAT = F + IZR$ (see Fig. 2.1) is the proposed model. RCZ stands for Random Crop Zoom (see Sec. 3.5.3). The * indicates separate decoders for `left/right' and `front/back'. `Output heads' – total number of output channels of a model. `No factorization' – parts are labelled with concatenated category and associated part name. `Object' – predicting object labels separately. . . . 18

2.4 Compute comparisons of FLOAT with previous methods. Train time is per epoch. Test time is per instance. (Batch size = 5). Total output heads for FLOAT given in brackets under method. Test time in brackets for FLOAT quotes time without IZR. 20

3.1 Pascal-Part-201 results (mIoU/mAvg - top table and sqIoU/sqAvg - bottom table). "+ O": augmented with OLAF, "†": with ViT-H backbone. 30

3.2 Pascal-Part-58&108 segmentation results. `+ O':augmented with our approach OLAF, `†': with ViT-H backbone. 31

3.3 Results on validation set of the PartImageNet [41] dataset. The backbone for each approach is speci ed separately. "+ O": augmented with our proposed method OLAF. 31

3.4 Ablation experiments on PASCAL-Part-201 (Sec. 3.5.3). We use FLOAT† as the base model. 31

3.5 Compute metrics for various methods with inclusion of OLAF ('+ O') and without (baseline) on Pascal-Parts-58. The batch size is 16. 36

3.6 FLOAT [91] with different train and inference time resolutions for Pascal-Parts-58. The underlined value is the default setup for FLOAT. OLAF's results are included for reference. 36

Chapter 1

Introduction

Figure 1.1 Figure depicting scene parsing at the object level alongside a visual representation of scene parsing at the part level.

Scene parsing is a complex problem that has long captured the attention of researchers. With the rapid development in computing technology, scene parsing has found applications in various elds, including autonomous driving, robotics, medical imaging, and augmented reality. Critical challenges in this eld include segmenting objects under extreme conditions, such as small-size segmentation, wild environments, and occlusion.

Parsing scenes solely based on objects offers only a high-level representation, which often falls short for tasks requiring intricate, ne-grained details. Part-level scene parsing (Figure. 1.1) aims to segment both the object and its constituent parts, providing a more comprehensive and detailed understanding of

the scene. This ne-grained segmentation is crucial in various applications. For instance, in robotics, part-level segmentation aids in object manipulation and interaction, while in augmented reality, it allows for more precise overlays and interactions. In autonomous driving, understanding the components of road signs and vehicles enhances decision-making processes. Similarly, in medical diagnosis, identifying speci c parts of anatomical structures is essential for detailed analysis.

In this work, we focus on advancing scene parsing techniques to achieve detailed scene parsing at the part level. Additionally, we analyzed the core challenges/limitations (Section 1.2) typically encountered in part-level scene parsing and proposed methods/modules (Section 1.3) to address these issues.

1.1 Multi-Object Multi-Part Scene Parsing

Multi-object multi-part segmentation is a challenging task that involves simultaneously segmenting multiple objects in an image while also segmenting their individual parts. This task goes beyond conventional object segmentation [124, 55, 122, 7, 96, 17, 5, 126], aiming to achieve a multi-granular understanding of the scene. Granular semantic detail is crucial for applications in robotics [109, 80], visual question answering [45], object interaction and modeling [28, 1], and other domains [3, 18, 25, 52], where comprehending both objects and their constituent parts is essential.

1.1.1 History and Popular Approaches:

Parsing object parts in real-world scenarios, involving the simultaneous detection of multiple object classes and precise segmentation of semantic parts within each class, poses challenges due to ambiguities at both class and part levels. Despite its signi cance, this complex problem remains inadequately addressed in current literature. Current approaches primarily address simpler variants, such as single-object part parsing [63, 60, 62, 82, 37, 100] or part parsing for objects with fewer or larger, more visible parts [40]. Recent methods [76, 129, 95] have been developed to speci cally tackle the more complex task of multi-object multi-part parsing. However, these methods face signi cant challenges/limitations.

1.1.1.1 Some of The Popular Approaches:

- BSANet [129] : Proposed leveraging part edge features to aid part segmentation models in learning intricate edge details alongside part segmentation. This approach signi cantly enhanced the model's ability to distinguish edge boundaries between objects and their parts. However, the accuracy of learned edges remains a limitation, constraining further improvements in part segmentation results.

- GMNet [76]: Tackle multi-object multi-part parsing by providing object-level feature guidance to the part segmentation network during optimization. The authors used the features from the last layer of an object segmentation network as guidance to the part segmentation decoder.

- Compositor [40]: The Authors rede ned object and part segmentation as an optimization problem, constructing a hierarchical feature representation that incorporates pixel, part, and object-level embeddings. This approach allowed them to address the part parsing through a bottom-up clustering strategy.

1.2 Challenges in Multi-Object Multi-Part Scene Parsing

- Long Tail Distribution of Part Instances: Parts within certain object categories often exhibit varying occurrences and appearances, posing a challenge for segmentation models to effectively generalize and segment less frequent or irregularly represented parts. This issue addressed by the factorization method in our paper, FLOAT.

- Segmentation of Extremely Small Parts: Accurately identifying and segmenting very small and intricate parts of objects in images is particularly challenging due to their size and potential similarity to surrounding textures or structures. The encoder in a segmentation model typically relies on downsampling, which often results in the loss of features related to these very small or thin parts. Both our papers, FLOAT and OLAF, address this issue effectively.

- Lack of Challenging Part-Parsing Datasets: Due to the complexity and tedium of annotations, there have been few challenging datasets for scene understanding at the part level. Our paper, FLOAT, introduced the Pascal-Part-201 dataset, the most challenging and comprehensive dataset for part parsing.

- Absence of Metrics Insensitive to Part Size Biases: The commonly used mIoU metric for evaluating segmentation models inherently favors larger objects or parts. In our paper, FLOAT, we propose the sqIoU metric, which more accurately quanti es the performance of part segmentation models by mitigating this bias.

- Failure to Capture Intricate Edge Details Between Parts/Objects: Segmentation models often struggle to precisely delineate the boundaries between objects and their constituent parts, which is crucial for accurate part segmentation. Our paper, OLAF, addresses this issue by incorporating edge masks into the multi-channel input, signi cantly improving boundary accuracy.

- Failure to Capture Basic Foreground-Background Separation in a Scene: Segmentation models often struggle to precisely distinguish between foreground and background, which is crucial for accurate part segmentation. Our paper, OLAF, addresses this issue by incorporating binary

foreground-background masks into the multi-channel input, signi cantly improving foreground accuracy.

1.3 Our Contributions

In this thesis, we address the challenging task of multi-object multi-part scene parsing through two novel frameworks: FLOAT and OLAF. These frameworks are designed to enhance semantic understanding by segmenting both objects and their constituent parts in images. Following are our proposed moethods/modules that are particularly designed to solve the challenges metioned in Section. 1.2

- Factorized Label Space Framework: We introduce FLOAT, which utilizes a factorized label space to group object parts based on shared attributes such as root and side components. This approach enhances scalability by reducing label complexity, thereby improving performance on infrequent part categories.
- Inference-Time Re nement (IZR): FLOAT incorporates IZR, an inference-time technique that re nes object part segmentations. By re ning preliminary label maps at different scales, IZR improves segmentation quality, particularly for smaller and intricate parts, without additional training requirements.
- Dataset Expansion (Pascal-Part-201): We expand the Pascal-Part dataset to include previously excluded part attributes, creating Pascal-Part-201. This comprehensive dataset facilitates more accurate evaluation metrics such as sqIOU and sqAvg, demonstrating FLOAT's superior performance in semantic segmentation.
- Augmented Input Representation: OLAF introduces a 5-channel input representation augmented with object and boundary cues. This enhancement enables more precise initial segmentation by guiding the model to focus on relevant parts from the outset.
- Weight Adaptation Technique: A novel weight adaptation technique ensures stable optimization of pre-trained backbones on the augmented input, enhancing the model's ability to handle complex part segmentation tasks effectively.
- Low-Level Dense Feature Guidance (LDF): OLAF incorporates the LDF module, providing critical low-level dense feature guidance for improved segmentation accuracy, particularly for small and intricate parts across various datasets and segmentation architectures.

1.4 Thesis Layout

Chapter (2) provides a comprehensive overview of FLOAT, highlighting its role in enhancing scene parsing models. This section illuminates how our framework leverages critical cues such as shared root

and side components to improve the robustness of scene parsing methodologies. It includes discussions on our experimental setup, results, and insights into improving model performance.

Chapter (3) offers an extensive overview of our approach, OLAF, detailing the challenges it addresses within the scene parsing domain, particularly at the part level. This section encompasses our experimental setup, results, and additional insights into our proposed methodology.

Chapter (4) concludes the thesis with nal remarks, summarizing our efforts in addressing scene parsing at the part level and outlining our contributions to the eld. This section also highlights publications stemming from our research group.

Chapter (5) explores future directions and potential avenues for researchers to further advance scene parsing solutions.

Chapter 2

FLOAT: Factorized Learning of Object Attributes for Improved Multi-object Multi-part Scene Parsing

2.1 Introduction

Semantic scene parsing is a foundational image understanding problem in the vision community [131, 125, 56, 117, 116, 123, 120]. Typically, the goal is to segment objects and "stuff" regions (e.g. road, background) in the scene. Multi-object multi-part parsing is a signi cantly more challenging variant which requires part-level segmentation of each scene object [129, 76, 95]. Compared to traditional object-level segmentation, semantic representations infused with ne-grained part-level knowledge can provide richer information for downstream reasoning tasks including visual question answering [46], perceptual concept learning [9], shape modelling [2, 29] and many others [26, 19, 4, 121, 94, 53].

For part-based object segmentation, some existing approaches tackle the simpler problem of single-object part parsing [35, 33, 101, 97, 38]. Although a few recent approaches have addressed multi-object multi-part parsing [129, 76, 95], they consider part labels to be independent and do not take advantage of intra/inter ontological relationships among objects and parts at label level. They also tend to perform poorly on smaller and infrequent parts/categories. To address these shortcomings, we propose FLOAT, a novel factorized label space framework for scalable multi-object multi-part parsing. Our approach is motivated by the following observations:

Observation #1: Object part names in datasets typically consist of a root component and a side component(s). Many object categories contain parts with the same root component. For example, the root component of `left front leg' found in horse, cow etc. and `right leg' found in person, is leg. Therefore, parts can be grouped based on their root component.

The example also suggests that object categories whose instances contain shared category-level attributes (e.g. "living things that move") are likely to contain same root components (such as leg). Using this criterion, some object categories (e.g. cow, person, bird) can be grouped as `animate'. Similarly, some categories (e.g. "rigid bodied") can be grouped as `inanimate'. As with the

`animate' group, `inanimate' group categories also share many part components (e.g. `wheel' in aeroplane, bicycle, car).

Observation #2: Similar to Observation #1, parts can also be grouped by side component – e.g. `front' is a side component of `front wheel' found in bike and `left front leg' in person.

Factoring the object/part label space in terms of these groups (`animate', `inanimate', `side') greatly reduces the effective number of output labels. In turn, this increases scalability in terms of object categories and part cardinality. The design choice (`factoring') also enables efficient data sharing when learning semantic representations for grouped parts and improves performance for infrequent classes.

A second key feature of our framework is IZR, an inference-time segmentation refinement technique. IZR transforms `zoomed in' versions of preliminary per-object label maps into refined counterparts which are finally composited back onto the segmentation canvas. Apart from the advantage of not requiring additional training, IZR is empirically superior to alternate inference-time schemes and significantly improves segmentation quality, especially for smaller objects/parts.

In existing works, results are reported on simplified, label-merged versions of the original dataset (Pascal-Part [19]). In our work, we incorporate previously excluded part attributes and other minor parts to create Pascal-Part-201, the most comprehensive and challenging version of Pascal-Part [19]. Along with the standard mean IOU (mIOU) and mAvg scores, we report sqIOU [50] and sqAvg – normalized segmentation quality measures which are less affected by spatial scale of objects and parts.

2.2 Related Works

Semantic segmentation is a broad area with intensive research. We do not attempt to summarize all approaches to enable focus on more directly relevant works. A common design pattern for semantic segmentation is the encoder-decoder setup [6, 127, 12, 13]. In particular, the baselines, existing approaches and our proposed approach all adopt the popular DeepLab architecture [12] for various components of the segmentation task pipeline.

Single-Object Multi-Part Parsing has been extensively explored. Existing approaches typically consider object category subsets such as persons [33, 58, 61, 83, 106, 107, 108, 128, 35, 64, 73, 70], animals [38, 97, 101] and vehicles [61, 83, 92, 68]. However, in this setting, most works assume a single object of interest per image.

Multi-object multi-part parsing is a relatively new and under studied problem [129, 76, 95]. The approaches of Zhao et al. [129] and Michieli et al. [76] tackle multi-object multi-part parsing by providing object-level feature guidance to the part segmentation network during optimization. Zhao et al. [129] additionally provides boundary-level awareness to features. Tan et al. [95] create a semantic co-ranking loss modelling intra and inter part relationships. Xiao et al. [110] introduce a composite dataset and an approach for predicting perceptual visual concepts in scenes. However, in contrast to our framework, these approaches report results on simplified (label-merged) versions of standard datasets and empirically exhibit inferior performance for smaller parts.

Figure 2.1 An overview diagram of our FLOAT framework (Sec. 2.3). Given an input image I, an object-level semantic segmentation network (M_{obj}, in blue) generates object prediction map (S_{obj}). Two decoders (in orange) produce object category grouped part-level prediction maps for `animate' ($S_{an}$) and `inanimate' objects (S_{in}) in the scene. Another decoder (in red) produces part-attribute grouped prediction maps for `left-right' ($S_{lr}$) and `front-back' (S_{fb}). At inference time (shown by dotted lines), outputs from the decoders are merged in a top-down manner. The resulting prediction is further re ned using the IZR technique (see Fig. 2.2) to obtain the nal segmentation map (S_{pp}).

Factorization: In machine vision applications, early works such as Zheng et al. [130] used factorial Conditional Random Field models to separately predict object category, coarse object labels and object attributes such as shape, material and surface type. Other works involve jointly learning object and attribute-related information as a separable latent representation [81] or using graph networks [79]. Misra et al. [78] propose a factorization over global object attributes and object classi ers to enable compositionality. Other works extend this idea to inter-object relationships, e.g. noun-preposition-noun triplets [75, 54, 46]. In all these works, a single global property of the object (e.g., material, texture, color, size, shape) is learnt jointly with the object category information. In their work on panoptic part segmentation, Geus et al. [22] conduct experiments involving two categories from Pascal-Part-58 with some parts grouped by semantic similarity. Graphonomy, a framework by Lin et al. [65] can span multiple datasets with a at label structure and requires a manually speci ed graph per category. Such rigid connectivity relationships are unsuitable for modelling highly articulated objects (e.g. animals) found in our setting. To the best of our knowledge, we are the rst to show that object parts can be factorized across diverse object categories at scale, and that such factorization signi cantly improves segmentation performance, in resonance with theories of visual recognition [8, 44].

Figure 2.2 An overview of Inference-time Zoom Re nement (IZR) - Sec. 2.3.4. During inference, predictions from the object-level network N_{obj} are used to obtain padded bounding boxes for scene objects (B). The corresponding object crops (C) are processed by the factorized network F_k (Sec. 2.3). The resulting label maps (D) are composited to generate S_p, the nal re ned part segmentation map (E). Notice the improvement in segmentation quality relative to the part label map without IZR (included for comparison).

Zooming in on image regions using bounding boxes generated by attention maps [104] and reinforcement learning policies [27, 113] have been found to improve detection and segmentation. Other works use the technique on object instances for video interpolation [119] and on part instances for object parsing [106]. Porzi et al. [87] use zoomed in crops based on object classes for improving panoptic segmentation of high resolution images. Similar to the latter set of approaches, FLOAT also employs zooming in on object regions. However, our zoom-based re nement does not require any extra training and can be directly used during inference for improved performance.

2.3 Our framework (FLOAT)

As mentioned earlier, FLOAT's design leverages the shared-attribute groups that naturally exist within object categories (`animate', `inanimate') and part attributes (`left', `right', `front', `back') - see Fig. 2.1. The sections that follow describe how we operationalize the idea. Although our approach is general in nature, we use object categories and part names from the Pascal-Part dataset [19] for ease of understanding.

2.3.1 Relabeling images with factored labels

The original Pascal-Part dataset contains object and part level label maps. We re-label or partition these maps to obtain ve new label groups as described below.

object: The label set for this group comprises unique object category labels. For example, S_{obj} in Fig. 2.1 is a label map from this group containing person and bicycle objects.

animate: For this group, the label set comprises root components of part labels from the object categories bird, cat, cow, cat, dog, horse, person, sheep. The part labels are pooled across all object categories. For example, a single label leg covers all corresponding part instances from all objects in the `animate' group. This can also be seen in $S_a$ in Fig. 2.1 – the left foot and right foot of person are color-coded the same (`orange') and assigned the common label foot.

inanimate: The label set comprises root components of part labels from aeroplane, bicycle, bottle, bus, car, motorbike, pottedplant, train, tv. Note that (i) these categories are disjoint from the `animate' group (see $S_i$ in Fig. 2.1) (ii) the part label pooling mentioned for `animate' is applicable here as well.

side: In this case, two disjoint label groups exist. One group comprises all part labels which have the words `left' or `right' in their name (e.g. left hand, right wing). Label map regions whose part labels contain `left'/`right' are considered seed pixels for a flood-fill style procedure which produces corresponding `left'/`right' label maps (e.g. S_{lr} in Fig. 2.1). The same procedure is used for the label groups which have the words `front' or `back' in their name (see S_{fb} in Fig. 2.1).

Broadly, object parts from living things that move are in the `animate' group while other parts, typically from rigidly shaped non-living things, are in the `inanimate' group. As mentioned before, such grouping enables data-efficient representation learning for common parts (e.g. in `animate' group). A similar reasoning holds for `side' directional grouping (`left', `right'; `front', `back').

2.3.2 Factorized semantic segmentation architecture

We configure the segmentation architecture to output the factorized label maps described in previous section. As Fig. 2.1 shows, we employ two semantic segmentation networks, one for object-level and other for part-level label maps. The object-level network (Mk_{obj}) outputs the object prediction map (S_o). The part-level network consists of a shared encoder (E_{part}), and three decoders: the `animate' decoder ($D_{animate}$) which outputs the `animate' label maps (S_a), the `inanimate' decoder ($D_{inanimate}$) which outputs the `inanimate' label maps (S_i). The `side' decoder ($D_{side}$) outputs the `left/right' (S_{lr}) and `front/back' (S_{fb}) label maps. The outputs from the object-level network (S_o) and part-level network (S_i; S_a; S_{lr}; S_{fb}) are merged at inference time. We describe this merging process next.

2.3.3 Top-Down Merge

To combine the factorized label maps output by segmentation architecture (see Fig. 2.1), we adopt a top-down merging strategy. For each object (e.g. bicycle) in the object prediction map (S_o), we examine the labels of corresponding pixel locations in the part-level label maps. Depending on the type of object (`animate' or `inanimate'), the corresponding label regions are copied to the scene-level prediction canvas. (e.g. for bicycle, the considered labels in S_i would be wheel, chainwheel, handlebar, headlight, saddle). Similarly, the object-level map's pixel locations are referenced

from `side' label maps {`left',`right'} - $S_{lr}$, {`front',`back'} - S_{fb}). In case of con icts, the prediction defaults to background. The corresponding label regions are copied to the scene prediction canvas.

The owchart in the Fig. 2.3 describes the "Top Down Merge" algorithm per pixel to obtain the nal label for that pixel (aggregation across the image gives the nal prediction). As described in the paper, each label consists of an object, a root part component and side component(s). For FLOAT, these are determined separately and merged to obtain the nal label at each pixel. For each pixel :

1. We obtain the object category predicted. We now have an "object" label.
2. Choose the part from the animate part map or the inanimate part map depending on the object category. We now have an "object part" label.
3. We now add side components :
   (a) Animate:
      i. For animate categories, a part can have both left/right and front/back labels.
      ii. Depending on what side components the "object part" needs to match the original label space, the same are added from the S_{lr} and S_{fb} side maps.
      iii. To make sure each pixel has a left/right and a front/back label, while taking the softmax, we ignore the background category prediction.
   (b) Inanimate:
      i. For animate categories, a part can have only one of left/right/front/back labels.
      ii. We compute the combined Left-Right-Front-Back (LRFB) map by combining the Left-Right (S_{lr}) and Front-Back (S_{fb}) maps using con dence (softmax) values.
      iii. If the "object part" needs the side component, the same is added from the LRFB map.

Hence, we get all components required from predicting the nal label for each pixel: "Object L/R F/B Part" for animate and "Object L/R/F/B Part" for inanimate objects.

In the next section, we describe how the resulting prediction map is re ned using a per-object `zooming' technique.

2.3.4 Inference-time Zoom Re nement (IZR)

The Inference-time Zoom Re nement (IZR) technique improves segmentation quality by `zooming' into each scene object. As the rst step, the input image I is processed by the object-level network M_{obj} to obtain object-level map (see A in Fig. 2.2). The bounding box corresponding to each object component is then padded so that the object is centered and aspect ratio is preserved (see B in Fig. 2.2). Image crops corresponding to the padded bounding box extents are then obtained (C). Note that the padding enables scene context to be included for each cropped object and also helps account for inaccuracies in the object map prediction. The cropped object images are then processed by FLOAT's factorized

Figure 2.3

network F to obtain the corresponding part-level label maps D_p. These label maps are then composited to generate the final refined segmentation map E_p. In the next two sections, we describe the optimizer formulation for the networks in FLOAT and implementation details.

2.3.5 Optimization

We train the object model M_{obj} (Sec. 2.3.2) using the standard per-pixel cross-entropy loss. For training the part-level model, we use a combination of cross-entropy loss (L_{CE}) and graph matching loss (L_{GM}) [76]. The cross-entropy loss is applied to each of the 4 output part-level maps $S_{fa}; S_{ba}; S_{lr}; S_{fb}$ (see Fig. 2.1).

The graph matching loss [76] captures proximity relationships between part pairs within the map and scores the matching of these pairs between the ground truth and the predicted map. The degree of

Figure 2.4 Toy example comparing mIOU and sqIOU with two images from *toy-person* category containing parts *head* and *torso*. `Red' and 'blue' represent ground-truth, `pink' and 'green' represent prediction overlap areas. mIOU fails to re ect the bad segmentation of head in image I_2 while sqIOU is fairer.

Figure 2.5 An illustration of labelling granularity in different versions of the Pascal-Part dataset. Pascal-Part-108 adds smaller parts (e.g. *eyes*, *ears*) to Pascal-Part-58. Our newly introduced Pascal-Part-201 further adds directional information to parts as appropriate (e.g. {`left', `right'} to *eyes*, *ears*; {`front', `back'} to *legs*).

proximity between a part pair is represented by the number of pixels in one part situated T pixels or less from the other part, where T is an empirically set threshold. For ef ciency, the pairwise proximity map is approximated by dilating each part mask by $T=2$ and computing the intersecting region. The ground truth proximity map $\mathbf{M}^{GT}$ (and similarly predicted map $\mathbf{M}^{pred}$) is formally de ned as: $m^{GT}_{i;j} = |\{s \in \mathcal{D}(p^{GT}_i) \cap \mathcal{D}(p^{GT}_j)\}|$ where $m^{GT}_{i;j}$ is the proximity between the ith and jth parts, p_i; p_j are the respective part masks, s is a generic pixel, $\mathcal{D}$ is morphological 2D dilation operator and $|\cdot|$ is the cardinality of the given set. A row-wise normalization is applied to the proximity matrix: $\mathbf{M}^{GT}_{[i;:]} = \mathbf{M}^{GT}_{[i;:]} / \|\mathbf{M}^{GT}_{[i;:]}\|_2$. The graph matching loss L_{GM} is computed as the Frobenius norm between the two adjacency matrices: $L_{GM} = \|\mathbf{M}^{GT} - \mathbf{M}^{pred}\|_F$.

Additionally, for the `animate' and `inanimate' branches, a composite foreground-background binary cross-entropy loss serves as extra guidance. The loss for the part level network is a weighted combination of the losses for all part branches: $L^{part} = L^{anim} + L^{inanim} + L^{side}$, where $L^{anim} = L^{anim}_{CE} + \lambda_{GM} L^{anim}_{GM}$.

2.3.6 Implementation and Training Details

For fair comparison with previous works [129, 76, 95], we employ the DeepLab-v3 [12] architecture with a ImageNet pre-trained ResNet-101 [42] as the encoder (backbone) and follow the same training scheme and augmentations. During training, images are randomly left-right ipped and scaled between 0.5 and 2 times the original resolution with bilinear interpolation. The results at testing stage are reported at the original image resolution. The threshold T employed for proximity matrix (Sec. 2.3.5) is empirically set to 4. The model is trained for 40K steps with the base learning rate set to $7 \cdot 10^{-3}$ which is decreased with a polynomial decay rule with power 0.9. We employ weight decay regularization of 10^{-4}. We use a batch size of 16 images and use $_{GM} = 0.1$ for weighting graph matching loss relative to the cross-entropy loss. We use 2 NVIDIA A100 GPUs each with 40GB GPU memory to train our models, and for experiments. Full computational and memory requirement can be found in Table. 3.5.

2.4 Datasets and Evaluation Metrics

Pascal-Part: For experiments, we use the Pascal-Part [19] which is currently the largest multi-object multi-part parsing dataset. It contains 10,103 variable-sized images with pixel-level part annotations on the 20 Pascal VOC2010 [31] semantic object classes (plus the background class). We use the original split from Pascal-Part with 4998 images for training and 5105 images in the publicly provided validation set for testing.

Pascal-Part-58/108: For comparison with previous work, we use the datasets Pascal-Part-58 [129] and Pascal-Part-108 [76] which contain 58 and 108 part classes respectively. Both the Pascal-Part variants simplify the original semantic classes by grouping some parts together, and contain 58 and 108 part classes respectively. Pascal-Part-58 mostly contains large parts of objects such as head, torso, leg etc. for animals and body, wheel etc. for non-living objects. Pascal-Part-108 is more challenging and additionally contains relatively smaller parts (e.g. eye, neck, foot etc. for animals and roof, door etc. for non-living objects).

Pascal-Part-201: We incorporate part attributes (`left', `right', `front', `back', `upper', `lower') and other minor parts (e.g. eyebrow) excluded in both the mentioned variants (58/108), to create the most comprehensive and challenging version of the dataset containing 201 parts which we dub Pascal-Part-201. We observed that the original part labelling scheme in Pascal-Part leaves out large chunks of an object's pixels unlabelled for the bike, motorbike and tv categories which lead to disconnected objects. To address this, we add a body part annotation for bike, motorbike, and a frame part for tv. An example illustrating the differences in part labelling and granularity of the Pascal-Part variants can be seen in Fig. 2.5.

Model	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV	mIOU	mAvg
Baseline	91.0	31.6	47.7	24.3	56.7	46.4	31.0	36.7	24.2	35.6	17.5	38.6	27.3	20.7	38.0	26.9	50.8	13.3	42.1	14.2	76.5	37.6	36.8
GMNet[76]	90.8	26.6	33.1	21.2	55.0	43.5	24.6	27.5	21.7	35.5	15.1	40.3	25.0	17.5	31.9	21.9	44.2	11.9	43.3	14.2	72.5	33.2	33.2
BSANet[129]	91.2	34.6	41.7	27.9	61.2	51.7	34.1	38.1	26.1	35.4	24.0	43.6	28.4	23.0	37.4	54.7	74.3	40.4	17.8	59.4	28.5		38.7
FLOAT	92.5	36.7	49.7	34.4	75.3	51.4	35.8	42.0	37.8	59.6	35.5	58.2	41.0	34.0	40.2	52.0	88.5	69.0	15.1	56.1	37.1		46.9

	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV	sqIOU	sqAvg
Baseline	89.6	28.9	39.3	17.1	57.4	32.3	27.1	26.0	20.5	39.8	14.8	34.7	22.7	17.2	31.5	19.2	34.9	10.8	52.6	14.2	41.5	33.8	32.6
GMNet[76]	89.4	20.7	23.5	12.6	53.1	25.8	19.3	17.2	18.1	38.2	11.2	35.2	15.9	14.2	25.4	13.8	26.9	8.5	52.0	13.8	36.4	26.9	27.7
BSANet[129]	89.9	30.7	33.5	18.6	60.2	31.2	29.2	26.4	21.2	37.8	17.5	38.0	22.3	17.8	31.2	18.2	33.6	10.1	37.4	52.4	22.1		32.8
FLOAT	90.8	32.5	41.8	24.5	63.9	36.1	30.4	29.9	33.0	50.8	28.1	47.6	35.6	26.1	33.6	29.9	34.5	20.1	63.6	50.8	29.6		39.5

Table 2.1 Category-wise results for Pascal-Part-201. FLOAT outperforms competing methods by large margins w.r.t mIOU (top) and sqIOU (bottom).

2.4.1 Evaluation Metrics

For performance evaluation, we use two versions of Intersection over Union (IOU) metric. We first describe mIOU and mAvg, the standard segmentation quality metrics reported for the problem setting. We then describe balanced variants of these metrics – sqIOU and sqAvg.

mIOU: Let $Pred_p^j$ and GT_p^j be the prediction and ground truth respectively for the part p in the jth image I_j. Suppose the dataset contains N images. The mIOU for the part ($mIOU_p$) is calculated as:

$$mIOU_p = \frac{\sum_{j=1}^{N} (Pred_p^j \cap GT_p^j) \cdot I[p \in I_j]}{\sum_{j=1}^{N} (Pred_p^j \cup GT_p^j) \cdot I[p \in I_j]} \tag{2.1}$$

where $I[:]$ is the indicator function (i.e. summation is performed only for images where part p is present). The mIOU for the dataset is then calculated as $mIOU = \sum_p mIOU_p / N_p$, where N_p is the number of part categories (classes) in the dataset (58/108/201).

mAvg: The mIOU score for an object category is the average of its per-part scores, $mIOU_c = \sum_p mIOU_p / N_c$ where N_c is the number of unique part labels in object category c. Finally, mAvg is calculated as $mAvg = (\sum_c mIOU_c)/C$, where C is the number of object categories (21 for Pascal-Part datasets).

sqIOU: This is a modified version of Segmentation Quality (SQ) metric [50] tailored for semantic segmentation. The sqIOU for the part p is calculated as:

$$\text{sqIOU}_p = \sum_{j=1}^{N} \Big(\underbrace{\frac{\text{Pred}_p^j \cap \text{GT}_p^j}{\text{Pred}_p^j \cup \text{GT}_p^j}}_{\text{IOU}(\text{Pred},\text{GT})_p^j} \cdot \mathbb{I}[p \in I_j] \Big) / N \tag{2.2}$$

The calculation for sqIOU and sqAvg is similar to that of mIOU. Due to their formulation, mIOU and mAvg [76, 129] tend to be dominated by contributions from biggest instances. In contrast, sqIOU and sqAvg weight parts of all sizes equally – compare Eqn. 2.1 and 2.2 and also see the toy example in Fig. 2.4. Therefore, sqIOU and sqAvg can be considered a more `fair' measure for segmentation quality.

2.5 Experimental Results

For evaluation, we compare the performance of FLOAT with BSANet [129], GMNet [76] and CO-Rank [95]. As a baseline, we train a DeepLab-v3 [12] model with independently paired object category and associated part names (e.g. cow left eye, cow right ear) as labels. BSANet and CO-Rank report results on Pascal-Part-58 while GMNet additionally reports results on Pascal-Part-108. We report results on all variants of the Pascal-Part dataset, including our newly introduced Pascal-Part-201. To enable comparison, we train GMNet and BSANet on our dataset, Pascal-Part-201. For evaluation, we employ the mIOU, mAvg and sqIOU, sqAvg metrics described previously (Sec. 2.4.1). In addition, we analyze the relative contribution of various components in FLOAT via ablation studies.

2.5.1 Pascal-Part-201

Table 3.1 shows the category-wise and overall performance on Pascal-Part-201. Overall, we see that FLOAT outperforms baselines and existing approaches by a signi cantly large margin. We obtain large gains of 10.8% on mIOU and 8.1% on sqIOU relative to the baseline. We outperform the next best method BSANet [129] by large margins of 8.6% on mIOU and 7.5% on sqIOU as well.

Empirically, we obtain signi cant sqIOU gains of 10%-30% on small parts – for left/right eye, left/right ear, left/right horn etc. of `animate' categories such as bird, cat, cow. For `inanimate' categories (e.g. bus, car, aeroplane), we obtain sqIOU improvements in the range of 5%-11% on small parts such as front/back plate, left/right wing. The performance improvement is also similarly substantial for most parts containing side components (`left/right' or `front/back').

Method	Dataset	mIOU	mAvg	sqIOU	sqAvg
Baseline	58	54.3	55.4	46.0	48.4
BSANet[129]		58.2	58.9	49.3	51.5
GMNet[76]		59.0	61.8	49.4	54.3
CO-Rank[95]		60.7	60.6	-	-
FLOAT		61.0	64.2	54.2	57.1
Baseline	108	41.3	43.6	32.2	36.1
BSANet[129]		45.9	48.4	36.6	41.0
GMNet[76]		45.8	50.5	35.8	41.9
FLOAT		48.0	53.0	40.5	45.6

Table 2.2 Results on Pascal-Part-58, Pascal-Part-108: FLOAT outperforms the baseline and other existing methods on mIOU and with a signi cant gap on sqIOU. Missing CO-Rank entries are due to incomplete of cial codebase and missing details in the paper.

Figure 2.6 Qualitative comparison on Pascal-Part-201. We observe that FLOAT gets small objects parts – person in the upper image, cat in the middle image. FLOAT also gets the left-right and front-back correct – leg(s) of dog and cat, side of car, wheel of bike.

Method	Dataset	Output Heads	No Factorization	Object	Part	Anim/Inanim	Side	Inference Augmentation	mIOU	sqIOU
Baseline	58	58	X				-		54.3	46.0
$M_{obj} + M_{part}$		45		X	X		-		60.7	51.5
F		45		X		X	-		60.9	51.7
FLOAT		45		X		X	-	IZR	61.0	54.2
Baseline	108	108	X				-		41.3	32.2
$M_{obj} + M_{part}$		68		X	X		-		46.1	36.7
F		68		X		X	-		47.8	38.4
FLOAT		68		X		X	-	IZR	48.0	40.5
Baseline	201	201	X						26.3	21.5
$M_{obj} + M_{part}$		119		X	X				29.1	22.8
$F \quad D_{side}$		119		X		X			31.3	24.1
F		80		X		X	X		36.9	27.8
F *		80		X		X	X *		36.9	27.6
F + RCZ		80		X		X	X	RCZ	36.6	28.0
FLOAT		80		X		X	X	IZR	37.1	29.6

Table 2.3 Ablation study: Starting from baseline with no factorization at all, we see that systematically adding components of FLOAT pipeline noticeably improves segmentation quality. M_{part} is combined decoder for all part-level labels, FLOAT = F + IZR (see Fig. 2.1) is the proposed model. RCZ stands for Random Crop Zoom (see Sec. 3.5.3). The * indicates separate decoders for `left/right' and `front/back'. `Output heads' – total number of output channels of a model. `No factorization' – parts are labelled with concatenated category and associated part name. `Object' – predicting object labels separately.

2.5.2 Pascal-Part-58 and Pascal-Part-108

We also show results on previously proposed datasets Pascal-Part-58 [129] and Pascal-Part-108 [76]. As shown in Table 2.2, FLOAT framework achieves the best performance on both these datasets. In

[1]Informally, an instance is deemed “big” if it is among the largest instances for an object part category by area.

terms of mIOU, we outperform CO-Rank [95] by 0.3% on Pascal-Part-58 and GMNet [76] by 2.0%. In terms of sqIOU, we outperform other methods by large margins as well – 4.8% over GMNet and 4.9% over BSANet. A similar trend is seen for Pascal-Part-108 with large improvements of 2.1% on mIOU and 3.9% on sqIOU over the next best method BSANet [129].

Overall, the results across existing and challenging new variants of Pascal-Part dataset demonstrate the strengths of our factorized label space setup. In particular, the increasing gains with increasing dataset complexity demonstrates the superior scaling capacity of the FLOAT framework.

2.5.3 Ablation Studies

We perform multiple experiments with ablative variant models of FLOAT to verify the effectiveness of our design choices. From the results in Table 2.3, we see that starting from baseline (rst row in each dataset variant), systematically adding components of FLOAT pipeline noticeably improves segmentation quality. The gains are most apparent for Pascal-Part-201 dataset, particularly when factorized components are included. From the last two rows, we also see that IZR is a superior choice compared to Random Crop Zoom (RCZ) - a variant which uses random crops whose cardinality matches the number of objects in the scene. Some part names in the original Pascal-Part dataset [19] contain the side component `upper/lower'. We attempted to train a FLOAT variant with these components as outputs of $D_{side}$ decoder. However, the model failed to converge. We hypothesize this is due to the drastically smaller quantum of training data compared to other side attributes, i.e. `left/right' and `front/back'.

2.5.4 Qualitative Analysis

Fig. 2.6 shows qualitative comparisons of our framework with existing approaches on Pascal-Part-201, re ecting the improvements gains we observe for mIOU and sqIOU metrics (Table 3.1). FLOAT is visually superior at segmenting smaller object parts – notice the signi cantly improved segmentation for parts in object categories person (rst row) and cat (second row). From the examples, we see that FLOAT is also better at learning directionality (`left/right', `front/back'). Some limitations of FLOAT include missing predictions for the smallest of parts (e.g. eye in people far from camera) and partial predictions for thin parts leading to disconnections.

2.6 Conclusion

FLOAT is a simple but effective framework for improving semantic segmentation performance in multi-object multi-part parsing. Our idea of factorized label space is a key contribution which fully takes advantage of label-level intra/inter ontological relationships among objects and parts. The factorization not only enables scalability in terms of both object categories and part labels, but also improves segmentation performance substantially. Another key contribution is our inference-time zoom. By focusing only on object-centric regions of interest, IZR ef ciently enhances segmentation quality without

Method	Dataset	Params(M)	Train Time (mins)	Test Time (secs)
BSANet	58	63.9	40.2	0.45
GMNet		124.9	33.6	0.49
FLOAT (45)		135.4	30.1	1.02 (0.55)
BSANet	108	63.9	43.7	0.72
GMNet		124.9	37.2	0.75
FLOAT (68)		135.4	34.8	1.38 (0.80)
BSANet	201	64.0	47.1	1.30
GMNet		124.9	40.3	1.35
FLOAT (80)		153.6	38.6	2.14 (1.43)

Table 2.4 Compute comparisons of FLOAT with previous methods. Train time is per epoch. Test time is per instance. (Batch size = 5). Total output heads for FLOAT given in brackets under method. Test time in brackets for FLOAT quotes time without IZR.

requiring explicit object feature guidance or other modi cations to the part network setup. Apart from our framework, we introduce a new variant of Pascal-Part called Pascal-Part-201 which constitutes the most challenging benchmark dataset for the problem. Our experimental evaluation, using fairer versions of existing measures, shows that FLOAT clearly outperforms existing state-of-the-art approaches for existing and newly introduced Pascal-Part variants. The gains from our framework increase with increased part and object dataset complexity, empirically supporting our assertion of FLOAT's scalability. Although presented in a 2D scene parsing setting, we expect ideas from FLOAT to be useful for the 3D scene parsing counterpart and in general, for scenarios with appropriately factorizable attributes.

Chapter 3

OLAF: A Plug-and-Play Framework for Enhanced Multi-object Multi-part Scene Parsing

3.1 Introduction

Multi-object multi-part segmentation is a challenging task that involves simultaneously segmenting multiple objects in an image while also segmenting their individual parts. The task goes beyond conventional object segmentation [124, 55, 122, 7, 96, 17, 5, 126] and aims to enable multi-granular scene understanding. The availability of granular semantic detail is crucial for applications in robotics [109, 80], visual question answering [45], object interaction and modeling [28, 1] and other domains [3, 18, 25, 52] where understanding the scene in terms of objects and their constituent parts is crucial.

Related approaches primarily address simpler variants such as single-object part parsing [63, 60, 62, 82, 37, 100] or part parsing for objects with fewer or visibly larger parts [40]. Some recent methods [91, 76, 129, 95] have been developed to speci cally tackle the more complex task of multi-object multi-part parsing. However, these suffer from three signi cant limitations:

Limitation 1: Foreground (union of object regions) is often incorrectly segmented, impacting the constituent part segmentation (Fig. 3.1, rst row). Limitation 2: Crucial boundary details between objects and parts are not captured accurately (3.1, second row). Limitation 3: Small and thin parts especially fail to be segmented (Fig. 3.4, Fig. 3.7).

To address Limitation 1 & 2, we rst obtain a plausible boundary edge mask using a pre-trained network. We use another pre-trained network to obtain preliminary object segmentation and combine the object label channels to obtain a binary foreground mask. These masks are included as additional channels to constitute the 5-channel (3 RGB + 2 masks) input to the reference segmentation network (see 1 in Fig. 3.2). The masks provide an inductive bias and guide the model to focus on relevant parts from the onset of training. We also propose a weight adaptation technique that enables the pre-trained segmentation encoder to process the new 5-channel input without destabilizing optim 3 ation (in Fig. 3.2).

Figure 3.1 The segmentation results for state-of-the-art approach FLOAT [91] and its limitations can be seen in the second column. In the first row, FLOAT completely fails to identify TV Frame and TV Screen. In the second row, FLOAT fails to capture the edge partition between Car-Body, Car-Tire and also between Car-Body, Car-Window. The third column shows results by incorporating our plug-and-play approach OLAF into FLOAT, leading to significantly improved object and part segmentation results.

To address Limitation 3 (i.e. small and thin parts), we introduce an encoder module termed Low-level Dense Feature (LDF) - see 2 in Fig. 3.2. This module, in conjunction with augmented input representation, provides low-level dense feature guidance enabling better segmentation, especially for small and thin parts.

3.2 Related Works

Single-Object Multi-Part Segmentation considers a single object and its constituent parts. Most works explore segmentation of non-rigid object categories such as person [21, 63, 71, 36, 59, 84, 60, 114, 34], some animals [100, 37, 98] and rigid object categories (e.g. vehicles [82, 93, 69, 62]). Some works have also examined open-world part segmentation [85, 105]. However, the single object condition is restrictive and not representative of general scenes which contain multiple objects from distinct categories and associated occlusions.

Multi-Object Multi-Part Segmentation has recently got increased attention due to its complexity and importance in downstream applications. There have been multiple different approaches to tackle this hard problem [91, 129, 76, 95]. The initial work of Zhao et al. [129] and the works of Micheli et al. [76, 77] leverage object and boundary awareness through auxiliary tasks and model design changes. The current state of the art approach (FLOAT [91]) employs label-space factorization to reduce the number

Figure 3.2 The recipe for OLAF, our plug-and-play framework for enhanced multi-object multi-part scene parsing. (1) Augment RGB input with object-based channels (fg/bg, boundary edges) obtained from frozen pre-trained models $M(M_O; M_E)$. (2) Use Low-level Dense Feature guidance from segmentation encoder (LDF, shaded green). (3) Employ targeted weight adaptation for stable optimization with augmented input. We show that following this recipe leads to signi cant gains (4.0 mIoU) across multiple architectures and across multiple challenging datasets.

of output heads. Typically, existing approaches do not attempt segmentation of very small/thin parts although annotations are available [91]. Beyond the popular datasets (Pascal-Parts-58), our approach enables improved performance for the harder variants (Pascal-Parts-108, Pascal-Parts-201).

Object level guidance and Boundary/edge awareness is typically present as an auxiliary task or in terms of guidance from an object network's features [129, 76, 77, 74, 10, 132, 90, 85]. For multi-object multi-part segmentation, Zhao et al. [129] add an auxiliary task of predicting object segments from the learned part segmentation representation. Michieli et al. [76] use the features from the last layer of an object segmentation network as guidance to the part segmentation decoder. In contrast, our work OLAF adds object segmentation and edge information directly as additional channels to the input which is observed to be more bene cial.

Low-level Feature Guidance has been used in previous works [39, 118, 86, 15] to enhance the performance of segmentation by leveraging low-level visual (spatial) cues. While some works incorporate skip-connections [14, 48], others utilize downsampling strategies along with skip-connections to obtain suf cient receptive eld for context capturing [118, 86]. While this strategy generally works well, it may not be suitable for tasks such as part segmentation because the information in low-level features is too coarse. In particular, downsampling compromises details of small or thin parts. By contrast, our