Hiroshima University

Doctoral Thesis

Enhanced Object Detection and Instance Segmentation Through Advanced Prior Information Integration in Deeplearning

(ディープラーニングにおける高度な事前情報統合による物体検出とインスタンスセグメンテーションの強化)

Author:
Shinji Uchinoura (D193064)

Supervisor:
Takio Kurita

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Information Engineering

March, 2024

Declaration of Authorship

I, Shinji UCHINOURA (D193064), declare that this thesis titled, "Enhanced Object Detection and Instance Segmentation Through Advanced Prior Information Integration in Deeplearning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Shinji Uchinoura

Date: March 2024

HIROSHIMA UNIVERSITY

Abstract

Graduate School of Engineering
Department of Information Engineering

Doctor of Philosophy

Enhanced Object Detection and Instance Segmentation Through Advanced Prior Information Integration in Deeplearning

by Shinji UCHINOURA (D193064)

Object detection and instance segmentation are foundational tasks in the field of computer vision, and they have made significant strides, particularly with the proliferation of deep learning. While these tasks find a multitude of practical applications in the real world, they are not immune to the challenges of false recognitions and detections, which can have detrimental consequences. One of the contributing factors to this inconsistency is our belief that the solutions derived from the provided supervision and losses lead to suboptimal solutions that lack an understanding of real-world structures and physical constraints.

In response to this challenge, our research addresses this issue by augmenting the information provided to existing object detection and instance segmentation models. We believe that introducing additional information regarding the structural constraints and prior knowledge about the images can lead to closer-to-optimal solutions. The proposed techniques introduce novel forms of prior information, namely, "grid boundary information," "foreground-background information," and "class label boundary information."

The first piece of prior information, the grid boundary information, refers to the rule-based boundaries of grids that divide the entire image, which are adopted by state-of-the-art object detection techniques. We experimentally discovered that when these grid boundaries coincide with the center positions of detected objects, the discriminative performance significantly deteriorates. Therefore, we propose the necessity of instructing the model about the existence of these designer-specific grid boundaries. To address this issue, we introduced two techniques: a module for feature extraction from grid boundaries and data augmentation that involves shifting object centers parallel to the grid boundaries. Combining these methods enhances the model's shift-invariant features and contributes to improved generalization performance.

The second piece of prior information involves the proposal of a method to address the issue of class imbalance between foreground and background in object detection, which is a concern often faced by the field. While object detection and instance segmentation are similar tasks, combining them has not been actively explored until now. Given the recent feasibility of zero-shot learning in segmentation, we believe that it is a viable approach to utilize segmentation information for other tasks. Consequently, we introduce SODet, a two-step learning framework. In the first step, we train a model for instance segmentation, enabling the construction of a model capable of generating background masks. The second step involves integrating the generated background masks into the input images and conducting retraining. This innovative framework allows us to leverage prior information about foreground and background, leading to finer-grained discrimination capabilities and performance improvements across multiple benchmarks.

The third aspect of prior information pertains to introducing a regularization method in the context of instance segmentation, which involves providing boundary information of segmented instances as penalties. This approach stands out by

addressing the common trend in conventional methods, where errors are predominantly considered at the pixel level. We observed that most existing techniques do not adequately account for the spatial structural information formed among adjacent pixels at object boundaries, whether it's the boundary with the background or with other objects. In our proposed approach, we aim to incorporate this spatial structure information into the segmentation process. Specifically, the spatial structure of the image of which the image of the teacher mask is composed introduces a regularisation mechanism that ensures that if neighboring pixels belong to the same class, they also belong to the same class as the neighboring pixels on the predicted image (and vice versa). This mechanism is derived from the spatial structure of the image formed by the ground truth mask, ensuring consistency in class labeling between adjacent pixels in the predicted image. This innovative regularization technique yields a more distinct mask, particularly at the boundary regions between instances and the background. As a result, it contributes to an enhancement in the quality of the output generated by the instance segmentation process.

Acknowledgements

I am immensely grateful to Professor Takio Kurita of the Graduate School of Advanced Science and Engineering at Hiroshima University for his invaluable mentorship and unwavering support throughout my doctoral process. Thanks to Professor Takio Kurita, I, who had previously been working in the software development industry, was able to step into the world of research. The time spent with him allowed me to experience the depth and joy of research. His extensive expertise in pattern recognition provided me with a profound understanding of the research process and its fundamental allure. He not only shared his deep knowledge generously but also provided me with frequent guidance, demonstrating both a keen understanding of the field and a genuine dedication to nurturing my academic growth.

I would also like to extend my gratitude to Professor Junichi Miyao, who provided insightful comments and constructive feedback on my research, and Professor Bisser R. Raytchev, who contributed to enhancing the quality of my thesis with his valuable insights. Their wisdom greatly enriched my work.

I am also deeply thankful to my colleagues at the Pattern Recognition Laboratory at Hiroshima University. The exchange of ideas and daily experiences with them has not only helped me develop as a researcher but also as an individual.

During my doctoral program, I had unwavering support and encouragement from my friends, colleagues and family, who were my pillars of strength and unwavering support. I extend my deepest appreciation to them.

Furthermore, my wife, who not only supported my research activities but also took on the responsibility of childcare and housework, has enabled me to concentrate on my academic work in a dual-earner household. Her unwavering support and sacrifice have been an essential foundation for my academic path, for which I am sincerely grateful.

Last but not least, I am grateful to all those who participated in my research, dedicating their time and knowledge willingly. This thesis would not have been possible without the support and assistance of all these individuals, and for that, I am profoundly thankful.

Contents

Declaration of Authorship iii

Abstract vi

Acknowledgements ix

1 Introduction 1

1.1 Overview of Object Detection and Instance Segmentation 1

1.1.1 Overview of Object Detection 1

1.1.2 Overview of Instance Segmentation 1

1.1.3 Differences of Each Task 2

Ground-Truth Information 2

Prediction Resolution 2

1.2 The Emergence of Real-World Object Detection Applications 3

1.2.1 Survellance Camera Systems 3

1.2.2 Medical Imaging 3

1.2.3 Automotice Safety Systems 3

1.3 Risk of malfunction 4

1.3.1 Surveillance and Security: Detecting Intruders or False Alarms? 4

1.3.2 Medical Imaging: The Consequences of False Positives and Negatives 4

1.3.3 Automotive Safety: Avoiding Catastrophic Accidents 4

1.4 Addressing the Challenge 5

1.4.1 Is object detection weak against translations? 5

1.4.2 Can We Address Class Imbalance in Object Detection by Questing for Balanced Foregrounds? 6

1.4.3 Why does the segmentation mask fail at the boundary? 8

1.5 structure of the thesis 9

1.5.1 Related works 9

1.5.2 Improved head and data augmentation to reduce artifacts at grid boundaries in object detection 9

1.5.3 An Object Detection Method Using Probability Maps for Instance Segmentation to Mask Background 9

1.5.4 Graph Laplacian Regularization based on the Differences of Neighboring Pixels for Conditional Convolutions for Instance Segmentation 10

1.5.5 Conclusion 10

2 Evolution of Object Detection and Instance Segmentation 11
2.1 Object Detection Methods 11
2.1.1 Traditional Detectors 11
2.1.2 CNN-based Detectors 12
2.1.3 Two Stage Methods 13
Single Stage Methods 15
2.1.4 Anchor Free Detectors 17
2.2 Instance Segmentation Methods 18
2.3 Loss Function for Object Detection and Instance Segmentation 22
2.3.1 Classification Loss 22
2.3.2 Bounding Box Regression Loss 23
2.3.3 Mask Loss 24
2.4 Datasets for Object Detection and Instance Segmentation 25
2.4.1 MS-COCO 25
2.4.2 CityScapes 25
2.4.3 Evaluation Metric 26

3 Improved head and data augmentation to reduce artifacts at grid boundaries in object detection 29
3.1 Problems at Grid Boundaries 29
3.1.1 What is the Grid? 29
3.1.2 What Is the Problem? 30
3.1.3 Previous Methods for Shift Invariance 30
Grid-Based One-Stage Object Detector 30
Shift Invariance and Equivalence 31
SwinTransformer 31
3.1.4 Reducing Class Scores in Object Detection at Artefactual Grid Boundaries 33
3.2 Proposed Method 35
3.2.1 Overview of Our Proposed Method 35
3.2.2 Network Architecture 36
3.2.3 Sub-Grid Feature Extractor 37
3.2.4 Auxiliary Loss 38
3.2.5 Grid-Aware Data Augmentation 38
3.3 Experiments 39
3.3.1 Dataset and evaluation protocol 39
3.3.2 Experimental settings 39
3.3.3 Comparison With Baseline Method 40
3.3.4 Impact on Grid Boundaries 41
3.3.5 Comparison with the Standard Data Augmentation Methods . 42
3.3.6 Quantitative evaluation on grid boundaries 42

3.3.7 Ablation Study . . . 43
Grid-Aware Data Augmentation . . . 43
SGFEM . . . 44
Auxiliary Learning . . . 44
3.3.8 Experimental Application to Yolox . . . 45
Reproducibility Study in Yolox . . . 45
Network Modification Details . . . 46
Data Augmentation Improvement . . . 47
Experimental Settings . . . 47
Experimental results in YOLOX . . . 48
3.3.9 Experimental Application to the Faster RCNN method . . . 48

4 An Object Detection Method Using Probability Maps for Instance Segmentation to Mask Background 51
4.1 Problem of Imbalance Between Foreground and Background . . . 51
4.2 Can We Further Bridge the Imbalance Gap Between Foreground and Background? . . . 53
4.2.1 Uncertainties and Limitations of Banding Box Information . . . 53
4.2.2 The Evolution of Instance Segmentation Techniques . . . 53
4.2.3 Overview of Our Proposed Method . . . 54
4.3 Proposed Method . . . 55
4.3.1 Mask Functions . . . 55
4.3.2 SODet Overview . . . 55
4.3.3 Network Architecture . . . 56
Feature Extractor . . . 56
Object Detection Branch . . . 56
Instance Segmentation Branch . . . 57
4.3.4 Loss Function . . . 57
4.3.5 Probability Map . . . 57
4.4 Experiments . . . 59
4.4.1 Dataset . . . 59
4.4.2 Implementation Details . . . 59
4.4.3 Comparison of Mask Functions . . . 59
4.4.4 Comparison of proposed method and state-of-the-art methods . 60
4.4.5 Qualitative Evaluation . . . 61
4.4.6 Quantitative Evaluation . . . 62
4.4.7 Comparison on the COCO dataset . . . 63

5 Graph Laplacian Regularization based on the Differences of Neighboring Pixels for Conditional Convolutions for Instance Segmentation 67
5.1 Blurring of Instance Masks . . . 67
5.1.1 Developments and Challenges in Instance Segmentation . . . 67

5.1.2 Spatial Regularization for Improved Instance Segmentation . . 68
5.1.3 Graph-based regularization 68
5.2 Proposed Method 69
5.2.1 Formula as a Graph Laplacian Matrix 71
5.2.2 Architecture 72
5.2.3 Loss function 73
5.3 Experiments 74
5.3.1 Implementation Details 74
5.3.2 Results 75
Comparison on COCO Instance Segmentation 75
Comparison on Cityscapes Instance Segmentation 76
5.3.3 Qualitative Results 76
5.3.4 Ablation Study 78

6 Conclusion **81**
6.1 Summary 81
6.2 Future Works 82

Bibliography **85**

List of Figures

1.1 A sample annotated image from the COCO dataset 2
1.2 Qualitative result of shift variance 6
1.3 Number of background and foreground label assignments in COCO and CityScapes 7
1.4 An example of mask failure at boundary 8

2.1 Basic network architecture of modern object detector 13
2.2 Evolution of the two-stage method 15
2.3 FCOS network architecture 18
2.4 Mask R-CNN network architecture 19
2.5 SOLO network architecture 21
2.6 Sample images from the COCO dataset with annotation 26
2.7 Sample images from the CityScapes dataset with annotation 26

3.1 Effect of image shift on class score 33
3.2 Location relationship between the grid boundary and object center . . 35
3.3 Overview of the network architecture of the proposed method 36
3.4 Illustration of SGFEM 37
3.5 Comparison of robustness to shift on COCO val dataset 40
3.6 Plot of class scores for a small horizontal translation 45
3.7 Overview of the network architecture of YOLOX with the proposed method 46
3.8 Observed score variation for shifts in the One-stage methods FCOS and YOLOX and the Two-stage method Faster RCNN. 49

4.1 SODet Architecture 54
4.2 Network of SODet 56
4.3 Sample images of masked cityscape verification data 58
4.4 Results of comparative evaluation of AP at varying IoU thresholds . . 61
4.5 Confidence score histogram for classification. 62
4.6 Displays the results of the object detection 63
4.7 Displays the results of the instance segmentation 64

5.1 Visualization of our proposed approach 70
5.2 Illustration of graph Laplacian matrix 71
5.3 Illustration of graph Laplacian matrix for difference of neighboring pixels 72

5.4 Network architecture used in our experiments 73
5.5 Successful examples of improvements in the foreground and background boundaries 77
5.6 Successful examples of improvements at the boundaries with other classes 78
5.7 Failure examples where the proposed method has reduced the quality of the mask 79
5.8 Histogram of posterior probabilities of mask prediction 80

List of Tables

3.1 Comparison of baseline on COCO val dataset 39
3.2 Comparison of Data Augmentation Methods on the COCO val set . . 42
3.3 Comparative evaluation on grid boundaries 43
3.4 Analysis of different hyper-parameters for data augmentation ratio . . 43
3.5 Analysis of different head configurations for SGFEM on the COCO val set 44
3.6 Analysis of different hyper-parameters for auxiliary loss weight 44
3.7 Comparison of proposed Methods on the COCO val set with YOLOX.FPS measured in runtime on a GeForce RTX3090. 48

4.1 Comparison results between the binary mask and soft mask 58
4.2 Comparison of the average precisions for mask functions 60
4.3 Comparison of Average Precision(AP) between our proposed method and various state-of-the-art methods 60
4.4 Comparison of Average Precision(AP) between proposed method and various state-of-the-art methods on COCO val set 65

5.1 Results of instance partitioning with varying number of λ_g 75
5.2 Comparison of baseline and several regularization methods on COCO val dataset 75
5.3 Comparison of baseline and several regularization methods on Cityscapes val dataset. 76

List of Abbreviations

CNN	**C**oncolutional **N**eural **N**etwork
FCN	**F**ully **C**onvolutional **N**etwork
FPN	**F**eature **P**yramid **N**etwork

Chapter 1

Introduction

1.1 Overview of Object Detection and Instance Segmentation

In the field of computer vision, two fundamental tasks have gained remarkable prominence over the years: object detection and instance segmentation. These tasks serve as the cornerstones for a wide range of applications, from autonomous navigation to content understanding, and play a pivotal role in modern technology.

1.1.1 Overview of Object Detection

Object detection is a fundamental computer vision task that revolves around the identification and localization of objects within an image. At its core, it entails answering two critical questions for each object: "What is it?" and "Where is it located?" Object detection goes beyond simple object recognition by providing precise spatial information in the form of bounding boxes. These bounding boxes encapsulate the object's position and dimensions, allowing for object localization and categorization simultaneously.

1.1.2 Overview of Instance Segmentation

Instance segmentation is a closely related task that builds upon the foundations of object detection. In instance segmentation, the objective is twofold: not only to identify and localize objects but also to provide a pixel-wise delineation of each individual object instance within an image. This results in a fine-grained segmentation mask for each object, allowing for a comprehensive understanding of the object's spatial extent.

While object detection is concerned with bounding boxes and object categorization, instance segmentation takes the analysis to a pixel level, providing a more granular representation of the objects. This level of detail is particularly valuable in scenarios where objects may overlap or interact closely within the same region of the image.

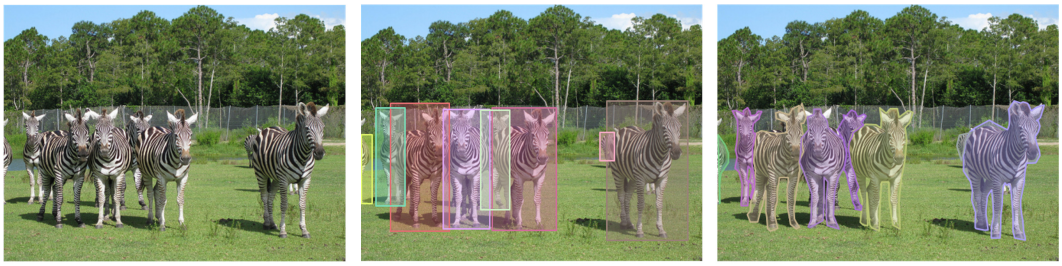


FIGURE 1.1: A sample annotated image from the COCO dataset, illustrates the difference between object detection and instance segmentation.

1.1.3 Differences of Each Task

Object detection and instance segmentation share a substantial similarity in terms of their fundamental task. Both aim to identify objects within images and provide spatial information about them. While the basic network structure used for predictions is fundamentally similar for both object detection and instance segmentation, variations arise in the head components of the network, tailored to the respective tasks. The backbone, responsible for feature extraction and object representation, is often shared.

The critical differences lie in predictions and the ground-truth information used during training:

Ground-Truth Information

In object detection, the ground truth for training typically consists of bounding box coordinates and class labels, specifying the object's category. Fig. 1.1 (b) shows some examples of annotated bounding boxes. In contrast, instance segmentation requires pixel-wise annotations, where each pixel is associated with a specific object instance. Fig. 1.1 (c) shows some examples of annotated segmentation masks.

Prediction Resolution

Object detection networks produce predictions at the level of bounding boxes, defining the object's position and category. Instance segmentation networks, on the other hand, provide pixel-level predictions, creating detailed segmentation masks for each object instance.

Despite these nuanced differences, both object detection and instance segmentation serve as indispensable and practical technologies with diverse applications across numerous domains. Their significance lies in their ability to provide machines with the capability to interpret and interact with the visual world, fostering advancements in automation, safety, and content understanding.

We treated these two domains as the same task with the same challenges and approached our research with this in mind.

1.2 The Emergence of Real-World Object Detection Applications

In recent years, the field of computer vision has witnessed significant advancements, primarily driven by the development of Deep Neural Networks (DNNs). Among the various computer vision technologies, object detection stands out as a crucial and rapidly evolving area. Object detection refers to the process of identifying and locating objects of interest within images or video frames. One of the key advantages of object detection is its versatility. It can be used in a variety of real-world scenarios in automating tasks that require the identification and localization of objects in images, making it an important technology in many applications.

The impact of object detection extends far beyond the confines of academia and research laboratories. It has permeated into various industries, enriching them with enhanced capabilities and safety measures. Here, we explore some of the practical applications where object detection has made a substantial difference:

1.2.1 Survellance Camera Systems

Surveillance camera systems have become ubiquitous in our modern world. Object detection plays a pivotal role in enhancing the effectiveness of these systems. It enables automated monitoring and alerting, allowing security personnel to focus their attention on potential threats. In crowded spaces, object detection can identify individuals, track their movements, and trigger alarms in the event of suspicious activities, contributing to improved public safety.

1.2.2 Medical Imaging

In the field of healthcare, object detection has found vital applications in medical imaging. Radiologists and healthcare practitioners rely on accurate and timely identification of anomalies and pathological structures within medical images. Object detection models can assist in locating tumors, identifying fractures, and detecting abnormalities, thereby expediting diagnoses and improving patient outcomes.

1.2.3 Automotice Safety Systems

The automotive industry has witnessed a significant transformation with the integration of object detection into vehicles. Advanced Driver Assistance Systems (ADAS) utilize object detection to enhance road safety. These systems can detect pedestrians, other vehicles, and road signs, providing crucial information to the vehicle's control system. In critical situations, object detection can trigger autonomous emergency braking or other safety measures to prevent accidents.

1.3 Risk of malfunction

Object detection systems have become integral components of various systems, with a common goal of achieving autonomous operations. However, as we embrace this advancement, it is imperative to recognize and address the inherent risks associated with the deployment of object detection in critical domains, where the consequences of system malfunction can lead to severe accidents and unintended outcomes. These unintended consequences can range from false positives to false negatives, each carrying its own set of risks.

1.3.1 Surveillance and Security: Detecting Intruders or False Alarms?

Consider the scenario of using object detection to monitor a high-security facility. The consequences of a malfunction in this context are dire. If the system fails to detect an intruder, it could result in a significant security breach, potentially jeopardizing lives and valuable assets.

On the other hand, if the system repeatedly misidentifies small animals as humans, it can lead to a cascade of false alarms, eroding trust in the system's reliability. This not only diminishes the effectiveness of the security measures but also diverts valuable resources to investigate false incidents.

1.3.2 Medical Imaging: The Consequences of False Positives and Negatives

In the realm of healthcare and medical imaging, object detection plays a pivotal role in diagnosing and treating patients. Medical professionals rely on accurate and consistent results to make critical decisions about patient care. However, the risk of malfunction in medical object detection systems can have profound consequences.

Consider a scenario where a medical imaging system frequently produces false positives. This means that it erroneously identifies anomalies or abnormalities that do not exist. Such frequent false alarms can create confusion and doubt among healthcare practitioners, leading to unnecessary interventions, additional tests, and potentially misdiagnoses. This not only places an additional burden on the healthcare system but also poses risks to patient safety.

Conversely, the risk of false negatives in medical imaging object detection is equally concerning. If the system fails to detect critical abnormalities, it can lead to delayed diagnoses and missed opportunities for timely interventions. This delay can significantly impact patient outcomes and the effectiveness of medical treatments.

1.3.3 Automotive Safety: Avoiding Catastrophic Accidents

Another domain where object detection is in Advanced Driver Assistance Systems (ADAS), which are designed to enhance road safety. Accurate object detection is crucial for identifying pedestrians, vehicles, and other obstacles on the road.

Failure to detect pedestrians or other vehicles can lead to catastrophic accidents, especially in situations where a human driver might not have sufficient time to react. Moreover, misclassifying objects, such as mistaking a stationary object like a utility pole for a human, can trigger emergency braking systems unnecessarily, potentially causing rear-end collisions or loss of control.

1.4 Addressing the Challenge

The challenges posed by the risk of malfunction in object detection for critical applications are undeniable. As object detection continues to permeate our daily lives and essential systems, there is an urgent need to address this challenge comprehensively.

As researchers and practitioners, we have been committed to bridging the gap between academic achievements in object detection and their practical applications in various industries. However, this journey has been fraught with challenges, primarily stemming from the susceptibility of object detection systems to unexpected anomalies.

In this dissertation, we embark on a comprehensive exploration of three pivotal questions that have guided our research endeavors:

1.4.1 Is object detection weak against translations?

In recent years, Convolutional Neural Networks (CNNs) have become the cornerstone of deep learning for computer vision tasks, including object detection. The foundation of CNNs can be traced back to Fukushima's Neocognitron, proposed in 1980. Inspired by the brain's structural principles. In particular, the combination of convolution and pooling employed by Fukushima et al. [20] in their Neocognitron for CNNs is motivated by the desire to give the network invariance to image translations, scaling, and other small deformations. Building upon Fukushima's ideas, Yann LeCun introduced LeNet [34], a neural network capable of end-to-end learning through error backpropagation. This marked a significant milestone in the exploration of neural network capabilities, expanding their potential for various tasks.

However, CNNs, which are widely employed in classification tasks, have been reported to be sensitive to image transformations such as translations, rotations, and scales [1, 52]. Even subtle changes introduced by such transformations can dramatically reduce the accuracy of classification models.

Given the prevalence of CNN-based architectures in object detection, we questioned whether object detection systems were also vulnerable to this sensitivity. Our experiments revealed that indeed, even slight translations of objects within an image could lead to significant variations in prediction results.

Fig. 1.2 shows the results we observed in our shift-invariance evaluation. The left and right pictures in Fig. 1.2 show the images before and after an 18-pixel translation in the horizontal direction. These show that the prediction results for all objects are different, even though the results were obtained from the same network. The dog in

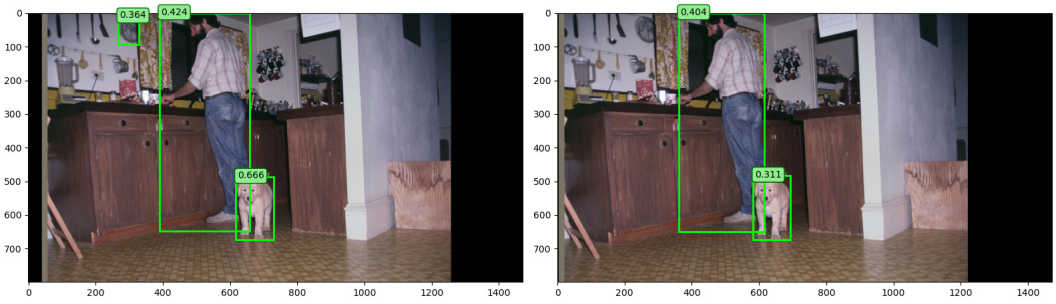


FIGURE 1.2: Qualitative result of shift variance for FCOS on the COCO validation set, with confidence threshold set to 0.3. Inference results on the image before and after shifting the input image horizontally by 18 pixels.

the picture shows a decrease in class score from 0.666 to 0.311. In the right picture, the clock fails to detect because it is below the detection threshold of 0.3.

This vulnerability is particularly problematic in practical scenarios. For instance, in surveillance applications, the potential for object detection to fail when an intruder is slightly translated within the frame poses a severe security risk. Similarly, in the context of Advanced Driver Assistance Systems (ADAS), where pedestrian detection is crucial, small movements of pedestrians could lead to inconsistent detection results, which is far from ideal.

Object detection models inherently possess a multi-resolution information propagation component, which is often considered a strength. This multi-resolution structure allows fine-grained information, such as the effects of parallel translations, to be transmitted effectively to the final output. Despite this design, we found that object detection systems were still susceptible to the influence of subtle translations. Our research seeks to delve into the factors contributing to this vulnerability and propose solutions to enhance the robustness of object detection systems against parallel translations.

1.4.2 Can We Address Class Imbalance in Object Detection by Questing for Balanced Foregrounds?

The application of object detection to pedestrian safety systems in autonomous vehicles suffers from the problem of background misrecognition. One of the factors contributing to this misrecognition is the problem of class imbalance. This is even worse in the context of object detection than in classification tasks.

In standard image classification tasks, the problem of class imbalance arises when some categories are vastly overrepresented while others are underrepresented or even absent in the training data. This imbalance can severely impact the model's ability to generalize effectively, leading to biased predictions in favor of the majority class. In

this case, one solution would be to spend an enormous amount of effort re-collecting training data.

However, in the domain of object detection, the class imbalance problem takes on a new dimension. Beyond the standard class imbalance, object detection faces the additional challenge of foreground-background class imbalance [44]. This arises due to the inherent nature of object detection, where the majority of the image regions correspond to the background, devoid of any objects of interest.

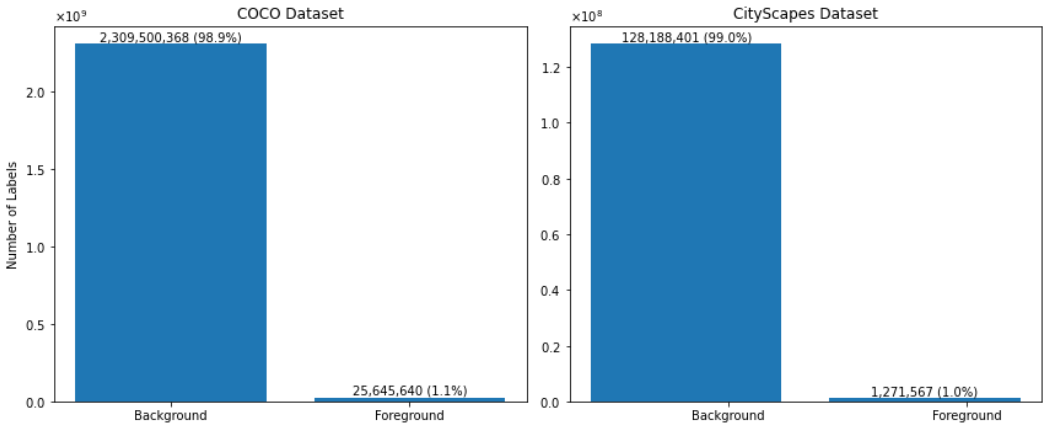


FIGURE 1.3: Number of label assignments in FCOS for background and foreground for MS-COCO dataset and CityScapes.

To underscore the pervasive nature of foreground-background imbalance in object detection, let us turn our attention to Fig. 1.3, which depicts the distribution of foreground and background labels in the COCO [39] and CityScapes [9] datasets when utilizing the FCOS (Fully Convolutional One-Stage) [57] object detection framework.

As Fig. 1.3 illustrates, both datasets exhibit a staggering dominance of background regions, encompassing approximately 99% of the dataset.

The ramifications of foreground-background imbalance extend beyond mere statistical curiosity. This imbalance introduces instability into the training process, making it challenging for the model to effectively learn from foreground examples.

Various indirect methods have been proposed to tackle this issue, including customized loss functions [38] and alternative strategies for label assignment [41, 57, 22]. While these approaches have shown promise, they still lack a direct means of addressing the imbalance at its core.

In response to this challenge, our research embarks on a quest to investigate whether we can directly leverage background information to alleviate the pervasive foreground-background imbalance in object detection. Rather than relying on indirect methods, we explore the feasibility of incorporating explicit background information as a part of the model's input. This approach represents a departure from traditional methods and holds the promise of directly mitigating the imbalance, potentially leading to more stable training and enhanced model performance.

1.4.3 Why does the segmentation mask fail at the boundary?

In recent years, instance segmentation has also seen dramatic improvements in mask accuracy. This progress has been driven by state-of-the-art techniques and architectures, making instance segmentation one of the most exciting and impactful areas in computer vision. The appeal of instance segmentation lies in its ability to provide a pixel-level segmentation mask that provides a more intuitive representation of objects than the bounding boxes required by object detection. The ability to make pixel-level predictions opens up new possibilities for applications that require a fine-grained understanding of visual data.

Instance segmentation suffers from the cost of annotation. However, zero-shot segmentation methods such as SAM (Segment Anything Model) [31] have recently been proposed, and an era where annotation is no longer necessary is approaching. The attractive prospect of easily segmenting objects that were not seen during training holds great promise for real-world applications.

While the achievements in instance segmentation are undeniable, a closer inspection of the predicted masks reveals areas ripe for enhancement. Notably, the challenging aspects lie at the boundaries of objects and classes. Predictions often falter in these regions, resulting in errors and inconsistencies. This is a critical issue that requires our attention.

FIGURE 1.4: An example of mask failure at boundary.

The root of this problem can be attributed to how predictions are currently made. Existing models tend to focus on pixel-level errors, scrutinizing predictions at a granular level but overlooking the holistic context that human perception inherently relies upon. When humans recognize object boundaries, they draw upon contextual information from surrounding pixels, facilitating a more coherent understanding of the

scene. The failure to effectively incorporate such spatial contextual information into the learning process is a prime contributor to the challenges at hand.

To address this issue, our research focused on the context of the image, with a particular focus on the pixel-to-pixel structure of the object boundary. We thought that the training of the network needed to provide information about the spatial context, i.e. the interactions of the pixels that make up the structure of the image. One of our research topics was how to get the network to learn the concept of spatial context as a guiding teacher for the model.

By providing models with spatial relationships among pixels, we aim to imbue them with a richer understanding of the context in which objects exist. This spatial context, we believe, will enable more accurate and coherent mask predictions, especially in regions where object boundaries meet background or other objects.

1.5 structure of the thesis

In the following chapters, we further explore the technical aspects of object detection and instance segmentation. We then describe in detail the solutions that we have developed through investigation and analysis of the three problems mentioned. Finally, we summarise our contributions throughout this thesis.

1.5.1 Related works

Chapter 2, we provide a comprehensive overview of the state-of-the-art techniques in object detection and instance segmentation. We explore the evolution of these fields, from traditional methods to the most recent innovations. By understanding the current landscape, we lay the foundation for the novel approaches presented in subsequent chapters.

1.5.2 Improved head and data augmentation to reduce artifacts at grid boundaries in object detection

Chapter 3 focuses on a novel approach that centers on grid boundaries as a fundamental element in improving object detection. We delve into the intricacies of this paradigm, elucidating how it alters the traditional methodologies and contributes to enhanced performance.

1.5.3 An Object Detection Method Using Probability Maps for Instance Segmentation to Mask Background

Chapter 4, presents techniques for refining object detection and instance segmentation by resolving foreground-background imbalances through effective background masking.

1.5.4 Graph Laplacian Regularization based on the Differences of Neighboring Pixels for Conditional Convolutions for Instance Segmentation

Chapter 5 delves into the art of enhancing instance segmentation boundaries. We propose novel methods to sharpen the boundaries between objects, resulting in clearer and more precise segmentation. The techniques presented aim to transform the way we perceive object boundaries.

1.5.5 Conclusion

The final chapter concludes our scientific journey by summarising the key findings, contributions and implications of our work. It also outlines the potential for future research and describes directions for further progress in the exciting areas of object detection and instance segmentation.

Chapter 2

Evolution of Object Detection and Instance Segmentation

This chapter delves into the history of object detection, tracing its development from its early days to the current state-of-the-art methods. Object detection has been a fundamental task in computer vision, and its evolution has been marked by a relentless pursuit of both accuracy and processing speed.

This section explores key milestones and breakthroughs in the field and details the techniques that have paved the way for modern object detection systems.

2.1 Object Detection Methods

2.1.1 Traditional Detectors

In the 1990s, object detection relied primarily on handcrafted feature representations. Face detection methods at the time used feature representations based on Gabor filters modeled after the human retina, which produced beneficial results [50, 16]. These representations were designed to capture specific features of objects in images. While they showed promise, they were limited in their ability to handle complex variations in object appearance, pose, and scale.

A significant breakthrough in real-time object detection came with the introduction of the Viola-Jones (VJ) detector in 2001 [61]. The VJ detector laid the foundation for real-time processing by employing a sliding window approach. It scanned all possible locations and scales within an image, checking whether each window contained a human face.

The VJ detector proposed several important technologies that will be influential in this field. One is an integral image. This is a technique that avoids redundancy in feature computation and speeds up the process. Another is detection cascades. The detection cascade allows negative samples to be quickly removed, further increasing processing speed. These innovations not only contributed to real-time object detection but also influenced subsequent developments in the field.

In 2005, the Histogram of Oriented Gradients (HOG) [11] emerged as a significant advancement in feature extraction. Inspired by the response of neurons in the primary visual cortex to image gradients, HOG provided a robust representation of object

edges and textures. HOG-based methods divided images into grids and computed gradient histograms within each grid cell. This approach proved effective in capturing object contours and patterns, making it widely adopted in object detection.

HOG paved the way for the development of deformable part-based models (DPM) [17, 19] and other derivatives, which extended its principles to handle more complex object structures and deformations. Techniques like bounding box regression and hard negative mining, introduced during this era, continue to serve as foundational concepts in modern deep learning-based object detection.

2.1.2 CNN-based Detectors

The advent of deep learning marked a significant breakthrough in the field of computer vision, including object detection. Deep neural networks revolutionized how features were extracted and how object detection was approached. The traditional handcrafted feature extraction methods struggled to capture the complex and hierarchical patterns present in images.

Deep learning-based object detectors brought about a paradigm shift by automatically learning discriminative features from data. Convolutional Neural Networks (CNNs) emerged as the backbone of these detectors, enabling the efficient extraction of meaningful features.

Object detectors based on CNNs can be broadly categorized into two-stage and single-stage methods, each with its strengths and weaknesses.

Two-stage methods, as the name suggests, involve a two-step process for object detection. These methods first propose a set of potential object regions or bounding boxes, often referred to as region proposals, and then classify these regions to determine the presence of objects and refine their locations.

Single-stage methods, in contrast, directly predict object bounding boxes and class labels from a fixed set of anchor boxes or default boxes, eliminating the need for a separate region proposal stage. These methods aim to achieve a balance between speed and accuracy by simplifying the detection pipeline.

The core architecture of modern object detectors consists of three primary components: the backbone, the neck, and the head. The following Fig. 2.1 illustrates the basic structure of modern object detectors:

Backbone is responsible for feature extraction from input images. It typically employs well-known CNN architectures like VGG [51] or ResNet [26]. These architectures have proven effective in capturing hierarchical features, from edges and textures to object parts and semantics. The backbone generates a feature map that serves as the basis for subsequent processing.

Neck positioned between the backbone and the head, the neck's primary role is to refine the feature map from the backbone. It enhances the representational power of the features, making them more suitable for object detection. One commonly used

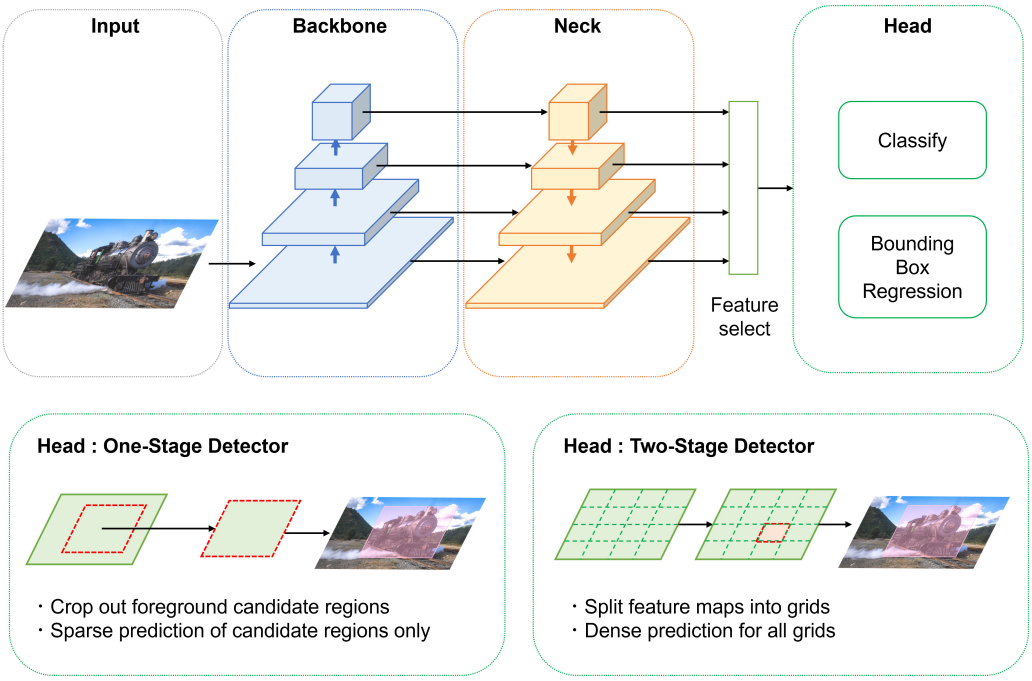


FIGURE 2.1: Basic network architecture of modern object detector.

neck architecture is the Feature Pyramid Network (FPN) [37], which helps address scale variation in object sizes.

Head takes the refined feature map from the neck and performs object detection. It predicts bounding box coordinates (object locations) and assigns class labels to the detected objects. The head's design can vary across different object detection models and has a significant impact on the model's accuracy and speed.

2.1.3 Two Stage Methods

The transition to deep learning for object detection began with the advent of R-CNN (Region-based Convolutional Neural Network) [23]. R-CNN introduced a groundbreaking approach by combining Convolutional Neural Networks (CNNs) with traditional object detection methods. Its architecture laid the foundations for two-stage detectors and became a fundamental technology for future advances.

R-CNN followed a two-stage detection process. The first stage involved generating region proposals—candidate bounding boxes that potentially contained objects of interest. In the second stage, the task of object detection is performed by predicting the categories of objects contained in the candidate regions. These proposals were obtained using an algorithm called Selective Search [59].

Selective Search, based on the Felzenszwalb method [18], was instrumental in generating region proposals. It operated by grouping similar pixels into segments and

gradually merging them to form larger regions. This hierarchical approach created a diverse set of region proposals by exploring multiple scales and shapes.

In the second stage, R-CNN extracted features from the candidate regions using a CNN. AlexNet [33], a pioneering deep neural network, was commonly employed for feature extraction. However, AlexNet had a fixed input size, necessitating the resizing of candidate region images into a square format.

Once features were extracted, a linear Support Vector Machine (SVM) [60] was utilized for classifying the objects within the candidate regions. R-CNN's learning process consisted of two separate steps: training the CNN-based feature extraction part and training the SVM classifier separately.

R-CNN laid the groundwork for future advancements in object detection. However, it had certain limitations, such as slow training and inference speeds due to its multi-step process of region proposal, feature extraction, and classification. Researchers focused on overcoming these challenges, leading to the development of more efficient two-stage detectors.

SPP-Net improvement upon R-CNN was SPP-Net (Spatial Pyramid Pooling Networks) [29]. SPP-Net introduced a novel pooling architecture that significantly enhanced both processing speed and accuracy.

SPP-Net addressed the inefficiency of fixed-size inputs by introducing Spatial Pyramid Pooling (SPP). Rather than resizing candidate regions to a fixed size, SPP-Net employed pyramid pooling, which created a spatial pyramid of bins of different sizes. This allowed the network to process regions of various dimensions effectively.

SPP-Net also introduced multi-task learning, simultaneously handling object detection and bounding box regression. It learned to predict class labels and refine bounding box coordinates in a unified manner. This innovation simplified the architecture and training process, making it more efficient.

SPP-Net's contributions, especially the spatial pyramid pooling technique, remain influential in modern object detection systems. Its ability to handle regions of varying sizes and multi-task learning set the stage for subsequent advancements in object detectors.

Fast R-CNN emerged as a breakthrough in the evolution of two-stage detectors [22]. It introduced the concept of end-to-end learning, streamlining the training process and improving both accuracy and efficiency.

One of the key innovations in Fast R-CNN was the integration of the entire detection process into a single CNN. This allowed for end-to-end learning, where the model learned to generate region proposals, extract features, and perform classification and bounding box regression in a single pass. This holistic approach reduced training time and complexity.

Fast R-CNN introduced Region of Interest (RoI) pooling, an efficient technique for cropping and resizing features from the feature map. RoI pooling enabled the

network to adaptively process candidate regions of various sizes and aspect ratios, eliminating the need for fixed-size inputs.

Fast R-CNN's achievements, particularly its end-to-end learning and RoI pooling, significantly influenced subsequent advancements in object detection. The ability to train the entire network in a unified manner paved the way for more efficient and accurate detectors.

Faster R-CNN represented a major leap forward in two-stage detectors by addressing the inefficiencies associated with region proposal [47]. It introduced the concept of Region Proposal Networks (RPNs) and achieved remarkable improvements in both speed and accuracy.

Faster R-CNN seamlessly integrated region proposal generation into the CNN. RPNs used the features extracted from the shared backbone network to predict region proposals directly. This eliminated the need for external algorithms like Selective Search and significantly accelerated the region proposal process.

One of the key innovations of Faster R-CNN was the introduction of anchors—fixed aspect ratio boxes that served as priors for object locations. These anchors helped guide the initial predictions of bounding box coordinates. By using multiple anchor scales and aspect ratios, Faster R-CNN improved the accuracy of initial predictions.

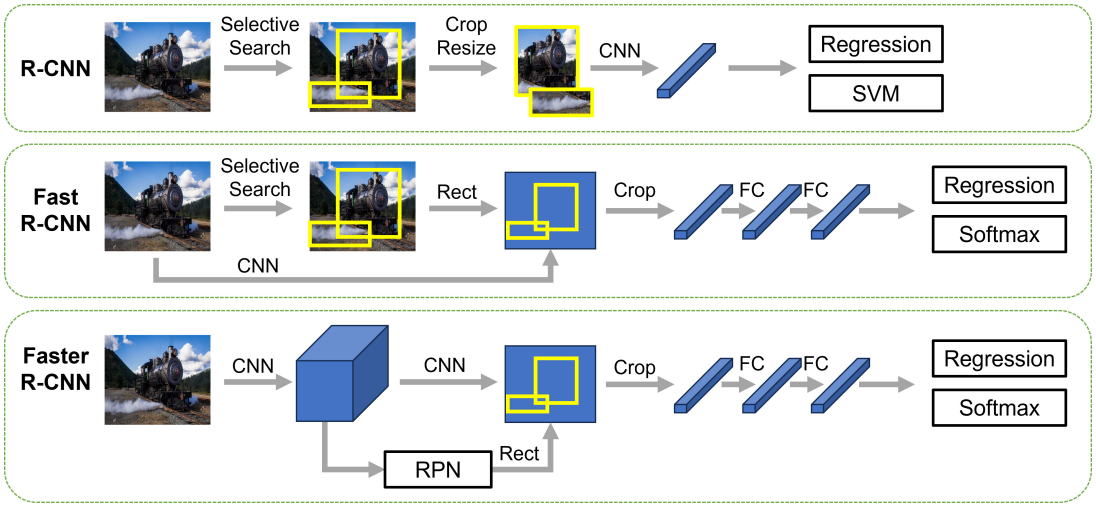


FIGURE 2.2: Evolution of the two-stage method.

Single Stage Methods

As discussed earlier, two-stage detectors brought substantial improvements in object detection accuracy, but they came with a trade-off in processing speed due to the need for an additional region proposal step. One-stage detectors emerged as a response to this challenge, emphasizing speed and simplicity while maintaining competitive detection performance.

Around the same time as Faster R-CNN, a new paradigm of object detection began to gain traction - Anchor-Based one-stage detectors. These methods introduced a significant departure from the two-stage pipeline by directly predicting object bounding boxes and class labels within a dense grid of anchor boxes placed across the image.

In Anchor-Based one-stage detectors, the feature maps produced by the backbone network are densely populated with anchors. These anchors span a range of scales and aspect ratios, and they are distributed uniformly across the spatial grid. Instead of proposing region candidates, these detectors process the entire image in a grid-wise manner, simultaneously predicting class probabilities and bounding box offsets for each anchor.

YOLO One of the pioneering one-stage detectors is YOLO (You Only Look Once) [46]. YOLO introduced a unique architecture, using the Darknet neural network as its backbone. YOLO's core idea is to divide the image into an $S \times S$ grid of equally sized cells, where each cell is responsible for predicting bounding boxes and class labels.

YOLO's prediction process involves three primary components: Bounding Box Regression: Each cell predicts multiple bounding boxes with associated coordinates relative to the cell's boundaries. These predictions aim to localize objects. Class Prediction: YOLO assigns class probabilities for each bounding box, indicating the presence of specific object classes within the box. Objectness Prediction: An additional prediction, known as "objectness," estimates the probability that an object's center falls within a given cell. This helps filter out boxes that do not contain objects.

YOLO then combines these three types of predictions and uses them as the basis for learning. By providing supervision for these aspects during training, YOLO learns to predict bounding boxes directly from the image in a single pass.

SSD Another notable Anchor-Based one-stage detector is SSD (Single Shot MultiBox Detector) [41]. SSD adopts a fully convolutional network (FCN) architecture and relies on pre-trained backbones such as VGG or InceptionNet [54].

Unlike YOLO, SSD applies multiple detection heads to feature maps of varying resolutions, enabling it to handle objects at different scales effectively. Each detection head predicts class scores and bounding box offsets specific to its associated feature map.

While SSD and YOLO offered significant speed improvements compared to two-stage detectors, they faced some limitations. For instance, YOLO had difficulty detecting small objects, and SSD was still outperformed by two-stage methods on certain datasets. These challenges motivated further research into refining one-stage detectors and addressing their weaknesses.

The Anchor-Based one-stage detectors, particularly YOLO and SSD, laid the foundation for modern one-stage detectors.

While one-stage detectors demonstrated advantages in processing speed compared to two-stage detectors, they faced challenges in achieving the same level of detection accuracy. One pivotal moment in overcoming these challenges, which continued to be understood and improved upon, was the introduction of RetinaNet [38] in 2017.

RetinaNet was proposed in a paper titled "Focal loss for dense object detection" citefocal. The author of RetinaNet recognized that the key issue hampering the accuracy of one-stage detectors was the problem of class imbalance during training. Class imbalance occurs when there is a significant disparity in the number of instances of different classes in the training dataset.

For one-stage detectors, which consider the entire image in one pass, the majority of regions contain background objects, leading to an unbalanced class distribution. Two-stage detectors partially alleviate the class imbalance problem by filtering out a majority of easy negatives in the first stage, focusing on more challenging regions in the second stage. This imbalance makes learning more efficient in the second stage. However, One-stage detectors do not have this luxury.

To tackle this challenge, RetinaNet introduced the innovative Focal Loss. The Focal Loss addressed class imbalance by assigning higher weights to hard-to-classify examples during training. Essentially, it "focused" on samples that had been misclassified in previous iterations, giving them more importance in the loss calculation. This idea effectively down-weighted easy negatives, thereby mitigating the problem of overwhelming background regions.

The introduction of Focal Loss marked a breakthrough in the field of object detection. It became a fundamental component in improving the accuracy of one-stage detectors. Focal Loss significantly mitigated the class imbalance issue and allowed one-stage detectors to compete with their two-stage counterparts in terms of accuracy, all while maintaining real-time processing capabilities.

2.1.4 Anchor Free Detectors

In anchor-based object detection, numerous anchor boxes are used to cover various aspect ratios and scales within an image. While this approach provides flexibility, it exacerbates the class imbalance problem during training. Anchor-free methods spurred the success of RetinaNet and Focal Loss, and several single-stage detectors appeared in the following years, pushing the boundaries of object detection performance.

The anchor-free approaches embraced by FCOS [57], CenterNet [14], and Foveabox [32] were motivated by a desire to mitigate this class imbalance issue. By eliminating the need for anchors and assigning one object prediction per grid cell, they restructured the detection process to focus on objects directly and alleviate the impact of the background class. This shift led to a more balanced training process, resulting in improved detection accuracy and performance. These methods marked

a significant step forward in the evolution of one-stage detectors and became pivotal in addressing the challenges associated with class imbalance in object detection.

FCOS (Fully Convolutional One-Stage Object Detection) : FCOS introduced the concept of an "anchor-free" detection approach.

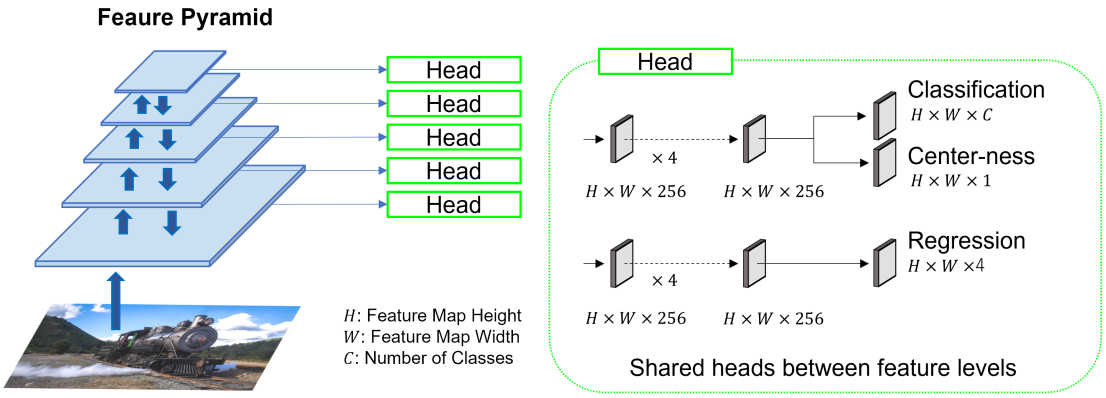


FIGURE 2.3: FCOS network architecture.

Traditionally, anchor-based methods lead to class imbalance, as the number of background labels is much larger than that of object labels due to the multitude of anchors. FCOS alleviated this problem by predicting objects without using anchors, improving accuracy by focusing on object instances. FCOS further refined the assignment of positive and negative samples, outperforming RetinaNet.

CenterNet : CenterNet took a different approach by considering the detection task as finding the center point of objects. It focused on learning the center key point and regressing the object size. This keypoint-based approach enabled accurate and efficient object detection and offered an alternative to anchor-based methods.

Foveabox : Foveabox continued the trend of "anchor-free" detection. It adopted the idea of assigning one object per grid cell, eliminating the class imbalance issue and allowing each object to have its dedicated prediction.

2.2 Instance Segmentation Methods

Instance segmentation is a challenging computer vision task that extends the capabilities of both object detection and semantic segmentation. It requires not only identifying and locating objects within an image but also precisely delineating each instance with pixel-level accuracy. It involves identifying and delineating individual object instances within an image, assigning each pixel to a specific object, and providing a unique label for each instance. This task requires a model to not only detect objects but also to generate fine-grained masks that precisely outline the shape of each object.

Instance segmentation is closely related to semantic segmentation, as both tasks involve pixel-wise predictions. However, the key distinction lies in their objectives. In semantic segmentation, the goal is to assign each pixel to a particular object category or class, ignoring instance-specific distinctions. In contrast, instance segmentation goes a step further by distinguishing individual instances of the same class, providing a distinct label for each object instance. This distinction brings an added layer of complexity to the task.

The history of instance segmentation is intertwined with object detection, particularly the development of Faster R-CNN, which served as a foundational framework. Faster R-CNN is a fundamental two-stage object detection model that employs region proposal networks (RPNs) for region-of-interest (ROI) generation. This methodology significantly improved object detection accuracy.

Mask R-CNN Following the success of Faster R-CNN, Mask R-CNN [28] was introduced in 2017 as a natural extension of the framework.

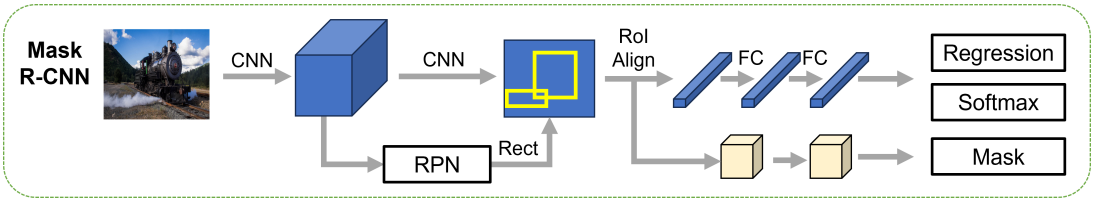


FIGURE 2.4: Mask R-CNN network architecture.

Mask R-CNN is a pioneering instance segmentation model that combines object detection with pixel-wise mask prediction.

Mask R-CNN retains the two-stage architecture of Faster R-CNN but extends it to include mask prediction for each region of interest. The model generates high-quality object masks in addition to bounding boxes and class labels. This breakthrough allowed computer vision applications to progress beyond object detection and semantic segmentation by providing fine-grained object delineation.

Subsequently, the following Mask R-CNN derivation methods have been proposed, leading to improved mask quality.

Cascade Mask R-CNN To further improve mask prediction and segmentation quality, Cascade Mask R-CNN was introduced. Cascade Mask R-CNN [4], an extension of Mask R-CNN, iteratively refines masks in multiple stages. This approach enhanced the accuracy of mask predictions by progressively adjusting and refining the masks through a series of stages. This iterative refinement process allowed the model to achieve highly precise instance segmentation.

Mask Scoring R-CNN Mask Scoring R-CNN [30] was another significant development in instance segmentation. It addressed a critical limitation of existing models.

While traditional instance segmentation models generate masks for all detected objects, these masks may not always be accurate. Mask Scoring R-CNN introduced a mask quality assessment mechanism that ranks and refines the predicted masks, enhancing the overall segmentation quality.

The contributions of Mask Scoring R-CNN and other Mask R-CNN-based approaches significantly improved the accuracy of instance segmentation. However, these methods predominantly rely on local information detected by the detectors, and they have a notable limitation in their inability to effectively utilize global image context information.

As discussed earlier, as RetinaNet outperformed Faster R-CNN in the context of object detection, the emergence of the one-stage method was expected to lead to a similar paradigm shift in instance segmentation. Transitioning to one-stage approaches holds the promise of facilitating the integration of global information, a limitation inherent to traditional methods. YOLACT [3] emerged as the pioneering solution to this challenge, effectively harnessing global context within the domain of instance segmentation.

YOLACT While most instance segmentation models, including Mask R-CNN, adopt a two-stage architecture, YOLACT took a different approach. YOLACT is a one-stage instance segmentation model that provides real-time performance with global context information. YOLACT leverages a single convolutional neural network to predict both masks and class scores concurrently. This streamlined approach demonstrated that one-stage models could achieve impressive instance segmentation results while maintaining real-time capabilities.

However, it has been demonstrated that relying solely on global information leads to ambiguities in the masks of object boundaries. Consequently, it became evident that the generation of precise masks requires the incorporation of local information.

Combining Local and Global Features CenterMask [35] and BlendMask [6] represent innovative approaches to instance segmentation by combining local and global features, while leveraging the FCOS network architecture. These models leverage a dual-head design, with one head focusing on local features and another on global features. This combination allows them to capture both fine-grained object details and contextual information, significantly enhancing the quality of instance segmentation.

Early iterations of one-stage instance segmentation models exhibited a promising blend of local and global information. However, these models often struggled to surpass the accuracy achieved by their two-stage counterparts, exemplified by Mask R-CNN. The challenge was to reconcile the efficiency of one-stage models with the precision of two-stage models.

SOLO In response to the challenge, the research group behind the Fully Convolutional One-Stage (FCOS) detection framework introduced SOLO (Segmenting Objects by Locations) [62]. The FCOS framework had successfully eliminated the need for anchors, pioneering anchor-free object detection on a grid-cell basis. SOLO expanded this concept to pixel-level segmentation, a remarkable achievement.

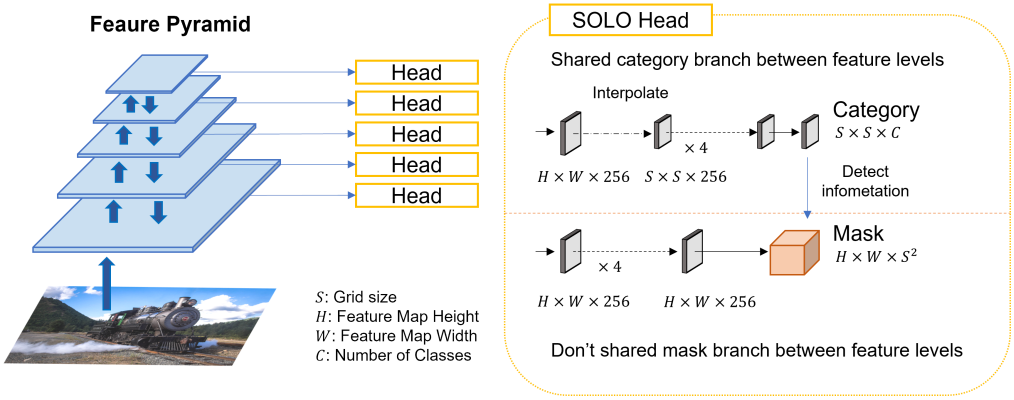


FIGURE 2.5: SOLO network architecture.

The presented approach simplifies the process by transforming the branch responsible for bounding box generation in FPN-based object detection into an instance mask generation branch. The detection process encompasses two branches: class estimation and mask estimation.

The class branch divides the image into $S \times S$ grids, and the grid cell corresponding to the center of the instance infers the class category C. Grid cells that do not contain centers are treated as background.

The mask branch produces masks of resolution $H \times W$ corresponding to the output of the class estimation branch. Specifically, since the class estimation branch outputs a resolution of $S \times S$, the mask generation branch yields an output dimensionality of $H \times W \times S^2$. The mask branch is structured as an $H \times W$ feature map to maintain high resolution, with one mask prediction for each grid cell in the category branch. Therefore, the output of the mask branch is $H \times W \times S^2$.

SOLOv2 SOLOv2 [63] sought to tackle the challenge of high computational costs. The original SOLO architecture necessitated extensive feature maps due to the large number of grid cells, resulting in a considerable amount of redundancy. SOLOv2 introduced the use of dynamic convolution, a method to mitigate this redundancy, providing a more memory-efficient solution without compromising accuracy. Furthermore, SOLOv2 demonstrated superior performance over both the original SOLO model and Mask R-CNN, not only in terms of accuracy but also in speed.

CondInst CondInst [56], a novel approach, introduced a simpler architecture compared to SOLOv2. In CondInst, the categories' predictions and mask predictions are integrated into a single branch. The innovation lies in predicting masks conditioned

on the class. This change resulted in a configuration that is independent of object detection.

Furthermore, the mask prediction was adjusted to utilize only high-resolution feature maps, avoiding the need for all hierarchical feature maps and achieving high-resolution masks.

2.3 Loss Function for Object Detection and Instance Segmentation

In the domain of object detection, the choice of appropriate loss functions is crucial for training models effectively. These loss functions play a vital role in optimizing the parameters of the detection networks. This section will delve into the different loss functions employed for class classification, bounding box regression, and instance segmentation mask prediction in the context of object detection.

2.3.1 Classification Loss

In the realm of object detection, the class classification loss is designed to determine the likelihood of an object belonging to a specific category. Three common loss functions are discussed here.

SoftMax Cross Entropy Loss Historically, the RCNN family and SSD employed SoftMax Cross Entropy Loss. SSD adopted an approach to classification by adding background as one of the categories. On the other hand, Faster R-CNN uses Binary Cross Entropy Loss for the first-stage Region Proposal Network (RPN) and SoftMax Cross Entropy Loss for the second stage.

SoftMax is defined by the following equation:

$$p_c = \frac{e^{z_c}}{\sum_j e^{z_j}}, \tag{2.1}$$

where z_c represents the logit for class c.

Let $\boldsymbol{p} = [p_1, p_2, ..., p_C]^T$ be the value of Softmax output $p_i \in [0, 1]$ by the model, N the number of mini-batches in training and $\boldsymbol{t} = [t_1, t_2, ..., t_N]^T$ the number of correct labels $t_i \in 1, 2, ..., C$ in the mini-batches, then the SoftMax Cross Entropy Loss can be expressed as:

$$L_{\text{ce}} = -\sum_{i=1}^{N} \log(p_{t_i}). \tag{2.2}$$

Binary Cross Entropy Loss Binary cross entropy (BCE) loss is used in object detection frameworks like YOLO. This loss function is particularly suited for binary classification tasks and calculates the cross-entropy loss for binary class prediction

problems. According to the authors of YOLO, class classification in object detection is better framed as a two-class classification problem rather than a multi-class classification; YOLO uses a " One versus the other" structure for optimization.

In the context of binary classification models, the logits output z_i is further processed using a logistic function to obtain the predicted class probability p_i. Logistic function is defined by the following equation:

$$p_i = \frac{e^{z_i}}{e^{z_i} + 1}, \tag{2.3}$$

then the BCE Loss can be expressed as:

$$L_{\text{bce}} = -\frac{1}{N}\sum_{i=1}^{N}\{t_i \log(p_i) + (1 - t_i)\log(1 - p_i)\}. \tag{2.4}$$

Focal Loss The Focal Loss addressed class imbalance by assigning higher weights to hard-to-classify examples during training. Essentially, it "focused" on samples that had been misclassified in previous iterations, giving them more importance in the loss calculation. This idea effectively down-weighted easy negatives, thereby mitigating the problem of overwhelming background regions. In recent years, focal loss has gained prominence, primarily associated with models such as RetinaNet and FCOS. Focal loss addresses the issue of class imbalance by giving higher weights to hard examples, thereby improving training efficiency and focusing on challenging cases. Its formulation is as follows:

$$L_{\text{focal}} = -\frac{1}{N_{pos}}\sum_{i=1}^{N}\{t_i\alpha(1 - p_i)^{\gamma}\log(p_i) + (1 - t_i)(1 - \alpha)p_i^{\gamma}\log(1 - p_i)\}, \tag{2.5}$$

where, α_i and γ are hyperparameters controlling the weight distribution and the rate of focusing.

2.3.2 Bounding Box Regression Loss

Bounding box regression is crucial for localizing objects accurately. Different loss functions have been used to measure the discrepancy between predicted and ground truth bounding boxes. Some notable ones include:

L1 Loss L1 loss (also known as the Mean Absolute Error) was a predominant choice in early object detection methods [23]. It computes the absolute differences between predicted and true bounding box coordinates. The L1 loss is given by:

$$L_{\text{L1}} = \frac{1}{N_{pos}}\sum_{i=1}^{N_{pos}}\{\sum_{j\in\{x,y,w,h\}}\text{abs}(b^*_{i,j} - b_{i,j})\}, \tag{2.6}$$

where, $b^*_{i,j}$ represents the ground truth bounding box of instance i, and $b_{i,j}$ is the predicted bounding box.

Smooth L1 Loss Smooth L1 loss [22] is another commonly used regression loss function. L1 Loss has limitations, particularly in areas near zero where it lacks differentiability and maintains large gradients. To address this, "Smooth L1 Loss" was introduced. It combines the characteristics of Mean Squared Error (MSE) near zero and Mean Absolute Error (MAE) further away, creating a piecewise loss function proposed in Fast R-CNN. The smooth L1 loss is defined as:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{2.7}$$

However, in the actual object detection, the loss in box regression task is

$$L_{\text{smooth}} = \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} \sum_{j \in \{x,y,w,h\}} \{smooth_{L1}(b^*_{i,j} - b_{i,j})\}. \tag{2.8}$$

Both Faster R-CNN and YOLO have adopted the Smooth L1 Loss for bounding box regression.

IoU Loss While Smooth L1 Loss works well, it may not sufficiently consider the diversity of Intersection over Union (IoU) scores. IoU loss [48] has become a recent favorite in object detection, particularly in anchor-based detectors. It measures the discrepancy between predicted and ground truth bounding boxes by considering the IoU between the two. IoU loss is defined as:

$$L_{\text{IoU}} = \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} \{1 - \text{IoU}(\boldsymbol{b}^*_i, \boldsymbol{b}_i)\}, \tag{2.9}$$

where $\boldsymbol{b}^*_i$ represents the box coordinate vector of ground truth of instance i, and $\boldsymbol{b}_i$ represents the predicted box coordinate vector.

2.3.3 Mask Loss

The choice of an appropriate loss function plays a pivotal role in the training and optimization of instance segmentation models.

Binary Cross Entropy Loss Early approaches in instance segmentation predominantly relied on binary cross-entropy loss, where $p_{i,j}$ represents the predicted probability that a pixel at j belongs to a particular instance of i, and $t_{i,j}$ is the ground truth label.

$$L_{\text{bce}} = -\frac{1}{N_{pos} \times M} \sum_{i=1}^{N_{pos}} \sum_{j=1}^{M} \{t_{i,j} \log(p_{i,j}) + (1 - t_{i,j}) \log(1 - p_{i,j})\}, \tag{2.10}$$

where M is the total number of pixels in the target image. MASK R-CNN and YOLACT both employ BCE loss for mask prediction.

Dice Loss BCE was also significantly successful in the task of instance segmentation. However, these methods had limitations when dealing with images in which the background occupies many pixels. A significant milestone in the evolution of instance segmentation loss functions was the introduction of the Dice loss [53].

The Dice loss measures the overlap between predicted and ground truth segmentations and primarily concentrates on foreground labels. Dice loss is defined as:

$$L_{\text{dice}} = \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} \{1 - \frac{2\sum_{j=1}^{M} p_{i,j} t_{i,j}}{\sum_{j=1}^{M} (t_{i,j} + p_{i,j})}\}, \tag{2.11}$$

2.4 Datasets for Object Detection and Instance Segmentation

Commonly used datasets for object detection and instance segmentation are MS-COCO [39] and CityScapes [9]. Both play a pivotal role in the advancement of models and high standards for assessing the performance of these computer vision systems have been widely adopted.

Both provide extensive annotations for each image. It includes annotations for object detection, where each instance is annotated with a bounding box, and for instance segmentation, where each instance is further outlined with pixel-level masks.

2.4.1 MS-COCO

The COCO dataset is one of the most widely used benchmarks for object detection and instance segmentation tasks. It is known for its large-scale and diverse set of images that depict complex scenes, making it an excellent resource for evaluating the performance of computer vision models. COCO contains 118,287 images for training and 5,000 images for validation spanning 80 object categories.

The dataset encompasses a wide variety of images, including both indoor and outdoor scenes. This diversity poses a significant challenge to computer vision models due to the varying lighting conditions, object scales, and complex backgrounds.

Below are sample images from the COCO dataset:

2.4.2 CityScapes

The CityScapes dataset is another crucial benchmark, but it differs from COCO as it focuses on urban scenes and is primarily designed for semantic and instance segmentation in the context of autonomous driving. CityScapes is composed of images captured in urban settings, mainly streets and roads. This dataset aims to evaluate how well models can perform in real-world traffic scenes.

It offers fine-grained pixel-level annotations for semantic segmentation, making it valuable for tasks like road and lane segmentation. Additionally, it includes instance-level annotations for certain object classes.

Below are sample images from the CityScapes dataset:

FIGURE 2.6: Sample images from the COCO dataset with annotation.

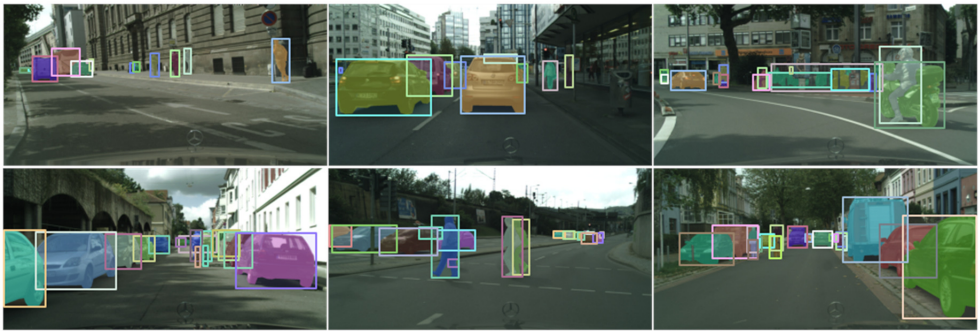


FIGURE 2.7: Sample images from the CityScapes dataset with annotation.

2.4.3 Evaluation Metric

In the realm of object detection and instance segmentation, the evaluation of model performance is of paramount importance. COCO and CityScapes datasets, a fundamental benchmark for these tasks, employ a range of evaluation metrics to assess the quality of results generated by algorithms. Among these metrics, Average Precision (AP), with variations including AP_{50}, AP_{75}, AP_S, AP_M, and AP_L, plays a central role in gauging detection and segmentation accuracy.

AP Average Precision serves as a comprehensive indicator of detection and segmentation performance. It calculates the area under the precision-recall curve, giving a clear sense of how well an algorithm can balance precision and recall. A higher AP signifies better accuracy in localizing objects and segmenting instances.

AP at Different IoU Thresholds: Intersection over Union (IoU) is a crucial parameter that dictates the extent of overlap between predicted bounding boxes and ground-truth instances. AP_{50} and AP_{75} evaluate performance at specific IoU thresholds of 0.50 and 0.75, respectively. AP_{50} provides an overview of the algorithm's

performance under moderate overlap, while AP_{75} emphasizes stricter matching criteria. Algorithms achieving higher AP_{50} and AP_{75} scores exhibit superior accuracy in localization and segmentation.

AP for Different Object Sizes: Objects within an image vary in size, from small to medium and large. AP_S, AP_M, and AP_L focus on evaluating performance for objects of different scales. AP_S represents the average precision for small objects with an area less than 32×32 pixels, AP_M for medium-sized ones that fall within the range of 32×32 to 96×96 pixels, and AP_L for large objects with an area exceeding 96×96 pixels. This segmentation enables a nuanced understanding of how well an algorithm caters to varying object sizes.

Chapter 3

Improved head and data augmentation to reduce artifacts at grid boundaries in object detection

3.1 Problems at Grid Boundaries

3.1.1 What is the Grid?

One-stage methods in object detection have gained prominence for their unique approach to handling images by partitioning them into a grid structure, enabling dense predictions across all grid cells [41, 46]. Unlike two-stage methods, which often deal with fixed-size inputs to their heads, one-stage methods take advantage of these grid cells to perform detection. This strategy offers several advantages.

One of the key benefits of one-stage methods is their adaptability to objects of various scales. To address multi-scale objects, one-stage models generate hierarchical feature maps with different resolutions. These feature maps are designed to detect objects at scales that match their sizes [37]. By using a coarse grid on low-resolution feature maps, these models can effectively detect larger objects, while employing a finer grid on high-resolution feature maps facilitates the detection of smaller objects. This flexible grid-based approach allows one-stage methods to maintain robustness across different scales, a notable advantage when compared to some two-stage methods that may struggle to detect small objects due to information loss from resizing [41, 46].

Furthermore, one-stage methods often employ heads built using Fully Convolutional Networks (FCN), which have yielded remarkable performance improvements in recent years [38, 14, 57]. These heads utilize stacked convolutional layers to expand the receptive field and obtain high-dimensional feature representations. This results in a more comprehensive understanding of the image and its contents, aiding in the accurate detection of objects across the grid structure.

3.1.2 What Is the Problem?

Recent research has brought to light the intriguing observation that Convolutional Neural Networks (CNNs) can experience a significant drop in classification accuracy due to even the slightest pixel translation [1, 15]. Consequently, there has been a surge in the development of various methods aimed at achieving shift-invariant feature representation [67, 5].

A noteworthy example comes from Manfredi et al. [43], who reported a noticeable decrease in object detection performance associated with minor translations. This revelation prompted a closer examination of the effects of translation on object detection and its implications.

In light of this context, both classification tasks and object detection are recognized to be susceptible to performance degradation due to minor translations. However, existing studies have primarily acknowledged this degradation without delving into the distinct factors contributing to it. They tend to treat the factors affecting object detection and classification as homogenous.

However, our investigation has unveiled a critical dissimilarity in the nature of degradation due to translations in object detection compared to classification. In particular, this degradation in object detection is shown to stem from artificially imposed grid boundaries, setting it apart from classification and underscoring unique characteristics intrinsic to object detection.

Of significant note is the susceptibility of object detection to misalignments in the input data. This vulnerability is particularly pronounced within the context of one-stage methods and hinges on a specific challenge a decrease in class scores attributed to the Euclidean distance between the object's location and the grid boundary. This difference highlights the distinctive features and challenges specific to object detection, particularly within the framework of one-stage methods.

This vulnerability assumes heightened significance in real-world scenarios. For instance, in surveillance applications, the prospect of object detection failing when an intruder undergoes slight translations within the frame presents a significant security risk. Similarly, within the domain of Advanced Driver Assistance Systems (ADAS), where pedestrian detection plays a pivotal role, the potential for minor movements of pedestrians to result in inconsistent detection outcomes is far from ideal. This realization underscores the critical importance of addressing and mitigating the issue of shift variance in object detection, particularly concerning one-stage methods, to ensure the reliability and robustness of these systems across various applications.

3.1.3 Previous Methods for Shift Invariance

Grid-Based One-Stage Object Detector

Recent one-stage object detection methods have achieved remarkable success using multi-scale feature pyramids on grids. EfficientDet [55] employs a multi-scale feature representation by adding a bottom-up path in addition to the top-down path. This

method uses conventional feature pyramid networks (FPN) [37] to propagate feature maps from low to high resolution. YOLOv4 [2] uses SPP [29] to obtain multi-scale feature representations by simultaneously using pooling layers with multiple kernel sizes $1, 5, 9$, and 13 to collect both local and global information.

Furthermore, the feature extraction part of the head of recent one-stage methods [57, 38, 55, 21] obtains features from adjacent grid cells by expanding the receptive field using multiple 3×3 convolution layers of feature maps from the FPN.

However, in evaluating robustness to shift in FCOS [57], we find that the class score drops when the object center aligns with the grid boundary of the feature map. This means that the FPN cannot directly improve feature representation at the grid boundaries, and the stacking of multiple convolutional layers is implicit and insufficient for information propagation at grid boundaries. Thus, it is necessary to introduce further ingenuity.

Shift Invariance and Equivalence

In the classification task, current CNN-based methods are reported as not shift invariant [15]. One approach to obtain shift-invariant models is data augmentation, such as random cropping, but the improvement is reported to be limited to similar images [1].

The research on shift perturbations in object detection tasks is limited. Manfredi et al. [43] noted that object detection is applied to safety-critical applications, such as autonomous driving, which requires equivalence, not invariance, for the shift. They proposed a method to evaluate the robustness to shift in object detection, which is the first step in consideration of shift equivalence, and confirm an improvement in robustness due to down-sampling [67]. Nevertheless, it shows a poorer performance as a general benchmark evaluation method.

Thus, it is necessary to develop new data augmentation techniques by which data are generated with adequate shifts depending on the target size.

SwinTransformer

Vision Transformer (ViT) [13] divides an image into multiple patches and obtains the feature values for each patch after calculating the relationship between the patches. Therefore, when applied to tasks such as object detection and segmentation, where detailed features of images should be considered, a smaller patch size increases the number of patches and the number of combinations between patches, significantly increasing the computational complexity during learning and inference.

Swin Transformer [42] addresses this problem by dividing patches into multiple groups called “windows” and limiting the calculation of the relationship between patches to within windows, thereby reducing the amount of computation during learning and inference. In this case, the target object is divided by a window, and the

relationship between originally adjacent and highly related patches may not be considered. Therefore, Swin Transformer extracts features at the boundary of a window by adding a Shifted Window to calculate the degree of relationship between patches.

This suggests the importance of introducing information from neighboring grids.

3.1.4 Reducing Class Scores in Object Detection at Artefactual Grid Boundaries

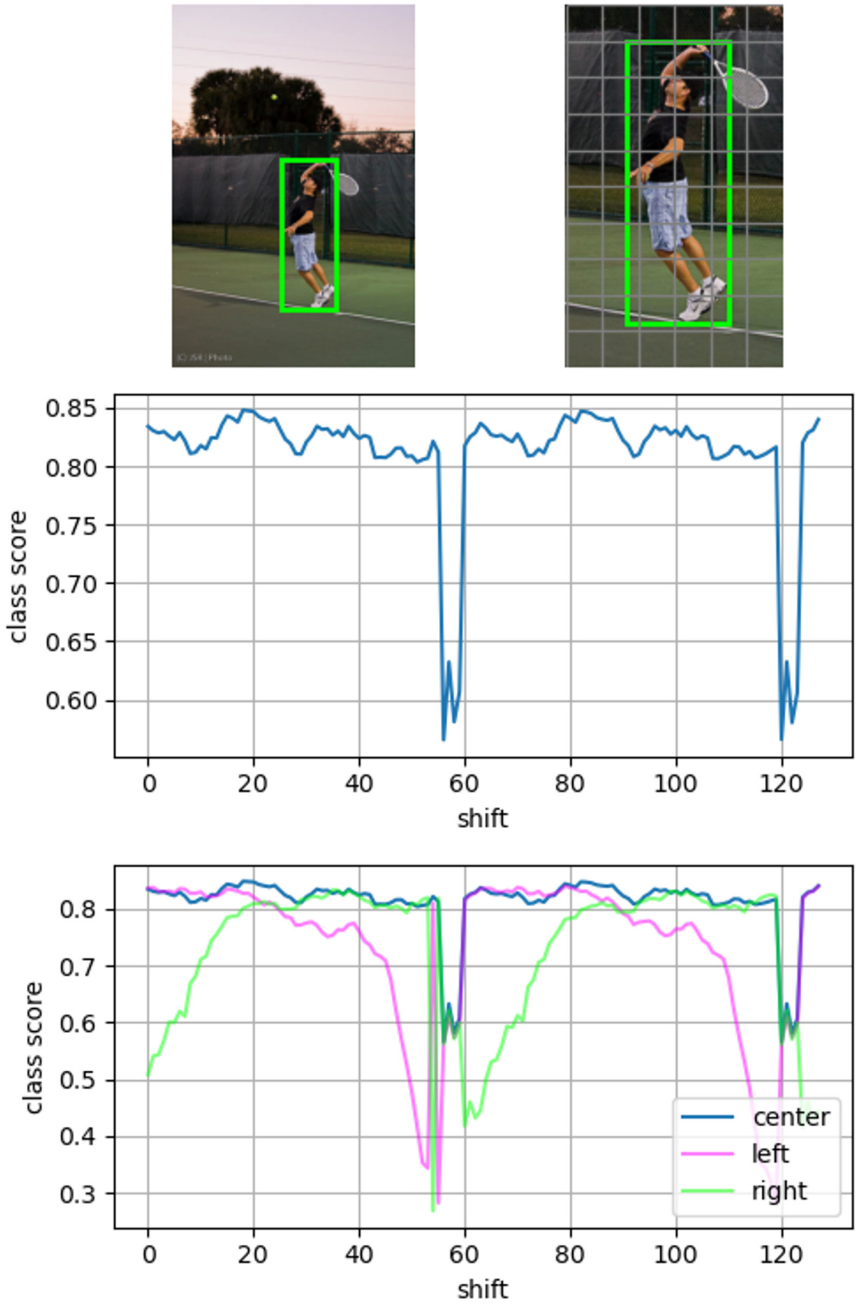


FIGURE 3.1: Effect of image shift on class scores. Results of FCOS inference of the image from the COCO train set. The first row shows the image and its ground truth. The second row shows the variation of the class score of the correct label when the image is horizontally translated; the third row shows the scores of the object's center (blue line), right neighbor (green line), and left neighbor (pink line) grid cell.

This section shows our observations of class scores dropping at the boundaries with adjacent grids when the input image is shifted. Then, we propose solutions to this problem.

In object detection, the kernel sizes of convolutions in both the network backbone and FPNs [37] are designed to consider the down-sampling strides. Moreover, the head is composed of enough convolution layers to construct a sufficiently large receptive field that considers the size of the targets. Therefore, the prediction results are expected to be robust to the object's position in the grid.

Let $\boldsymbol{f}_j^{\mathbf{0}}$ be the grid cell of the feature vector containing the center of the object j in the original input image I^0, and let I^{δ_x} and $\boldsymbol{f}_j^{\delta_x}$ be the image and feature vector of I^0 translated horizontally by δ_x pixels, respectively. Then, the classifying operation H_{cls} at the head is expected to be $H_{cls}(\boldsymbol{f}_j^{\mathbf{0}}) = H_{cls}(\boldsymbol{f}_j^{\delta_x})$.

We performed experiments to investigate the variations of the prediction results for each $H_{cls}(\boldsymbol{f}_j^{\delta_x})$ when the image I^0 is translated horizontally. Fig. 3.1 shows graphs of the variations obtained for a target object in an input image.

The first column of Fig. 3.1 shows the input image on the left and the zoomed-in image of the object to be detected with grids on the right. The second column shows the class scores of the grid cell containing the center of the ground-truth box when the input image is translated horizontally to the right. From this graph, we can see that the class scores are changing periodically depending on the shift value. The range of scores varies from 0.85 to 0.6. Also, from this graph, we can see that the score becomes minimum at the position where the center of the bounding box overlaps the boundary of the grid. The period of the scores is equal to the grid size of the feature map, which corresponds to the scale of the target object. In this example, the grid size assigned to the target object is 64 px. We observed that the same phenomenon occurs for other targets and,

a larger target object (assigned to a large grid size) tends to show larger drops in the class scores. This phenomenon in large objects suggests that the current one-stage object detection methods cannot extract sufficient information from adjacent grid cells using the implicit receptive field expansion method with convolution layers.

The bottom graph in Fig. 3.1 shows the class scores of the center (blue), right neighbor (green), and left neighbor (pink) grid cells when horizontal shifts are applied to the input image. From this graph, we can see that the class scores of the center grid cell suddenly drop at the horizontal axis position 55, which corresponds to the boundary of the grid for that target. Meanwhile, the class scores of the left neighbor grid cell gradually decrease and become minimum at horizontal axis position 55. Then, they suddenly increase when the shifts become larger than 55. Also, the right neighbor cell behaves oppositely to the left neighbor cell.

These results suggest that the class scores of the left cell become high when the target center is close to the left boundary, and vice versa for the right side. However, at grid boundaries, information for target classification is not well propagated between

adjacent grids. Thus, class scores at the grid boundary become low in all three grid cells.

This means we should train the classification head using the neighboring grid information. To improve the training further, we should train the head with the augmented data generated by adding the horizontal and vertical shifts of the input image depending on the target scale and location.

3.2 Proposed Method

3.2.1 Overview of Our Proposed Method

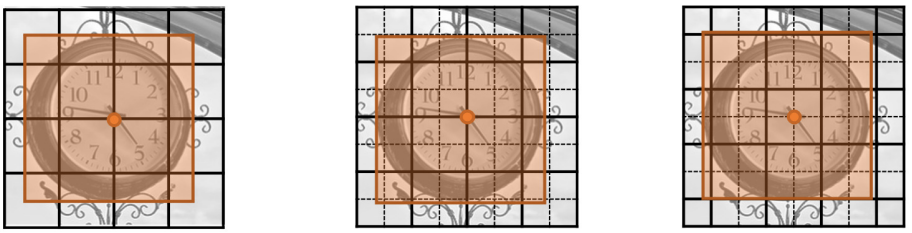


FIGURE 3.2: Location relationship between the grid boundary and object center., where the class score drops, and our proposed feature representation strategy. The orange circle represents the object center. The black lines represent the feature map grid. FCOS drops class scores when the grid boundary of the feature map aligns with the object center. (B) The feature map is one level higher resolution than the original feature map. (C) The feature map of (B) is shifted by one grid to shift the object center from the grid boundary to the center of the grid.

The vulnerability at the grid boundary arises when the bounding box center associated with an object coincides with the grid boundary, as illustrated in Fig. 3.2a. This issue becomes more prevalent when dealing with larger objects. Large objects often span multiple grids, necessitating the exchange of information between these grids for accurate detection. However, pixels near the grid boundaries exhibit disparate features inside and outside the grid, making precise object information estimation challenging. This discrepancy leads to a drop in class scores at the grid boundaries.

Concurrently, while down-sampling techniques have been instrumental in mitigating aliasing effects and preserving detailed information within feature maps, they fall short of completely resolving the challenges posed by grid boundaries.

In light of these challenges, this study introduces two novel methods aimed at ameliorating the decrease in class scores around the grid boundary and enhancing the overall robustness of trained networks, particularly concerning target locations. The first method, the Sub-Grid Feature Extraction Module (SGFEM), is integrated into the network's head. SGFEM augments the original feature map with a feature map at one-level higher resolution (as depicted in Fig. 3.2b) and a feature map shifted

by one grid (as shown in Fig. 3.2c). This process results in a new feature map that compensates for the information loss at the grid boundary.

The second method, Grid-Aware Data Augmentation (GADA), presents a data augmentation technique designed to shift the object's center, with a specific focus on the grid boundary's vulnerable points. The extent of translation is contingent on the size of the target objects.

By seamlessly integrating these two innovative approaches into the Fully Convolutional One-Stage Object Detection (FCOS) framework [57] and applying them rigorously, this research showcases substantial performance improvements, particularly on the COCO validation set [39].

3.2.2 Network Architecture

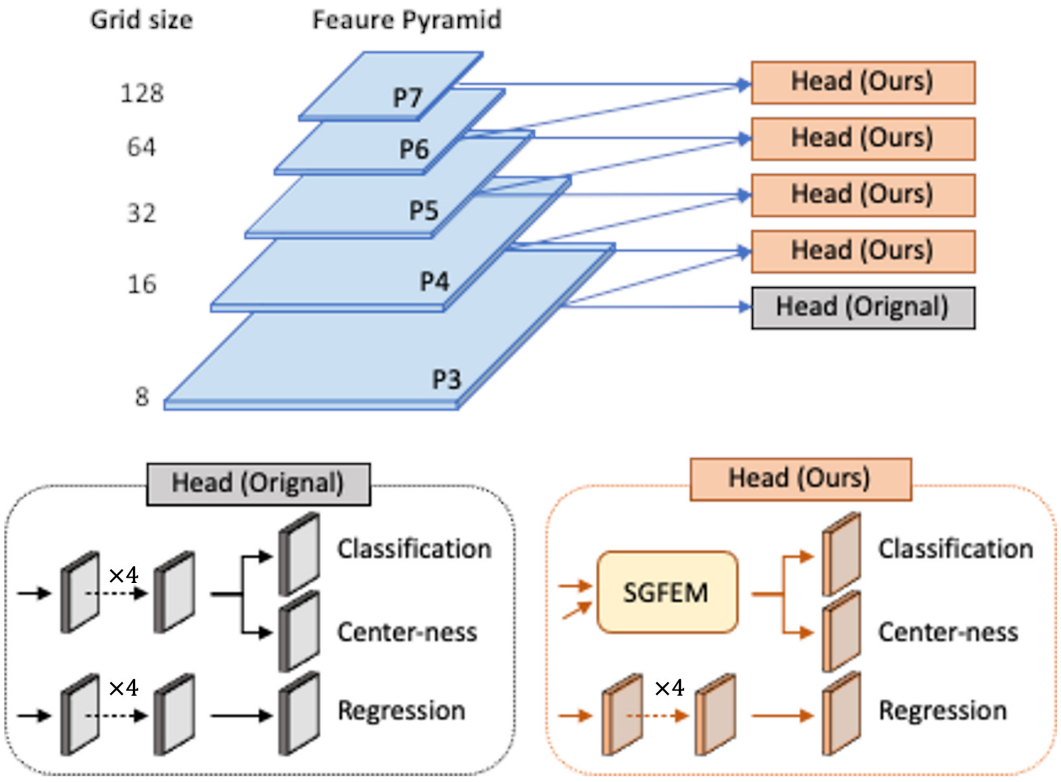


FIGURE 3.3: Overview of the network architecture of the proposed method. P3-P7 generated from the feature pyramid are the feature maps used for the final prediction. The grid size represents the step size width of the feature map. The head is a twin-head configuration of FCOS's original head and Our Head, where Our Head is a shared head between feature levels.

We implemented and investigated FCOS [57]. FCOS generates 5-level feature maps with different resolutions in the feature extractor. Let $P_i \in \{P_3, P_4, P_5, P_6, P_7\}$ be the feature map, where i denotes the layer index of the feature map. P_3, P_4, and P_5 are connected top-down with a convolution layer of 1×1 kernel from the backbone, and P_6 and P_7 are generated by applying a convolution layer of stride 2 to P_5 and P_6,

respectively. The grid cell size is $[8, 16, 32, 64, 128]$ for P_3, P_4, P_5, P_6, P_7, respectively. The receptive field M implicitly expanded by the four 3×3 convolution layers of the head is $[32, 64, 128, 256, 512]$.

The target object is assigned the feature map level with rules based on the maximum distance of the box regression. The rule is to assign positive labels to areas where the maximum distance of regression is within the receptive field M of each level. If positive labels overlap between feature maps, the lower resolution is employed, and the higher resolution label is redefined as a negative label.

3.2.3 Sub-Grid Feature Extractor

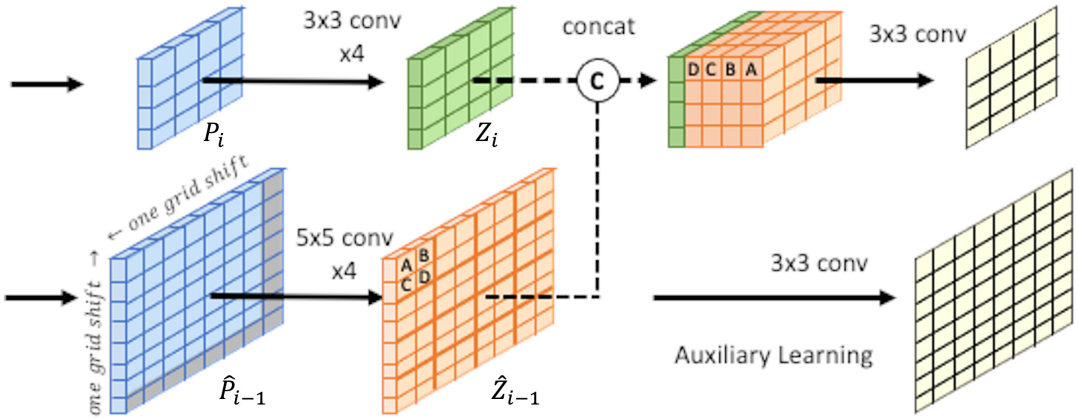


FIGURE 3.4: Illustration of SGFEM. The upper pass is the feature extractor from P_i in the regular head of FCOS, and the lower pass is the proposed sub-grids $\hat{P}_{i-1}$ one. The final layer outputs predictions for each task from the feature map, which fuses the features generated from each pass.

Fig. 3.3 presents an overview of the SGFEM. The network propagates the feature pyramid generated from the backbone to the head. It predicts bounding boxes and categories from the feature map that fuses the original and sub-grid features using SGFEM. For the highest resolution feature map $P3$, we apply the original FCOS head. For the other feature maps, we apply our proposed head.

Our head is fed two feature maps with different resolutions. SGFEM performs feature extraction from feature maps of these two resolutions, considering features at the grid boundaries. As shown in Fig. 3.4, SGFEM contains three parts.

The feature map P_i is embedded in Z_i using a 3×3 convolution layer, as in FCOS. Higher resolution features $\hat{P}_{i-1}$ are embedded in $\hat{Z}_{i-1}$ using a 5×5 convolution layer. In this case, $\hat{P}_{i-1}$ is a feature map with P_{i-1} shifted by one grid and can be regarded as a feature representation shifted by half a grid compared to P_i (Fig. 3.2c). Then, to fuse Z_i and $\hat{Z}_{i-1}$ features, the feature map of $\hat{Z}_{i-1}$ is re-organized into 4 channels, as denoted by $\{A, B, C, D\}$ in Fig. 3.4, aligned to the resolution of Z_i, and fused using

concatenation operations. Each branch consists of 3×3 convolution layers, the same as FCOS, to predict bounding boxes and categories from fused features.

3.2.4 Auxiliary Loss

The original FCOS loss function is defined as follows:

$$L_{FCOS} = \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(\boldsymbol{p}_{x,y}, c^*_{x,y}) + \frac{1}{N_{pos}} \sum_{x,y} \mathbf{1}_{\{c^*_{x,y}>0\}} L_{box}(\boldsymbol{b}_{x,y}, \boldsymbol{b}^*_{x,y}), \tag{3.1}$$

where L_{cls} is the focal loss [38] and L_{box} is the IoU Loss [48]. N_{pos} defines the number of positive samples. Let $\boldsymbol{b}^*_{x,y}$ denote the ground-truth box at position (x,y) on the feature map Z_i and its class label $c^*_{x,y}$, and let $\boldsymbol{b}_{x,y}$ and $\boldsymbol{p}_{x,y}$ denote the prediction results, respectively.

Here, we provide an auxiliary loss because the feature $\hat{Z}_{i-1}$ leads to a more stable optimization. For the auxiliary loss, we also assign a positive label $\hat{c}^*_{x,y}$ to the sub-grid high-resolution feature map $\hat{Z}_{i-1}$ and define it as follows:

$$L_{aux} = \frac{1}{\hat{N}_{pos}} \sum_{x,y} L_{cls}(\hat{\boldsymbol{p}}_{x,y}, \hat{c}^*_{x,y}), \tag{3.2}$$

where $\hat{N}_{pos}$ defines the number of positive samples on $\hat{Z}_{i-1}$. Let the category label and prediction result be $\hat{c}^*_{x,y}$ and $\hat{\boldsymbol{p}}_{x,y}$ respectively.

The final loss function is formulated as follows:

$$Loss = L_{FCOS} + \lambda L_{aux}, \tag{3.3}$$

where λ is a hyper-parameter to balance the auxiliary and original loss.

In the testing phase, the auxiliary branches are abandoned. Thus, the auxiliary loss path does not add any extra parameters or calculations to the model in inference.

3.2.5 Grid-Aware Data Augmentation

Let $\boldsymbol{b}^*_{x,y} = (l,t,r,b)$ be the ground-truth box assigned to object j at position (x,y) on feature map P_i, where (l,t,r,b) is the offset from the center of the grid cell to the four sides of the bounding box.

The value of the image shift $(\Delta x_j, \Delta y_j)$ to the grid boundary where the class score drops can be defined as

$$\Delta x_j = \frac{r+l}{2} - xs_i, \tag{3.4}$$

$$\Delta y_j = \frac{b+t}{2} - ys_i, \tag{3.5}$$

where s_i is the grid size of the feature map P_i.

In training, we generate an image shifted by $(\Delta x_j + \delta_x, \Delta y_j + \delta_y)$ pixels with probability δ to provide randomness. This experiment used $\delta_x, \delta_y \in [-8, 8]$.

3.3 Experiments

TABLE 3.1: Comparison of baseline on COCO val dataset. "Ours" represents the head with SGFEM. GADA represents Grid-Aware Data Augmentation.FPS measured in runtime on a GeForce RTX3090.

method	backbone	params(M)	FPS	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS	ResNet-18	19.1	43.2	155.6	33.3	51.6	39.7	18.8	35.8	42.8
FCOS+GADA	ResNet-18	19.1	43.2	155.6	34.2	52.6	39.8	19.1	37.0	44.4
SGFEM	ResNet-18	26.6	33.9	198.3	33.5	52.6	39.6	17.6	36.3	44.0
SGFEM+GADA	ResNet-18	26.6	33.9	198.3	**35.0**	**53.9**	**40.1**	**19.6**	**37.9**	**46.5**
FCOS	ResNet-50	32.0	29.5	200.6	36.3	55.4	38.4	19.6	39.6	47.8
FCOS+GADA	ResNet-50	32.0	29.5	200.6	38.2	57.3	**40.9**	**22.6**	42.0	48.9
SGFEM	ResNet-50	39.5	24.8	243.2	36.5	55.8	38.6	19.9	39.3	48.6
SGFEM+GADA	ResNet-50	39.5	24.8	243.2	**38.5**	**58.2**	**40.9**	22.1	**42.2**	**49.6**
FCOS	ResNet-101	51.0	22.1	276.6	39.4	58.8	41.9	22.6	43.3	51.6
FCOS+GADA	ResNet-101	51.0	22.1	276.6	**40.3**	**59.8**	**43.1**	**24.0**	**44.6**	51.3
SGFEM	ResNet-101	58.5	19.3	319.3	39.4	59.1	41.7	23.2	43.4	51.6
SGFEM+GADA	ResNet-101	58.5	19.3	319.3	40.0	59.7	42.9	23.5	43.8	**51.7**

3.3.1 Dataset and evaluation protocol

The COCO [39] benchmark is a large-scale object detection, instance segmentation, and image captioning dataset with 80 categories. Following standard practice, the COCO train2017 split (115k images) is taken as the training set and the val2017 split (5k images) as the validation set.

We use the COCO API to measure the average precision (AP) for IoUs in the range 0.5:0.05:0.95. We also checked the breakdown of AP for small (AP_S; area $\leq 32^2$), medium (AP_M; $32^2 <$ area $\leq 96^2$), and large (AP_L; area $> 96^2$) objects.

3.3.2 Experimental settings

In this section, we describe the experimental settings. For a fair comparison, the same learning settings are used in the baseline and the proposed method. The details of the settings are shown below.

We use ResNet-18/50/101 [26] as the backbone network, with weights pre-trained by ImageNet [12] for initialization. The weights of the newly added layers are initialized by the Kaiming initialization [27].

The network is trained by stochastic gradient descent (SGD) for optimization, with an initial learning rate of 0.01, 16 mini-batches, and 24 total epochs. The learning rate is reduced by 10 at the 16th and 22nd epochs. The weight decay and the momentum are set as 0.0001 and 0.9, respectively.

Input images are resized to a maximum scale of 1333×800, without changing the aspect ratio. Only random horizontal image flipping is used for data augmentation. The baseline author's experimental results report that the number of learning epochs is sufficient at 12, regardless of backbone size, and changes to 24 epochs when applying multiscale augmentation. Therefore, we compared the number of learning epochs when we applied GADA with 12 and 24 and confirmed that 24 epochs are more effective. We believe that this is because the augmentation's randomness increases the input data's variation. So, we conduct all experiments with 24 epochs.

The hyper-parameters used in the proposed method are as follows: GADA application ratio $\alpha = 0.5$ and original and auxiliary loss balance adjustment $\lambda = 1.0$.

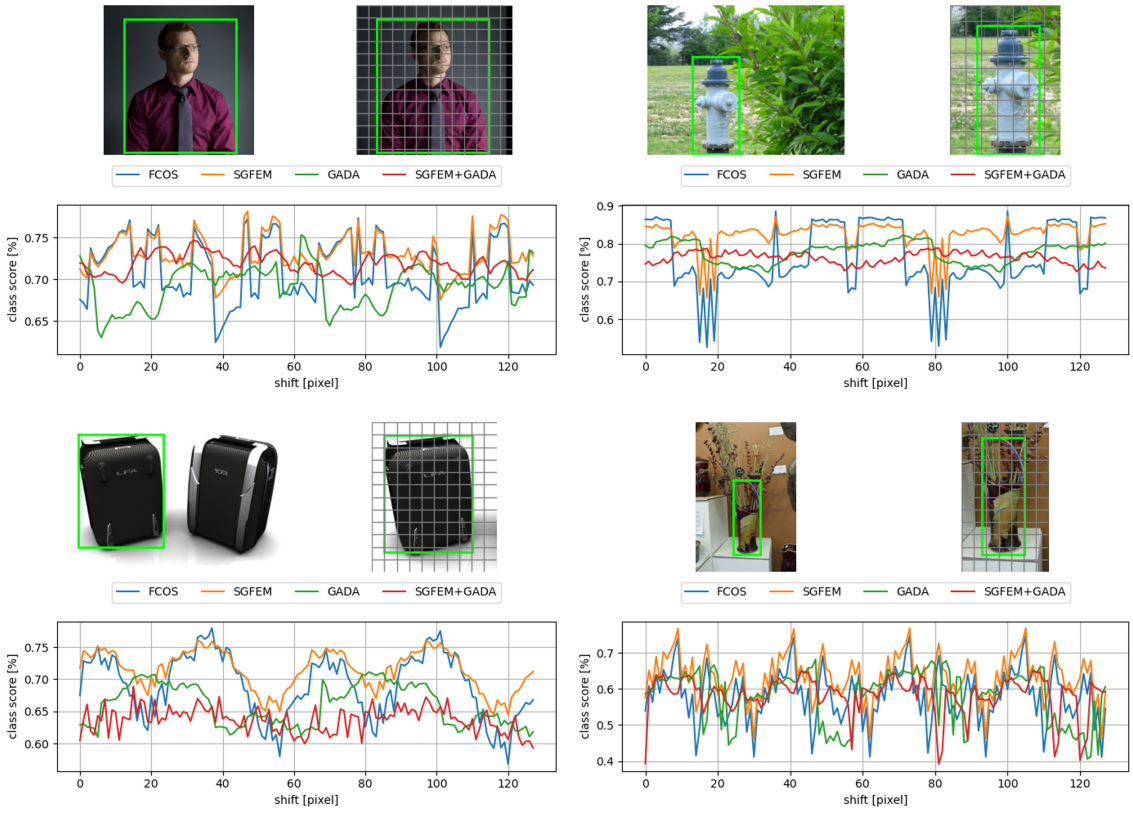


FIGURE 3.5: Comparison of robustness to shift on COCO val dataset. Four samples plotted images with ground-truth boxes on the target and variance of the class score relative to the shift. The top row shows cases where the proposed method improves the results. The lower row shows the failure cases.

3.3.3 Comparison With Baseline Method

We compare our proposed method with the baseline FCOS in object detection.

When ResNet-18 is used as the backbone, our proposed method shows improvement: adopting SGFEM results in an improvement of only $0.2AP$ in the typical metric and $1.2AP_L$ in the metric for large objects. This demonstrates that our head is more effective for feature maps with large grid sizes, where large objects are assigned.

In addition, a comparison with and without Grid-aware Data Augmentation shows improvements for both the baseline and proposed method. Data augmentation improves in all evaluation metrics regardless of the size of the object. We believe this is because features can be successfully extracted from two feature maps with different grid partitions.

Combining SGFEM with data augmentation yields improvements of $1.7AP$ and $3.7AP_L$.

Similarly, using ResNet-50 for the backbone gives improvements of $2.2AP$ and $1.8AP_L$.

When we apply ResNet-101 to the backbone, there is no improvement via SGFEM. This trend is attributed to the higher ability of feature extraction in deeper and more complex networks. Thus, we believe that deep networks can extract multi-resolution features without relying on SGFEM. On the other hand, we reach the highest accuracy when GADA is applied, yielding $0.9AP$ and $1.4AP_S$ and $1.3AP_M$ improvements. The combination of SGFEM and GADA yields an improvement of $0.1AP_L$ for large objects.

The network size increases by 7.5M parameters by adding the head of SGFEM. Therefore, the number of parameters increases by a factor of 1.39 for ResNet-18, 1.23 for ResNet-50, and 1.14 for ResNet-101. In addition, an increase of 42.69 GFLOPs in computational cost results in processing speeds of 33.9 fps for ResNet-18, 24.8 fps for ResNet-50 and 19.3 fps for ResNet-101 on the GeForce RTX3090.

3.3.4 Impact on Grid Boundaries

A comparison of the baseline and proposed method for class score drop at grid boundaries is shown in Fig. 3.5. The vertical axis shows the class score, and the horizontal axis shows the shift value. The blue, orange, green, and red lines represent the results obtained by the baseline method, applying SGFEM, applying Grid-aware Data Augmentation, and applying both SGFEM and Grid-aware Data Augmentation, respectively.

The top row of Fig. 3.5 shows successful improvement cases. Comparing the results of applying only SGFEM to the baseline shows an improvement specifically in grid boundary drop. Comparing the baseline with only Grid-aware Data Augmentation, the overall behavior changes with respect to the shift. We assume that this is due to the fact that the network is encouraged to obtain information from various locations in the grid. When both SGFEM and Grid-aware Data Augmentation are applied, the drop of the grid boundary is improved, and the effect is more robust to shifts.

The bottom row is a failure case. The left figure shows a case where equality is obtained but the highest class score is lower than the baseline. The right figure shows a case where the grid boundary drop cannot be improved.

TABLE 3.2: Comparison of Data Augmentation Methods on the COCO val set.

Head	Augmentation	AP	AP_S	AP_M	AP_L
FCOS	-	33.3	18.8	35.8	42.8
FCOS	Random Shift	**34.2**	18.8	**37.2**	43.9
FCOS	Random Crop	33.4	18.5	36.5	43.0
FCOS	GADA	**34.2**	**19.1**	37.0	**44.4**
SGFEM	-	33.5	17.6	36.3	44.0
SGFEM	Random Shift	34.1	18.4	37.2	44.6
SGFEM	Random Crop	33.4	18.6	36.6	43.5
SGFEM	GADA	**35.0**	**19.6**	**37.9**	**46.5**

3.3.5 Comparison with the Standard Data Augmentation Methods

Table 3.2 shows the comparison of our proposed data augmentation method, GADA, with commonly used data augmentation methods with random shift and random crop. The data augmentation application ratio and shift range are unified at 0.5 and $[-8, 8]$, respectively. The experiment employs Resnet-18 as the network backbone. The top rows of Table 3.2 show the FCOS results and the second rows show the SGFEM results. FCOS shows improvement in GADA and random shift. GADA also shows the highest performance for AP_S and AP_L. Furthermore, results applied to SGFEM show that GADA has the highest performance.

This indicates that GADA focusing on the grid boundaries achieves robustness more effectively.

3.3.6 Quantitative evaluation on grid boundaries

To quantitatively evaluate the identified weaknesses at the grid boundaries, we conducted an evaluation. For this assessment, we applied the GADA method to the COCO validation dataset, generating a new dataset for thorough analysis. In particular, we set δ_x, δ_y to zero, randomly chose objects when multiple objects coexisted within a single image, and shifted the entire image to ensure the selected objects overlapped with the grid boundary.

The outcomes of this evaluation are depicted in Table 3.3, presenting results for both the original COCO dataset and the newly created dataset that underwent the applied shift.

The findings revealed a clear distinction between the performance on the original dataset and the shifted dataset. Notably, the performance on the shifted dataset exhibited a noticeable decrease compared to the original dataset, highlighting the vulnerability associated with grid boundaries.

TABLE 3.3: Comparison results on a dataset created from COCO-val images by transforming the images to overlap the centre of the object and the grid boundaries.

method	backbone	original-coco *AP*	shifted-coco *AP*
FCOS	ResNet-18	33.3	31.6
FCOS+GADA	ResNet-18	34.2	32.2
SGFEM	ResNet-18	33.5	31.9
SGFEM+GADA	ResNet-18	**35.0**	**34.0**
FCOS	ResNet-50	36.3	35.3
FCOS+GADA	ResNet-50	38.2	37.5
SGFEM	ResNet-50	36.5	35.0
SGFEM+GADA	ResNet-50	**38.5**	**37.7**
FCOS	ResNet-101	39.4	37.2
FCOS+GADA	ResNet-101	**40.3**	39.4
SGFEM	ResNet-101	39.4	37.2
SGFEM+GADA	ResNet-101	40.0	**39.2**

These results substantiate the presence of weaknesses at grid boundaries, emphasizing the importance of addressing this issue in object detection models. Furthermore, the experiments also demonstrated that the proposed method, a combined model utilizing both GADA and SGFEM, outperforms other configurations.

3.3.7 Ablation Study

We verify the effectiveness of each component of our proposed method. For ablation, we use the ResNet-18 backbone and report the performance on the COCO val set.

Grid-Aware Data Augmentation

TABLE 3.4: Analysis of different hyper-parameters for data augmentation ratio α on the COCO val set.

α	AP	AP_S	AP_M	AP_L
0.0	33.3	18.8	35.8	42.8
0.1	33.7	**19.4**	36.7	43.1
0.3	34.1	**19.4**	**37.0**	43.8
0.5	**34.2**	19.1	**37.0**	44.4
0.7	**34.2**	18.7	36.9	**45.0**

We compare the effect of the application rate α of the GADA on performance in Table 3.4.

The highest performance is achieved when the ratios of data augmentation α are 0.5 and 0.7, which show an improvement of $0.9AP$ compared to without data

augmentation applied $\alpha = 0.0$. AP_L improves as α increases, meaning that large objects are more sensitive to grid boundaries.

SGFEM

TABLE 3.5: Analysis of different head configurations for SGFEM on the COCO val set. "cls" , "box" , and "ctr" indicate branches of class classification, box regression, and centerness, respectively.

branch						
cls	box	ctr	AP	AP_S	AP_M	AP_L
			33.1	17.9	35.2	44.4
✓			**33.7**	17.9	**35.6**	45.6
	✓		33.1	17.9	35.2	44.4
		✓	33.1	17.9	35.2	44.4
✓	✓		**33.7**	17.9	**35.6**	**45.7**
	✓	✓	33.1	17.9	35.2	44.4
✓		✓	**33.7**	17.9	**35.6**	**45.7**
✓	✓	✓	**33.7**	17.9	**35.6**	**45.7**

The FCOS head consists of three branches: classification, centerness, and box regression, and our proposed SGFEM can be employed for all branches. Table 3.5 shows the combination results for branches applying SGFEM. This table shows the effectiveness of applying SGFEM to the classification branch. It also shows that it has little effect on the box regression branch.

The configuration with the highest performance applies SGFEM to the classification and centerness branches but not to the box regression branch.

Auxiliary Learning

TABLE 3.6: Analysis of different hyper-parameters for auxiliary loss weight λ on the COCO val set.

λ	AP	AP_S	AP_M	AP_L
0.0	33.1	**18.3**	34.8	**44.1**
0.5	33.4	18.1	35.5	**44.1**
1.0	**33.5**	17.6	**36.3**	44.0
1.5	33.4	17.5	**36.3**	43.6
2.0	33.0	17.4	35.5	43.7

Table 3.6 compares the performance for different values of the hyper-parameter λ, which adjusts the balance between the original and auxiliary losses shown in Equation3.3. The highest performance is shown when $\lambda = 1.0$. The improvement with

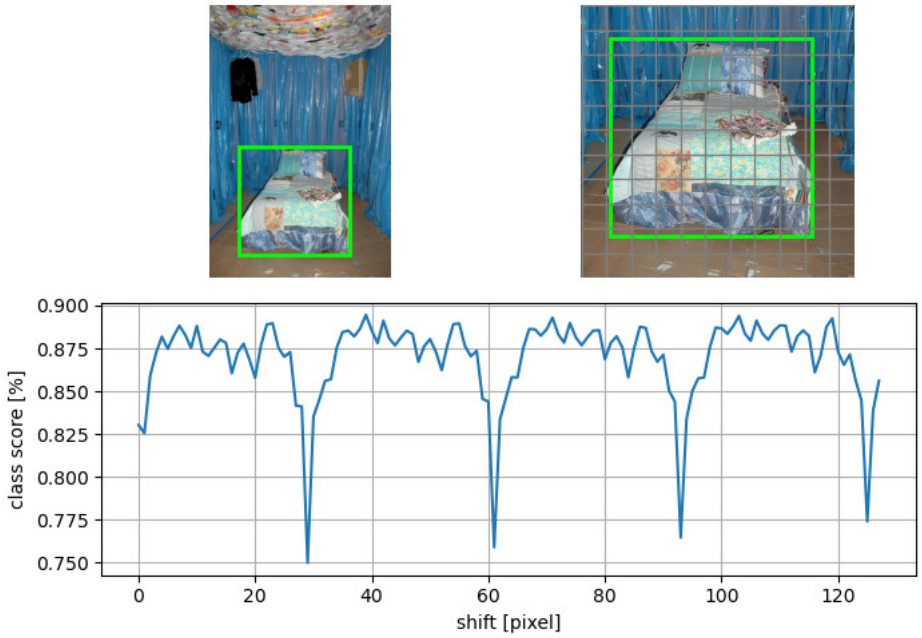


FIGURE 3.6: Plot of class scores for a small horizontal translation of the image using YOLOX.

auxiliary learning ($\lambda > 0.0$) compared to without auxiliary learning ($\lambda = 0.0$) indicates its effectiveness.

3.3.8 Experimental Application to Yolox

Reproducibility Study in Yolox

YOLOX [21] is a fast, high-performance one-stage object detection method that modifies YOLOv3 [45]. In the YOLO family, YOLOX is the first anchor-free system with a head similar to FCOS.

For the multi-scale object detection problem, the network employs Spatial Pyramid Pooling (SPP) [29] and Path Aggregation Network (PAN) [40] for the backbone and neck, respectively, to obtain feature representation from multiple resolutions. In addition, the data augmentation method employs Mosaic [2] and Mixup [66] to improve robustness.

There are four versions of the YOLOX architecture: YOLOX-S (small), YOLOX-M (medium), YOLOX-L (large), and YOLOX-X (extra large).

We experimented with YOLOX-S to investigate the variation in class scores when the image is translated horizontally. Fig. 3.6 shows the result of the variation of the class score for the target object in the input image.

The first row in the figure shows the image of the observed object, and the second row shows the variation of the class score when the input image is horizontally transformed in the right direction. The plot shows that the class score varies cyclically with shift values in YOLOX-S as well as FCOS. The score period is equal to the grid

size of the feature map, and the range of class scores varies from 0.89 to 0.75. This result suggests that even with the improvement of networks such as SPP and PAN modules and the data augmentation such as Mosic used in YOLOX, it is weak at extracting features at the grid boundaries.

Therefore, we apply our two proposed improvement methods for extracting grid boundary features, SGFEM and GADA, to YOLOX to confirm their effectiveness.

Network Modification Details

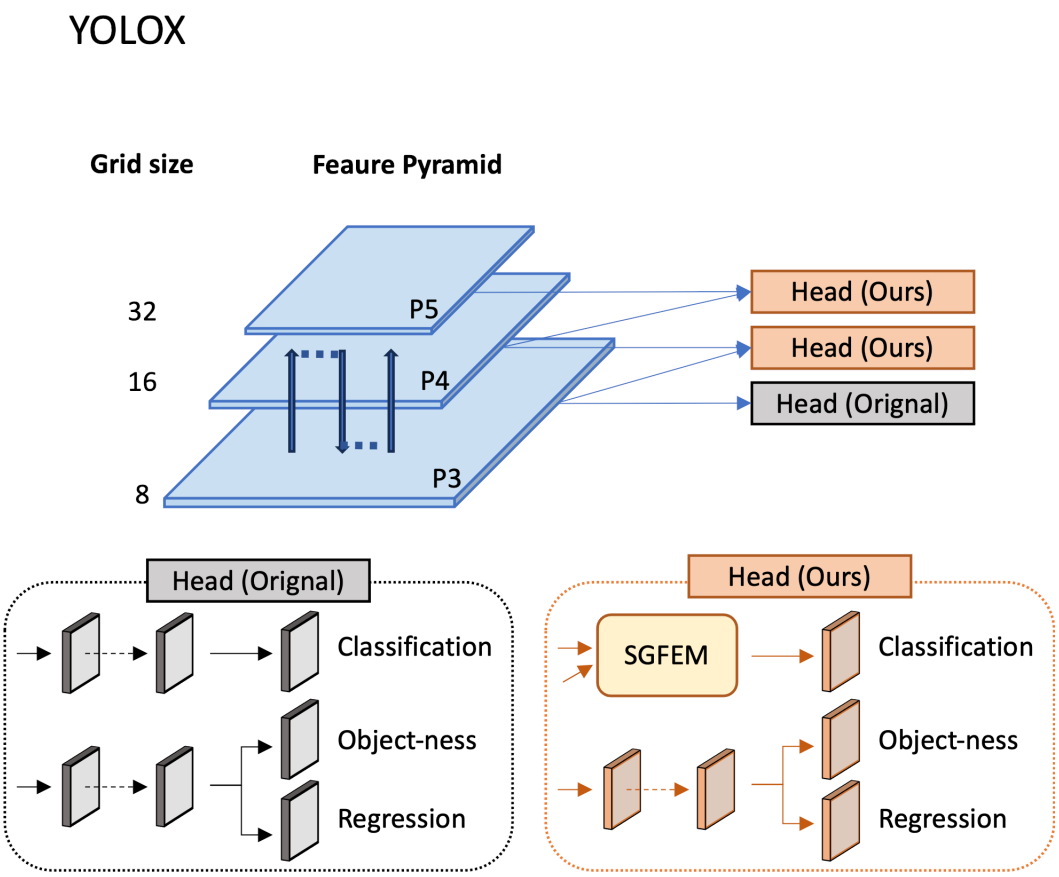


FIGURE 3.7: Overview of the network architecture of YOLOX with the proposed method. P3-P5 generated from the feature pyramid are the feature maps used for the final prediction. The grid size represents the step size width of the feature map. The head is a twin-head configuration of YOLOX original head and Our Head, where Our Head is a shared head between feature levels.

The backbone of YOLOX is Darknet53, a new network architecture that focuses on feature extraction characterized by small filter windows and residual connections.

The feature maps generated by the backbone are hierarchical, with $P3$, $P4$, and $P5$ consisting of grid cell sizes of 52×52, 26×26, and 13×13, respectively.

YOLOX uses SPP module. The SPP module is a method that allows a neural network to accept inputs of different sizes and then spatially pool the features in a

fixed-size feature map. It enables the network to handle objects of various sizes within the input image. The neck connecting the backbone to the head uses PAN module, which adds a bottom-up path to the FPN that combines the bottom-up and top-down paths to propagate high-resolution features to the upper layers. The head of YOLOX has a similar structure as FCOS, consisting of a branch for class prediction and a branch for bounding box regression prediction.

We apply our proposed SGFEM to this head section. SGFEM is applied to the class branches connected to the feature maps of $P4$ and $P5$, as shown in Fig. 3.7, in an attempt to capture features at grid boundaries.

Data Augmentation Improvement

YOLOX employs not only RandomFlip and ColorJitter data augmentation methods but also applies powerful data augmentation methods such as MixUp and Mosaic.

Mixup is a data augmentation technique that creates new training samples by combining pairs of images and their corresponding labels from the original training data set. The process involves taking a weighted sum of the two input samples and their labels to produce a new augmented sample and corresponding "mixed" label. Mosaic is a data augmentation technique combining four random training images into one image while merging and adjusting their bounding box annotations. This method yields images with varying object placement at each epoch, which is expected to result in a more shift-robust model.

However, as mentioned above, mosaic has not been able to extract grid boundary features well, so we expect our proposed GADA to be applied to more shift-robust feature representation.

To apply GADA, it is necessary to know the grid size of the feature map assigned to the object. However, YOLOX uses SimOTA to assign labels dynamically, so it is impossible to define by rule which of the grid sizes $\{8, 16, 32\}$ is selected, as is the case with FCOS. Therefore, we adopt a fixed grid size of 32, which is the greatest common divisor of the grid size.

Experimental Settings

The hyperparameter settings are almost the same as the original settings of YOLOX.

we conduct training 300 epochs, with the initial 5 epochs dedicated to a warm-up phase on the COCO train2017 dataset. We utilize Stochastic Gradient Descent (SGD) as our optimization algorithm for training. The learning rate is 0.01, and the training adopts a cosine learning rate schedule. To regularize the training process, we set the weight decay to 0.0005, while configuring the SGD momentum to 0.9. We employ Binary Cross-Entropy (BCE) Loss to train class and object branches, while the regression branch is trained using IoU Loss, ensuring a comprehensive optimization process.

TABLE 3.7: Comparison of proposed Methods on the COCO val set with YOLOX.FPS measured in runtime on a GeForce RTX3090.

Network	Proposed Methods	params(M)	FPS	GFLOPs	AP	AP_S	AP_M	AP_L
YOLOX-S	-	8.97	81.4	33.5	39.9	22.8	43.8	52.8
YOLOX-S	GADA	8.97	81.4	33.5	**40.4**	**24.0**	44.4	**54.3**
YOLOX-S	SGFEM	11.61	56.6	49.2	39.9	22.8	43.9	52.8
YOLOX-S	SGFEM + GADA	11.61	56.6	49.2	40.0	22.7	**44.5**	53.2

Furthermore, we employ various data augmentation techniques. These include Random Horizontal Flip, Color Jitter, Mixup, and Mosaic. Image size is fixed at 640×640.

The hyper-parameters used in the proposed method are as follows: GADA application ratio $\alpha = 0.5$ and original and auxiliary loss balance adjustment $\lambda = 1.0$.

Experimental results in YOLOX

In this section, we apply and evaluate the proposed methods to YOLOX. Table 3.7 shows the comparison results.

Applying GADA shows an improvement of $0.5AP$ in a typical metric. GADA improves the original YOLOX learning strategy in all metrics, regardless of object size.

When we apply SGFEM, the accuracy is almost identical to the original YOLOX. We believe this is because YOLOX successfully extracted the grid boundary information with the SPP and PAN modules employed in the backbone. The combination of SGFEM and GADA yields an improvement of $0.1AP_M$.

The evaluation in YOLOX shows that the best accuracy is obtained when only GADA is applied to the original model. These results suggest that YOLOX has successfully extracted features at the grid boundaries due to the ingenuity of the backbone. We believe that the original data augmentation suffers from feature extraction at the grid boundaries, but the proposed GADA successfully addresses this problem.

3.3.9 Experimental Application to the Faster RCNN method

In this experiment, we aimed to investigate the phenomenon of score drop at grid boundaries, specifically examining whether this issue is unique to one-stage methods that divide images into a grid or if it also occurs in two-stage methods such as Faster RCNN. The selected dataset for this investigation was the COCO validation dataset.

The results of the experiment are illustrated in Fig. 3.8. Notably, the figure demonstrates that one-stage methods, such as FCOS and YOLO, exhibit the phenomenon of score drop at grid boundaries. Despite this occurrence, the two-stage method, Faster RCNN, displays consistent class scores and robustness against shifts in the grid.

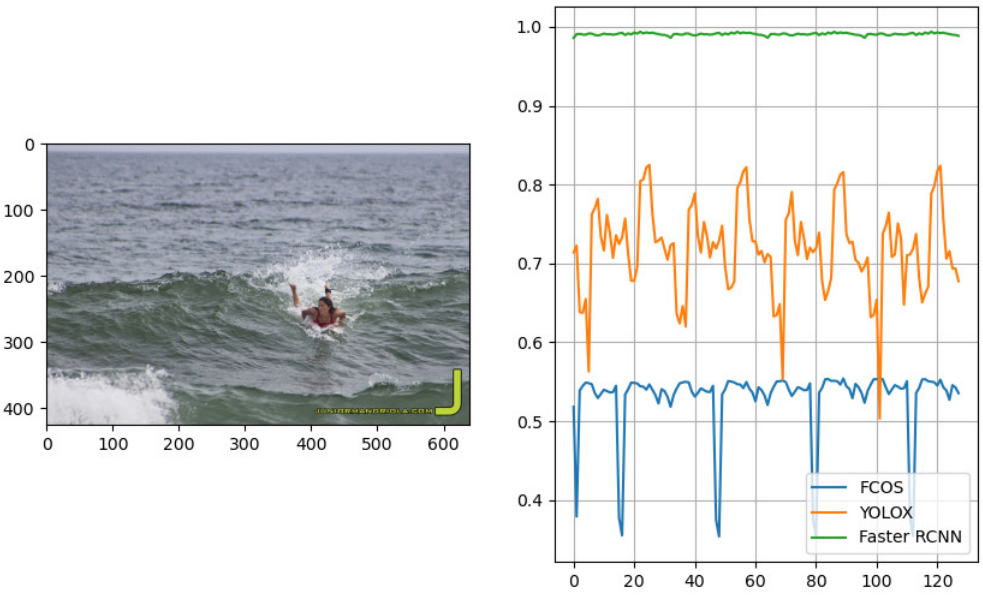


FIGURE 3.8: Observed score variation for shifts in the One-stage methods FCOS and YOLOX and the Two-stage method Faster RCNN.

These results suggest that the problem of reduced scores at the boundaries of the grid is often found in one-stage methods, highlighting a potential design limitation. In contrast, the behavior of the Faster RCNN towards this phenomenon shows the superiority of the two-stage method, which does not split the image into grids, and the promising property of the two-stage method of maintaining stable class scores.

Chapter 4

An Object Detection Method Using Probability Maps for Instance Segmentation to Mask Background

4.1 Problem of Imbalance Between Foreground and Background

In contemporary object detection, two predominant paradigms, one-stage and two-stage methods, offer distinct trade-offs in terms of speed and accuracy. The one-stage approach is celebrated for its high-speed processing capability, enabling real-time object detection. However, it generally exhibits reduced accuracy when compared to two-stage methods. The primary characteristic of one-stage methods involves the division of the entire image into a grid structure, whereby bounding boxes and corresponding object classes are predicted for each grid cell. This real-time processing capability, particularly showcased by popular solutions like YOLO [46] and SSD [41], makes one-stage detectors an appealing choice for applications prioritizing rapid object detection.

Nonetheless, the accuracy of one-stage methods tends to lag behind that of two-stage counterparts. A key factor contributing to this performance gap is the issue of class imbalance between foreground and background categories. With the grid-based partitioning of the image, a considerably larger proportion of grid cells are allocated to the background class. This class imbalance has the effect of tilting the learning process, giving the background class more influence during training, leading to an asymmetry that hinders the classification accuracy of the foreground objects.

In contrast, two-stage methods employ a distinctive strategy. Initially, they extract candidate foreground regions during the first stage, substantially alleviating concerns related to class imbalances between foreground and background. The second stage, which typically focuses on refining the region proposals, benefits from a better-balanced set of foreground and background classes. Consequently, two-stage

detectors experience less difficulty in managing class imbalance issues, which can be a key driver of their superior accuracy.

Notably, recent advancements in one-stage methods have demonstrated remarkable progress in addressing the class imbalance challenge. Prominent models like RetinaNet [38] and FCOS [57] have successfully implemented strategies to mitigate the impact of class imbalances. These innovations have effectively leveled the playing field between one-stage and two-stage methods, enabling one-stage detectors to achieve accuracy on par with their two-stage counterparts while preserving their renowned real-time processing capabilities. This breakthrough showcases the ongoing evolution of object detection techniques and their pursuit of optimal trade-offs between speed and precision.

To address the issue of foreground-background class imbalance in object detection, two notable advancements in the field, RetinaNet and FCOS, have introduced innovative strategies. These techniques have substantially contributed to the mitigation of class imbalance problems and improved the overall performance of object detectors.

RetinaNet introduced a pioneering solution in the form of the Focal Loss. The core concept behind Focal Loss is to suppress background error while prioritizing the accurate learning of foreground error. This is achieved through a dynamic weighting mechanism embedded within the cross-entropy loss function. Focal Loss takes into account that the accumulation of minor errors from easy samples classified as background may hinder the effective learning of foreground objects. To counter this, it assigns dynamic weights to the loss based on the magnitude of errors.

The dynamic weight allocation performed by Focal Loss is a pivotal component of its effectiveness. Specifically, the loss function assigns higher weights to challenging samples with substantial prediction errors, ensuring that gradients from these challenging instances are adequately reflected during the training process. Simultaneously, Focal Loss diminishes the impact of minor errors associated with easy samples that exhibit low prediction errors. By doing so, Focal Loss significantly alleviates the foreground-background class imbalance issue, leading to more robust and accurate object detection.

In contrast, FCOS tackled the class imbalance challenge through a distinctive approach the development of an anchor-free object detection method. Traditional object detectors, particularly one-stage approaches, often rely on anchor-based mechanisms to generate region proposals for objects. These anchors contribute to the disparity between foreground and background grids, exacerbating the imbalance issue. FCOS introduced an anchor-free alternative that reduces the foreground-background grid ratio.

The anchor-free design of FCOS eliminates the reliance on predefined anchor boxes, thereby eliminating the bias that such anchors may introduce. Instead, FCOS operates by directly predicting the location and size of objects within an image, without the need for anchor-based proposals. This fundamental shift in the detection

paradigm yields a more balanced distribution of grids for foreground and background objects, thus mitigating the class imbalance problem.

4.2 Can We Further Bridge the Imbalance Gap Between Foreground and Background?

In the domain of object detection, the management of foreground-background imbalance holds a paramount role in enhancing detection accuracy. It is widely acknowledged that addressing this imbalance can lead to significant improvements in the precision of object detection models. However, this imbalance is far from being a solved problem, and further explorations into effective strategies are essential.

4.2.1 Uncertainties and Limitations of Banding Box Information

Recognizing the persistent challenge of foreground-background imbalance, we embarked on a quest to explore alternative avenues for reducing the influence of background information. While previous state-of-the-art models such as RetinaNet and FCOS have demonstrated the effectiveness of certain approaches, we hypothesized that additional methods aimed at diminishing the impact of background elements could provide further enhancements in object detection accuracy.

In contemporary object detection methodologies, the reliance on bounding box information is a prevailing trend. This information aids in the localization of objects, making it instrumental for object detection tasks. However, it also introduces a particular challenge. The spatial confines of bounding boxes often encompass substantial background regions, a characteristic that complicates the learning process within the model. Background elements within bounding boxes may inadvertently interfere with the features extracted for object identification, leading to suboptimal performance.

4.2.2 The Evolution of Instance Segmentation Techniques

The landscape of computer vision has witnessed substantial progress in the domain of image segmentation. With the emergence of cutting-edge techniques, including SOLO [62, 63] and CondInst [56], the line between object detection and instance segmentation has begun to blur. These methods embrace anchorless grid-like prediction strategies similar to those proposed by FCOS.

SOLO, for instance, takes a holistic approach to segmentation, segmenting the entire image into grids and predicting instance masks based on the grid cell containing an object's center. CondInst, on the other hand, opts for grid-wise prediction, albeit with a unique twist. Instead of directly predicting instance masks, it focuses on the prediction of Fully Convolutional Network (FCN) parameters used in the mask branch. This innovative approach allows CondInst to achieve instance-aware masks, reflecting the evolution of techniques in this domain.

The art of meticulously painting a picture typically involves numerous iterations rather than being completed in a single attempt. It is when we strive for precision and perfection that we employ multiple careful strokes. Drawing an analogy from this practice, we conjecture that a recursive structure might hold the key to enhancing the effectiveness of instance segmentation.

In the realm of instance segmentation, where the task is to predict segmentation masks, a deviation from traditional bounding box predictions, our research has led us to an innovative approach. Inspired by the observation that achieving a seamless and detailed outcome often necessitates repeated, meticulous efforts, we propose the utilization of recursive structures. These structures offer a novel perspective on addressing the challenges of instance segmentation, where the inherent intricacies of predicting masks require a unique approach.

Remarkably, in the context of instance segmentation tasks, our exploration has unveiled an unconventional method that introduces self-predicted outcomes as prior knowledge in a feedback mechanism. To the best of our knowledge, this approach, which leverages self-predicted results as prior knowledge in a feedback structure, is unprecedented in the field of instance segmentation.

4.2.3 Overview of Our Proposed Method

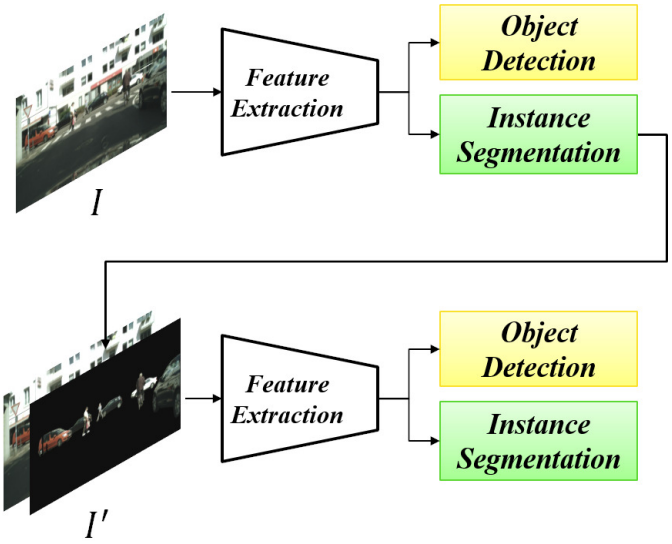


FIGURE 4.1: SODet Architecture. SODet is a two-step structure, the network having two branches, one for instance segmentation and the other for object detection. The first step, with the original image I as input, generates a mask image, and the second step predicts each task with the image I' transformed by the generated mask image.

Our novel approach, named Segmented Object Detection (SODet), represents a paradigm shift in the domain of object detection. It introduces a twin-stage traversal of the same neural network, as illustrated in Figure 4.1. This dual-pass architecture is underpinned by a transformative concept.

In the initial stage, SODet conducts instance segmentation to generate a probability map. This probability map plays a pivotal role as it assumes the guise of a background mask, effectively concealing the background information. It is during the second stage of the detection process that this background mask is reintroduced. This ingenious strategy allows the second stage to operate within an environment that has been meticulously purified of background interference.

4.3 Proposed Method

4.3.1 Mask Functions

The mask function $\mathcal{K}$ is used to fuse the mask image of the probability map with the input RGB image.

We select the most efficient mask function $\mathcal{K}$ from the following three equations.

a). Masked Image

$$\mathcal{K}_{(a)}(I, M) = I \otimes M \qquad \{I' \in \mathbb{R}^{W \times H \times 3}\} \tag{4.1}$$

b). Concatenation of the input Image and Mask

$$\mathcal{K}_{(b)}(I, M) = I \odot M \qquad \{I' \in \mathbb{R}^{W \times H \times 4}\} \tag{4.2}$$

c). Concatenation of the input Image and the Masked Image

$$\mathcal{K}_{(c)}(I, M) = I \odot (I \otimes M) \qquad \{I' \in \mathbb{R}^{W \times H \times 6}\} \tag{4.3}$$

Here $\otimes$ denotes element-wise multiplication, and $\odot$ denotes concatenate operation. In the case of $\mathcal{K}_{(b)}$, where the number of input channels has been expanded from 3 to 4, the initial value of kaiming [27] is applied to the added channel.In the case of $\mathcal{K}_{(c)}$, where the number of input channels has been expanded from 3 to 6, the pre-training weights are copied to the additional 3 channels.

4.3.2 SODet Overview

The proposed SODet consists of two steps of the processing as shown in Fig. 4.1.

In the first step, both object detection and instance segmentation are trained as a multi-task problem on a three-channel RGB color image I. Then we will use the trained network to estimate the pixel-wise probability map M as instance segmentation for the image I by feeding it to the network as input.

In the second step, the parameters of the network are re-trained by using the transformed image I' by transforming the mask function $\mathcal{K}$ from the original input image I with the background region. The mask function is given as

$$I' = \mathcal{K}(I, M). \tag{4.4}$$

By introducing this transformation in the second step, it is expected that the trained CNN can reduce the effect of the unnecessary background regions in the input image.

4.3.3 Network Architecture

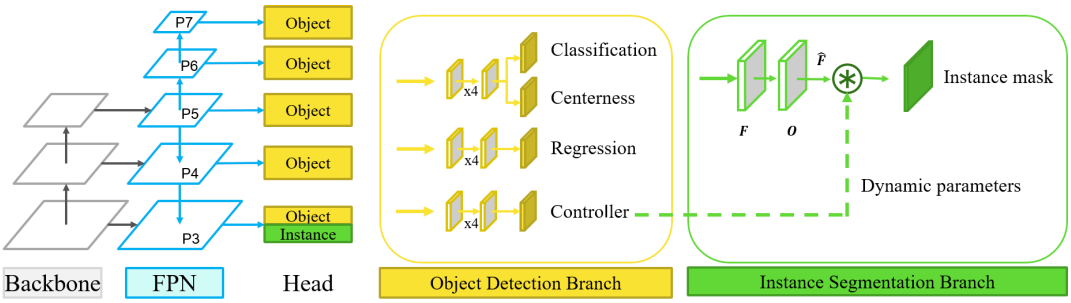


FIGURE 4.2: Network of SODet. FPN [37] generates the five-layer feature maps from the three-layer feature maps generated by Backbone and predicts the final result with a head from the feature maps.⊛ denotes the dynamic convolution operation.

The proposed method employs the CondInst architecture to train and evaluate both bounding box predictions by the object detection task and mask predictions by the instance segmentation task. The CondInst network consists of a backbone network and a feature pyramid network (FPN) to extract features, and branches for object detection and instance segmentation. In the instance segmentation branch, each mask is predicted using a dynamic convolution layer, and the parameters are changed based on the instance to generate a high-resolution mask. The schematic figure of the proposed network is shown in Fig.4.2.

Feature Extractor

The feature extractor generates feature maps with different resolutions using a fully convolutional fashion that combines the backbone bottom-up path and the FPN top-down path for input images in order to detect objects at various scales. As shown in Fig.4.2, the three-layer feature map produced by the backbone CNN and the upsampling feature maps is added to each element to generate a three-layer feature map ($P3$, $P4$, $P5$) with high-resolution. In addition, two more layers of feature maps ($P6$, $P7$) are added from the deeper layers by convolutional operations for a total of five layers of feature maps. Each feature map consists of 256 channels.

Object Detection Branch

Following the FCOS, the object detection branch adds four convolutional layers after the feature extractor. The feature map to predict an 8-dimensional vector $\boldsymbol{p}$ of classification labels, a 4-dimensional vector $\boldsymbol{t} = (l, t, r, b)$ of bounding box coordinates, and a 1-dimensional vector $\boldsymbol{q}$ for centerness that represents the location of object

center. The class label is given to any grid in the ground-truth box and is considered a positive sample. Otherwise, it will be a negative sample and will be the background class. Also, the vector $\boldsymbol{t}$ in the bounding box regression is the distance from the grid center to the four sides of the bounding box.

Instance Segmentation Branch

The segmentation branch uses NMS to remove duplicate detections for bounding boxes predicted by the object detection branch and predicts a mask of instances centered on the top 100 bounding box positions. Each bounding box is also associated with a filter parameter for the mask head generated by the controller head, which produces a dynamic mask head for the detected instance. The mask branch, which acts in parallel with the detection branch, receives feature map P3 generated by the FPN [37] and generates a feature map $\mathcal{F}$ with a resolution of $1/8$ size of the input image with eight channels. Feature map $\hat{\mathcal{F}}$ is generated by combining $\mathcal{F}$ with a coordinate map $\mathcal{O}_{x,y}$ relative to center position (x, y) of the instance. $\hat{\mathcal{F}}$ is input to an instance-aware dynamic mask head, and the number of channels is reduced from 8 to 1 using the FCN, which consists of three 1×1 Conv layers while preserving the resolution. However, in the study by Tian et al. [56], the final performance has been better at $1/4$ resolution than at the upsampled resolution of the input image; therefore, in our experiments, we also use a resolution mask of $1/4$ size of the input image as the final output.

4.3.4 Loss Function

The training loss function is defined as follows:

$$L = L_{bbox} + L_{segm} \tag{4.5}$$

$$L_{bbox} = L_{reg} + L_{cls} + L_{cent} \tag{4.6}$$

$$L_{segm} = L_{mask} \tag{4.7}$$

where L_{cls} is the Focal Loss [38] for boundingbox classification and L_{reg} is the IOU loss [48] for bounding box regression, and L_{cent} is binary cross entropy loss for centerness and L_{mask} is the Dice Loss [53] for mask prediction.

4.3.5 Probability Map

The mask used for the proposed SODet feedback is based on the method of Condinst, which uses NMS to generate a single probability map from the probability maps for each class generated in the instance segmentation branch.

The probability map is used to mask the background, so we do not give any category information, but use the foreground confidence S. If K instance masks

FIGURE 4.3: Sample images of masked cityscape verification data. Left is the original image I and right is the image masked by the probability map.

TABLE 4.1: Comparison results between the binary mask and soft mask.

Type of Mask	AP^{box}	AP^{mask}
binary mask	35.1	31.1
soft mask	35.2	31.3

overlap at pixel i, the probability map defines as:

$$M_i = \arg\max_{k \in K} S_i^k, \tag{4.8}$$

and employs the value with the highest foreground confidence S_i in the overlapping masks.

Two types of masks are possible: the soft mask, which uses the probability map as a sequence of values from 0 to 1, and the binary mask, which binarizes the probability map with a certain threshold value. We found that the performance of the soft mask was better than that of the binary mask for both object detection (AP^{box}) and instance segmentation (AP^{mask}). Table 4.1 shows the results of comparing the performance of applying soft and binary masks to the mask M applied in Equation 4.8. Thus, we decided to use the mask with a probability value.

As an example, Fig.4.3 shows an image masked by using the estimated probability map. By feeding back the probability map as a mask to the input image, we can reduce the influence of the background by setting the background region close to zero. It is expected that the detection accuracy of the trained model using masked images can be improved.

4.4 Experiments

4.4.1 Dataset

We present our experiments and results on the Cityscapes[9] benchmark for two tasks: object detection and instance segmentation. We use the COCO API to measure the AP (average precision) for IOUs in the range 0.5:0.05:0.95. As an evaluation metric for object detection, the AP of the bounding box is AP^{box}, and as an evaluation metric for instance segmentation, the AP of the mask is AP^{mask}.

Cityscapes is a dataset of real urban scenes, containing 3,475 images captured by an in-vehicle camera; 2,975 images are used for training and the remaining 500 images are used for validation. Since there are no annotations in the test set, we report the results of the validation set. We prepare the tightest bounding box of the instance segmentation mask as the ground truth using the conversion tool provided by MMDetection[7]. The dataset contains eight object categories: persons, riders, cars, trucks, buses, trains, motorcycles, and bicycles.

4.4.2 Implementation Details

The backbone network is ResNet-50, and 256-channel feature maps are generated in FPN. From this feature map, the object detection branch predicts the centerness (foreground or background), bounding box, and classifications for each grid, and the instance segmentation branch predicts the instance mask.

We trained the proposed SODet by using Stochastic Gradient Descent (SGD) for 64 epochs and an initial learning rate of 0.01 which is decreased by a factor of 10 at 56 epochs. The parameters of the weight decay and the momentum are set to 0.0001 and 0.9, respectively.

The image is randomly clipped from the original image width of 2048 to 1024 regions, and random flips are applied as data augmentation at a rate of 50% for training.

The network weights are not shared between the first and second steps, and both steps are starting to train from initialized parameters by pre-training with ImageNet [12].

4.4.3 Comparison of Mask Functions

In order to find the better use for the probability maps estimated in the second step of the proposed SODet, we experimentally investigate the three mask functions($\mathcal{K}_{(a)}$, $\mathcal{K}_{(b)}$,$\mathcal{K}_{(c)}$) described above.

Table 4.2 shows the average precisions obtained by applying mask functions $\mathcal{K}_{(a)}$, $\mathcal{K}_{(b)}$, $\mathcal{K}_{(c)}$. The top row in the table also includes the result obtained for the case of using the original image as input without using the mask function.

It is noticed that the average precisions obtained by the mask function $\mathcal{K}_{(b)}$ are higher than the precisions by the original image and the other map functions. On the

TABLE 4.2: Comparison of the average precisions for mask functions.

Method	AP^{box}	AP^{mask}
Original image	34.4	31.1
$\mathcal{K}_{(a)}$	33.2	29.9
$\mathcal{K}_{(b)}$	**35.2**	**31.3**
$\mathcal{K}_{(c)}$	33.3	29.5

TABLE 4.3: Comparison of Average Precision(AP) between our proposed method and various state-of-the-art methods.

Method	AP for object detection				AP for instance segmentation			
	AP^{box}	AP^{box}_S	AP^{box}_M	AP^{box}_L	AP^{mask}	AP^{mask}_S	AP^{mask}_M	AP^{mask}_L
FCOS	34.3	15.0	33.4	53.9	-	-	-	-
CondInst	34.4	**16.3**	**34.8**	52.7	31.1	8.4	**29.3**	55.2
SODet	**35.2**	16.2	34.1	**54.9**	**31.3**	**8.9**	27.7	**56.9**
Mask R-CNN	33.7	**16.8**	35.0	49.5	**30.9**	**9.2**	**29.1**	50.7
SODet (Mask RCNN base)	**34.3**	16.6	**35.1**	**51.2**	30.5	8.8	26.9	**51.2**

other hand, the other mask functions $\mathcal{K}_{(a)}$ and $\mathcal{K}_{(c)}$ did not give a better performance than the baseline.

For these reasons, we assume that $\mathcal{K}_{(a)}$ performs poorly because only the masked image is input, erasing even the edges of the object. The input image and mask are provided in separate channels for $\mathcal{K}_{(b)}$, so we assume that the edge and mask information is effectively obtained. We assume that $mathcalK_{(c)}$ performs poorly because the input consists of 6 channels of original and masked images, and the information is redundant and difficult to capture features.

From this result, we decided to use $\mathcal{K}_{(b)}$ for the following experiments.

4.4.4 Comparison of proposed method and state-of-the-art methods

To confirm the effectiveness of the proposed SODet, Table 4.3 shows the results of a comparative evaluation with the state-of-the-art object detection method FCOS and the state-of-the-art instance segmentation methods Mask R-CNN and CondInst. Since FCOS is the object detection method and is not an instance segmentation method, we can not evaluate the instance segmentation accuracy.

First, we compare the FCOS and CondInst results. FCOS and CondInst have the same architecture and loss function as object detection, but CondInst has a parallel instance segmentation branch, where a per-pixel instance mask is given as supervised. Comparing the AP^{box} of object detection, FCOS has 34.3 while CondInst has 34.4, almost the same results, but for small objects, the AP^{box}_s detection accuracy for CondInst has improved significantly from 15.0 to 16.3. We thought this is because

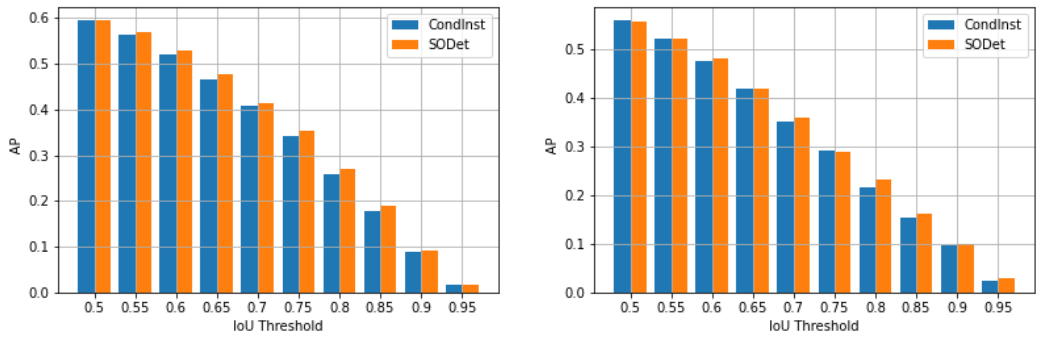
CondInst shares feature maps in both the object detection and instance segmentation branches only for the high-resolution feature map ($P3$), and thus the instance segmentation learning had a beneficial impact on the object detection learning.

Next, we compare CondInst and SODet results. Compared to CondInst, the proposed method improves the AP^{box} for object detection by 0.8 points and the AP^{mask} for instance segmentation by 0.2 points. We think this improvement is due to the effective learning of prior knowledge of instance segmentation, which is given in the second step using the probability map from the first step as a mask.

Further Comparing the results in more detail, for large objects, both the object detection accuracy AP_L^{box} and the instance segmentation accuracy AP_L^{mask} show significant improvement. The fact that the detection accuracy of large objects that should be detected in the lower resolution feature maps ($P4$, $P5$, $P6$, $P7$) improves can not be observed from the difference between FCOS and CondInst, which simply added an instance segmentation branch. This suggests that the proposed method can effectively propagate the information obtained from instance segmentation to the entire feature pyramid by providing instance segmentation information as input.

Similarly, compare the results with a mask feedback procedure like SODet in Mask R-CNN. Compared to Mask R-CNN, SODet (Mask R-CNN base) improves AP^{box} of object detection by 0.6 points and decreases AP^{mask} of instance segmentation by 0.4 points. For larger objects, the trend of improvement is the same as for CondInst. This suggests that improvement can be expected when Mask R-CNN is used as object detection (Faster R-CNN [22] configuration).

4.4.5 Qualitative Evaluation



(A) Object detection (B) Instance segmentation

FIGURE 4.4: Results of comparative evaluation of AP at varying IoU thresholds.

In order to investigate the factors that led to the improvement in the proposed method, we will discuss the differences between the CondInst and SODet results from two different perspectives.

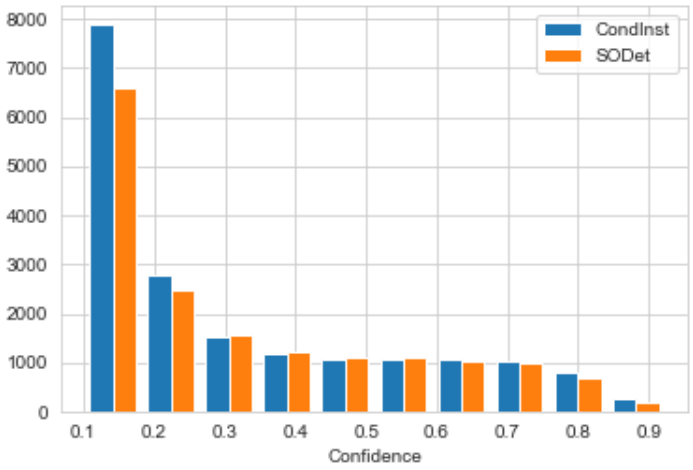


FIGURE 4.5: Confidence score histogram for classification.

The first is a comparison of the distribution of robustness with respect to the IoU threshold. The results are shown in Fig.4.4. The left graph in the figure shows the results of object detection and the right graph shows the results of instance segmentation, showing the change in detection accuracy *AP* for different IoU thresholds. The proposed method is able to preserve accuracy in both object detection and instance segmentation tasks, even if the IoU threshold is increased. Therefore, we consider that the proposed method is able to extract features that more accurately capture the boundary with the object by adding a background mask.

The second is a comparison of the distribution of confidence scores for classifications in the object detection branch. A histogram of the prediction results showing a confidence score of 0.1 or higher for class classification in object detection is shown in Fig.4.5. We can see that the proposed method has fewer detections at lower confidence levels with respect to CondInst. We believe that this is due to the fact that the proposed method is able to make more robust predictions by using the probability map given as input to suppress the effects of unnecessary information given by the background.

4.4.6 Quantitative Evaluation

The results of object detection are shown in Fig.4.6. Comparing the results of CondInst in the left column of this figure with the results of the proposed method in the right column, we can see that the proposed method can detect more objects as shown by the red dashed areas in the figure. We thought this was because the proposed method could separate the densely overlapping detection targets and make a sharper estimation due to the effect of the mask. In addition, as shown in the third column, there were a few cases where detection failed because the mask erased small objects.

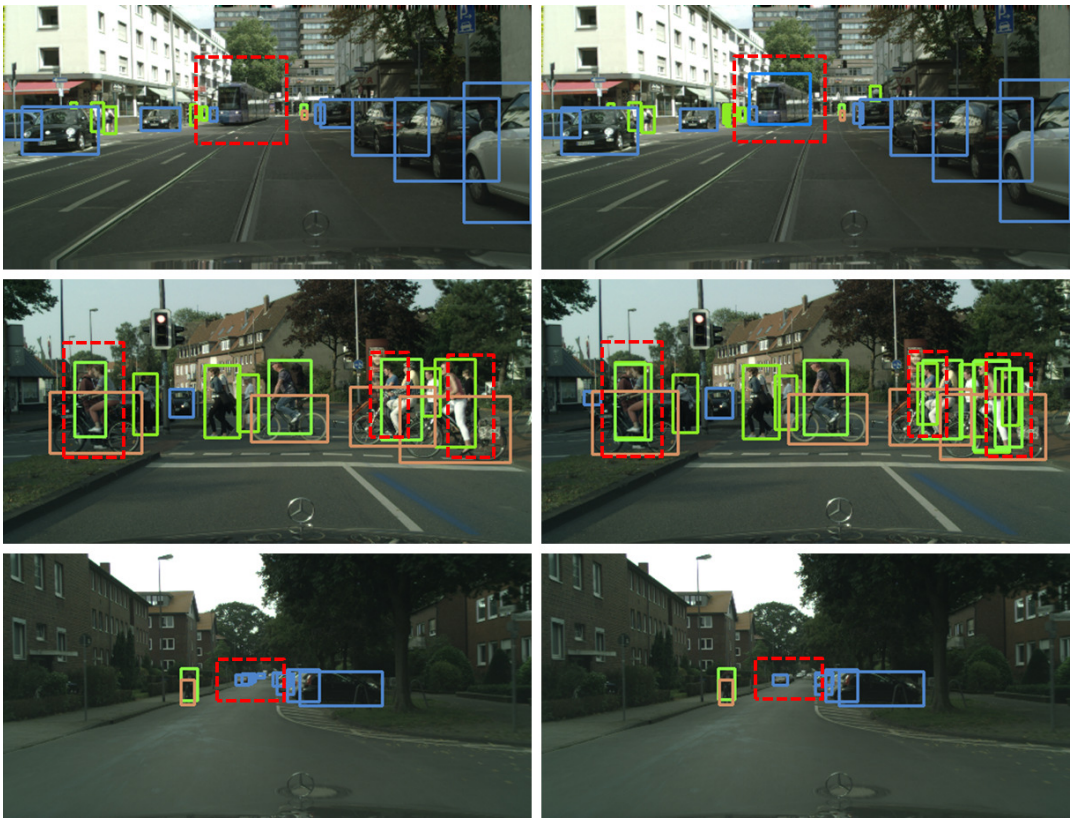


FIGURE 4.6: Displays the results of the object detection: the left column shows baseline results and the right column shows SODet results. The red dashed area is the change in results of interest.

Following the results of object detection, the results of instance segmentation are shown in Fig.4.7. CondInst results are shown in the top row and the results of the proposed method are shown in the bottom row. Also, columns 1 and 2 show the successful cases where the proposed method worked well, and column 3 shows the failure cases where the proposed method lost accuracy. Comparing the results of the instance segmentation, the proposed method can recognize the boundary more clearly, as shown by the red dashed area in the figure, and it can capture the shape of the instance. However, there are some cases of false positive detection. The failure case shown in the figure misclassified washed clothing as a person. We think that these false positives are due to local features caused by masking.

4.4.7 Comparison on the COCO dataset

To prove the generalization ability of the SODet, we evaluate our proposed method with the COCO [39] data set.

The network is trained by SGD for optimization, with an initial learning rate of 0.01, 16 mini-batches, and 12 total epochs. The learning rate is reduced by 10 at the 8th and 11th epochs. The weight decay and the momentum are set as 0.0001 and 0.9, respectively. Input images are resized to a maximum scale of 1333×800, without

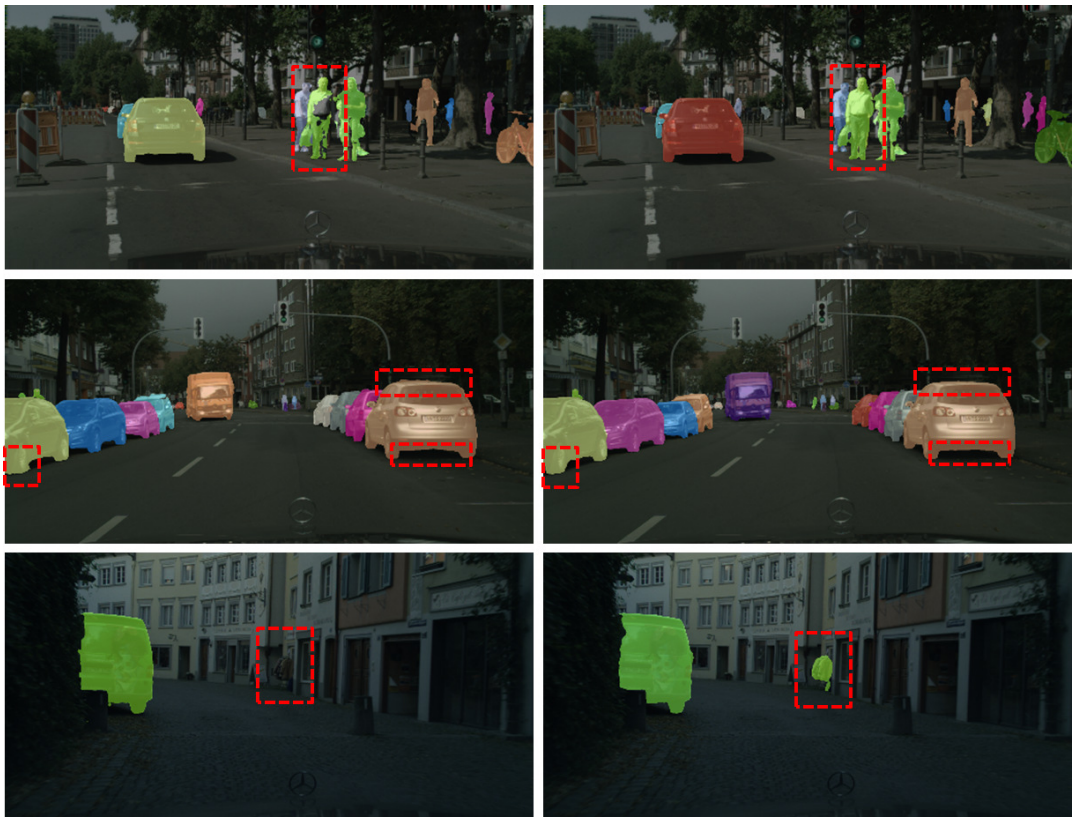


FIGURE 4.7: Displays the results of the instance segmentation: the left column shows baseline results and the right column shows SODet results. The red dashed area is the change in results of interest.

changing the aspect ratio. Only random horizontal image flipping is used for data augmentation.

As shown in Table 4.4, SODet shows improvement over state-of-the-art methods on the COCO data set. The improvements follow the same trend as CityScapes. Compared to CondInst, the proposed method improves the AP^{box} for object detection by 0.6 points and comparable performance on AP^{mask} for instance segmentation.

We also experiment with applying the SODet-like mask feedback procedure to Mask R-CNN. Mask R-CNN obtains an improvement of $0.4AP^{box}$ as object detection. For instance segmentation, it degrades by $0.1AP^{mask}$.

TABLE 4.4: Comparison of Average Precision(AP) between proposed method and various state-of-the-art methods on COCO val set.

Method	AP^{box}	AP^{mask}
FCOS	40.8	-
CondInst	41.8	**37.9**
SODet	**42.4**	**37.9**
Mask R-CNN	39.3	**35.9**
SODet (Mask R-CNN base)	**39.7**	35.8

Chapter 5

Graph Laplacian Regularization based on the Differences of Neighboring Pixels for Conditional Convolutions for Instance Segmentation

5.1 Blurring of Instance Masks

5.1.1 Developments and Challenges in Instance Segmentation

Instance segmentation has emerged as one of the most intricate and demanding tasks in the realm of computer vision. This task involves not only accurately delineating object instances within an image but also assigning each pixel to a specific object category. Recent years have witnessed remarkable progress in the field of instance segmentation, primarily propelled by the advent of convolutional neural networks (CNNs). These advanced neural networks have contributed significantly to the enhanced performance and efficiency of instance segmentation methods [28, 8, 68, 64, 65, 10, 36, 3, 6, 35].

Despite the impressive advancements and the utilization of state-of-the-art methodologies [63, 56], certain challenges persist in the instance segmentation domain. Notably, issues concerning the smoothness of instance boundaries and the presence of indistinct regions within the instance masks have been identified (Fig.1.4). It is our belief that these challenges stem from the prevailing approach in conventional instance segmentation methods, which predominantly focus on predicting masks at a pixel-by-pixel granularity. This approach tends to overlook the inherent spatial relationships and structural dependencies among neighboring pixels. Consequently, there exists a critical need for innovative strategies that consider the broader context of neighboring pixels in order to address these persisting challenges effectively.

5.1.2 Spatial Regularization for Improved Instance Segmentation

Even with the results of high-resolution masks, there are problems such as blurred boundaries and hollows in the instances. We assume that these problems are due to the spatial structure and contextual information contained in the relationships between neighboring pixels not being incorporated well into the model.

Hakim et al. [24] demonstrated that a regularization defined by using the difference between the differences of the predict and target images can improve the performance of a CNN model in super-resolution and image segmentation tasks.

This method assumes that Binary Cross Entropy (BCE) and Sum Square Error (SSE), provided as pixel-wise losses, do not preserve the relationships between neighboring pixels. For pairs of neighboring pixels, they defined a regularization that preserves the relationships between neighboring pixels by introducing a constraint. Consequently, the differences between pairs belonging to the same class are small, whereas between those belonging to different classes are large. Hakim et al. proposed a Graph Laplacian Regularizer method based on Differences of Neighboring Pixels (GLRDN). This regularizer can be defined as a graph Laplacian by representing the relationships between neighboring pixels as a graph. These authors successfully generated images with clear boundaries in super-resolution and image segmentation tasks by applying this regularization.

We propose a method for applying this regularisation to instance segmentation that penalizes errors in the spatial structure using a graph consisting of the differences between neighboring pixels.

5.1.3 Graph-based regularization

Many studies have viewed regularization in learning methods as a graph optimization problem. For simplicity, we describe the task of predicting a binary mask of the foreground and the background. Let node $\mathcal{V} = \{i | i = 1, ..., N\}$ represent the set of mask probability values for a pixel with N pixels, where the probability is 1 if the pixel belongs to the foreground and 0 if it belongs to the background. Let an edge $\mathcal{E} = \{(u, v) | u, v \in \mathcal{V}\}$ represent a set of adjacencies between pixels, where the adjacency is set as 1 if the edge belongs to the four nearest neighbors of a node, and 0 otherwise. Consider the following problem on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$\text{minimize} \sum_{i \in \mathcal{V}} \mathcal{L}(y_i, t_i) + \sum_{(u,v) \in \mathcal{E}} \mathcal{P}(y_u, y_v). \tag{5.1}$$

where y_i and t_i denote the prediction mask and the target mask, respectively, $\mathcal{L}$ denotes the loss term, and $\mathcal{P}$ denotes the regularization term.

We focus on the regularization term, for which the traditional method is Laplacian regularization. In Laplacian regularization assumes that the neighboring data are labeled the same, and the equation is defined as follows:

$$\text{minimize} \sum_{i \in \mathcal{V}} \mathcal{L}(y_i, t_i) + \lambda \sum_{(u,v) \in \mathcal{E}} w_{u,v} \|y_u - y_v\|_2^2. \tag{5.2}$$

The regularization term represented by an edge penalizes the differences between the variables of neighboring nodes, where $w_{u,v}$ is the weight of the importance of the edge between nodes u and v. This equation is known as a graph Laplacian. In the case of a graph consisting of four neighborhoods, modifying the regularization term of the above equation to the L1 norm is regarded Fused Lasso [58], and it is known to cluster and force neighboring pixels to have the same values if they belong to the same class.

However, these regularizations using a graph Laplacian begin to fail owing to a lack of scalability when the dataset becomes larger and more complex. For this problem, Hallac et al. showed that Network Lasso [25], which does not use all edge pairs in the graph, but only adjacent edges, is a helpful method for representing the above convex optimization problem. Let an edge $\mathcal{S} = \{\mathcal{S} \subset \mathcal{E}\}$ represent a set of adjacencies between pixels; then the above equation can be modified as follows.

$$\text{minimize} \sum_{i \in \mathcal{V}} \mathcal{L}(y_i, t_i) + \lambda \sum_{(u,v) \in \mathcal{S}} w_{u,v} \|y_u - y_v\|_2^2. \tag{5.3}$$

Therefore, we propose a regularization that incorporates important information such as the spatial structure, local context, and structural knowledge contained in pixel neighborhood relations into the learning of a model and enforces consistency for neighboring pixels belonging to the same label. We propose a new graph Laplacian regularizer for instance segmentation, in which neighboring pixels are forced to be closer if they belong to the same class and to be apart if they belong to different classes. We also conduct experiments to examine if the proposed regularization term can be modified like Fused Lasso to achieve the effect of clustering.

5.2 Proposed Method

In this section, we first introduce the proposed graph Laplacian regularizer for instance segmentation and subsequently present the network and the loss function used in our experiments.

Most instance segmentation methods, including CondInst [56], use a sigmoid function to convert the logit of the last layer that predicts the mask into a probability value for each pixel, which implicitly represents the spatial structure of the pixel neighborhood. Because neighboring pixels frequently belong to the same class in instance segmentation masks, we thought we could improve accuracy by explicitly incorporating important information obtained from pixel neighborhood relations into the training of the network.

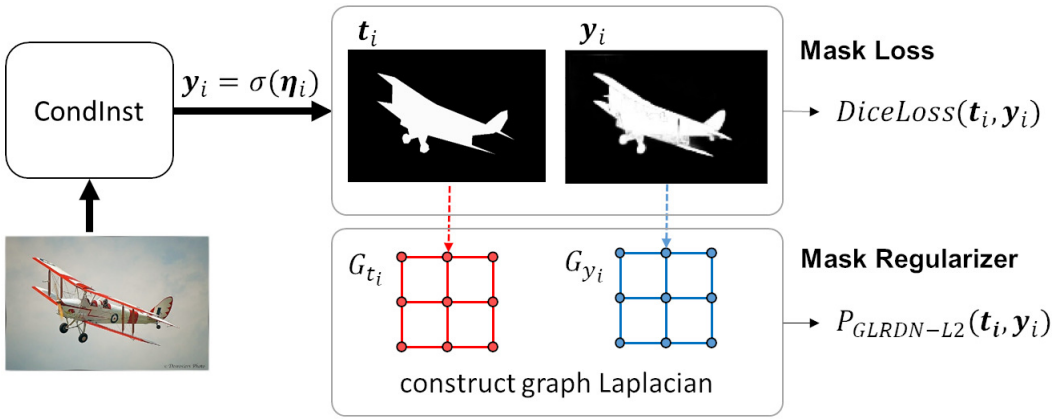


FIGURE 5.1: Visualization of our proposed approach, where η_i is output logit of mask head for instance i.

Moreover, the Graph Laplacian Regularization (GLR) used in segmentation is defined using the predicted mask $\boldsymbol{y}$ as in the following equation, which focuses on pixel-by-pixel regularization.

$$\mathcal{P}_{GLR}(\boldsymbol{y}) = \frac{1}{M} \sum_{(u,v)\in S} (y_u - y_v)^2, \tag{5.4}$$

where M is the number of pixels in the mask.

Our proposed regularization method aims to solve optimization problems that incorporate the spatial structure and context of the neighborhood by providing not only the pixel-level error but also the error of the differences between the surrounding pixels as a teacher in training the network.

The difference between our regularization and the general graph Laplacian regularization is that we apply the difference between $\Delta\boldsymbol{t}$ and $\Delta\boldsymbol{y}$ as the regularization, where $\Delta\boldsymbol{t}$ is the difference between adjacent pixels in target mask $\boldsymbol{t}$ and $\Delta\boldsymbol{y}$ is the difference between adjacent pixels in predicted mask $\boldsymbol{y}$. Thus, the relationships between neighboring pixels can be represented as constraints that follow the target mask, making the predicted mask more similar to the target mask. We call our regularization GLRDN-L2 (Graph Laplacian L2-Regularizer based on Differences of Neighboring Pixels) and define it as follows:

$$\begin{aligned}\mathcal{P}_{GLRDN-L2}(\boldsymbol{y}, \boldsymbol{t}) &= \frac{1}{M} \sum_{(u,v)\in S} ((y_u - y_v) - (t_u - t_v))^2 \\ &= \frac{1}{M} (\boldsymbol{y} - \boldsymbol{t})^T L (\boldsymbol{y} - \boldsymbol{t}),\end{aligned} \tag{5.5}$$

where L is the graph Laplacian matrix. This regularization term is expected to impose a grouping force on pixel pairs belonging to the same class to make predictions more consistent, whereas it applies a separation force on paired pixels belonging to different classes to make predictions more inconsistent.

5.2.1 Formula as a Graph Laplacian Matrix

In this subsection, we will express Eq.5.4 and Eq.5.5 in the form of a graph Laplacian. To begin, the graph for Eq.5.4 is constructed by considering the 4-neighbors of pixels as an undirected graph. We assign a value of 1 to edges connecting pixel nodes that are 4 neighbors, and a value of 0 to all other edges. The node values are represented as prediction values denoted as bmy. Under this setup, Eq.5.4, which represents the differences between predicted pixels, can be defined as follows using the Laplacian matrix L:

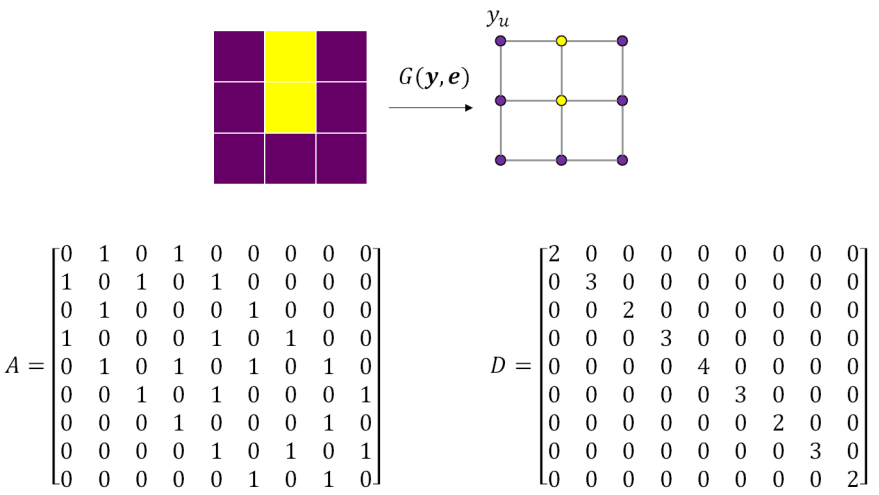


FIGURE 5.2: Illustration of the nodes and edges of a 4-neighborhood graph and the degree matrix (D) and adjacency matrix (A) in the graph Laplacian.

$$\sum_{(u,v)\in S} (y_u - y_v)^2 = \boldsymbol{y}^T L \boldsymbol{y}, \tag{5.6}$$

here, D represents the degree matrix of the graph, and A is the adjacency matrix. The Laplacian matrix L can be expressed as $L = D - A$.

In the case of Eq.5.5, we define the 4-neighbors as a directed graph. The nodes are assigned values of 1 for foreground and 0 for background. Let the target mask be denoted as $\boldsymbol{t}$ and the predicted mask as bmy. We define the difference between neighboring nodes as edges. In this setup, Eq.5.5, which calculates the difference of differences between nodes, can be defined as follows using the Laplacian matrix L:

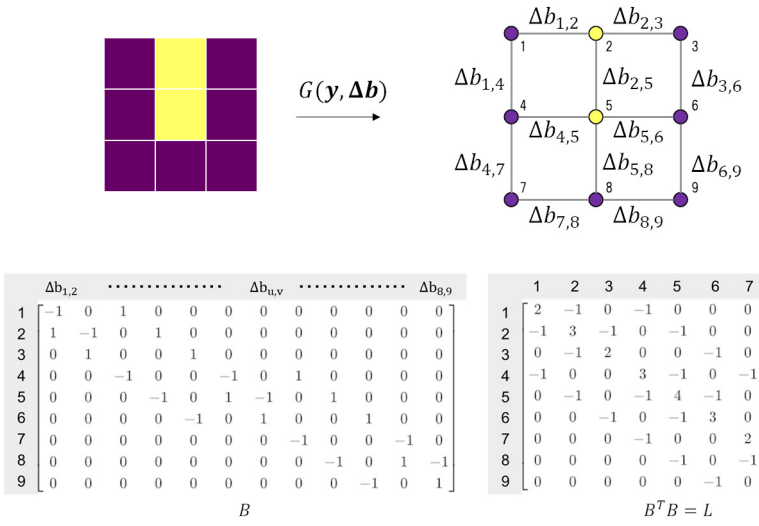


FIGURE 5.3: llustration of the nodes and edges of a 4-neighborhood graph and the incident matrix (B) in the graph Laplacian matrix for difference of neighboring pixels.

$$
\begin{aligned}
&\sum_{(u,v)\in S} \left((y_u - y_v) - (t_u - t_v)\right)^2 \\
&\qquad = \sum_{(u,v)\in S} \left(\Delta y_{u,v} - \Delta t_{u,v}\right)^2 \\
&\qquad = (\boldsymbol{\Delta y} - \boldsymbol{\Delta t})^T (\boldsymbol{\Delta y} - \boldsymbol{\Delta t}) \\
&\qquad = (B\boldsymbol{y} - B\boldsymbol{t})^T (B\boldsymbol{y} - B\boldsymbol{t}) \\
&\qquad = (\boldsymbol{y} - \boldsymbol{t})^T B^T B (\boldsymbol{y} - \boldsymbol{t}) \\
&\qquad = (\boldsymbol{y} - \boldsymbol{t})^T L (\boldsymbol{y} - \boldsymbol{t}) .
\end{aligned} \tag{5.7}
$$

5.2.2 Architecture

We use the CondInst [56] architecture for our experiments. CondInst is composed of a ResNet-based FPN [37] backbone, mask branch, and detection branch. CondInst adds a head called controller to the FCOS [57]-like detection branch. It dynamically predicts the weights of the mask head. By dynamically predicting the weight parameters of the mask head, it is possible to generate highly representational masks with instance-aware FCN.

When an input image is fed into the network, multiple feature maps (P1, P2, P3, P4, P5) with different resolutions are generated by the FPN, and the bounding box, category, and center-ness predictions for each instance are determined in the object detection branch. Subsequently, a bounding-box-based NMS removes duplicate detections and predicts the masks for the top 100 instances. Each bounding box is also associated with a filter parameter for the mask head generated by the controller head, which produces a dynamic mask head for the detected instance.

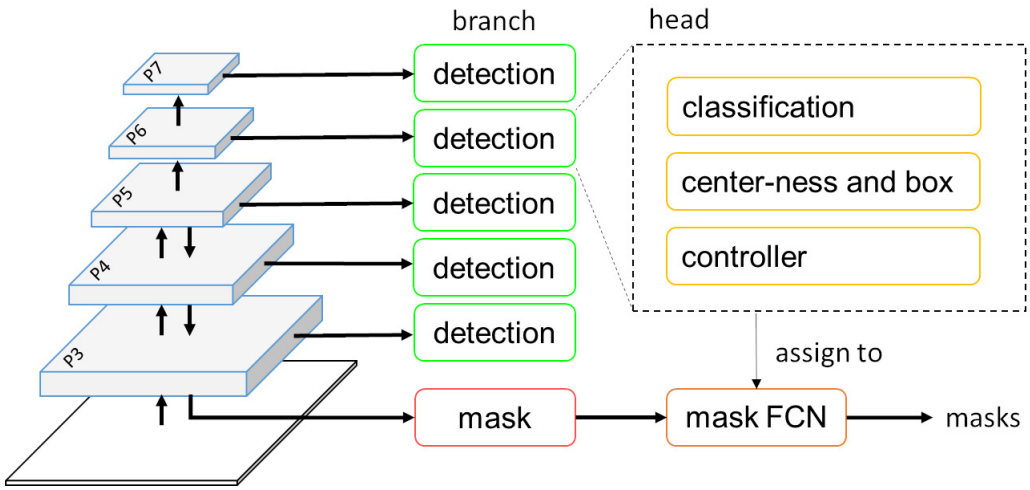


FIGURE 5.4: Overall of the network architecture used in our experiments.*P3 - P7* are feature maps generated by backbone network. There are mask and detection branches. Detection branch predicts category, bounding box, center-ness, and controller of target instance at position (x, y). Controller generates filter parameters of mask head for that instance. Mask head is instance-aware and applied to image same number of times as number of instances in image.

The mask branch, which acts in parallel with the detection branch, receives feature map P3 generated by the FPN [37] and generates a feature map $\mathcal{F}$ with a resolution of $1/8$ size of the input image with eight channels. Feature map $\hat{\mathcal{F}}$ is generated by combining $\mathcal{F}$ with a coordinate map $\mathcal{O}_{x,y}$ relative to center position (x, y) of the instance. $\hat{\mathcal{F}}$ is input to an instance-aware dynamic mask head, and the number of channels is reduced from 8 to 1 using the FCN, which consists of three 1×1 Conv layers, while preserving the resolution. However, in the study by Tian et al. [56], the final performance has been better at $1/4$ resolution than at the upsampled resolution of the input image; therefore, in our experiments, we also use a resolution mask of $1/4$ size of the input image as the final output.

5.2.3 Loss function

The overall loss function of CondInst is formulated as follows:

$$\mathcal{L}_{overall} = \mathcal{L}_{det} + \lambda_m \mathcal{L}_{mask} + \lambda_g \mathcal{L}_{reg}, \tag{5.8}$$

where $\mathcal{L}_{det}$, $\mathcal{L}_{mask}$, and $\mathcal{L}_{reg}$ represent the loss of the object detection, loss of the instance masks, and proposed graph Laplacian regularizer, respectively.

$\mathcal{L}_{det}$ is defined as follows:

$$\mathcal{L}_{det} = \lambda_k \mathcal{L}_{cate} + \lambda_r \mathcal{L}_{reg} + \lambda_c \mathcal{L}_{cent}, \tag{5.9}$$

where $\mathcal{L}_{cate}$ is Focal Loss [38] for the bounding box classification and $\mathcal{L}_{reg}$ is IoU Loss

[48] for the bounding box regression, and $\mathcal{L}_{cent}$ is Binary Cross Entropy Loss for the center-ness.

Denoting a sigmoid function as σ, the mask prediction result for location (x, y) is defined as follows:

$$\boldsymbol{y}_{x,y} = \sigma(MaskHead(\hat{F}_{x,y}; \boldsymbol{\theta}_{x,y})), \tag{5.10}$$

where $\boldsymbol{\theta}_{x,y}$ is the filter parameter for the mask head of the instance with central location at (x, y).

$\mathcal{L}_{mask}$ is defined as follows:

$$\mathcal{L}_{mask} = \frac{1}{N_{pos}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}>0\}} \mathcal{L}_{dice}(\boldsymbol{y}_{x,y}, \boldsymbol{t}_{x,y}), \tag{5.11}$$

where $c_{x,y}$ is the classification label of location (x, y). If $c_{x,y} = 0$, it is not associated with any instance. N_{pos} is the number of locations in the foreground region where $c_{x,y} > 0$. $\mathbb{1}_{c_{x,y}>0}$ is an indicator function, which is 1 if $c_{x,y} > 0$ and 0 otherwise. $\mathcal{L}_{dice}$ is Dice Loss [53], which is used to resolve the imbalance between the foreground and background pixels.

To calculate the loss between predicted mask $\boldsymbol{y}$ and target mask $\boldsymbol{t}$, they must be of the same size. As mentioned above, the best resolution for the final prediction on the COCO dataset is $1/4$ of the input image size; therefore, target mask $\boldsymbol{t}$ is also downsampled to $1/4$, making both sizes the same to calculate the loss.

We combined the proposed regularizer,

$$\mathcal{L}_{reg} = \frac{1}{N_{pos}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}>0\}} \mathcal{P}_{GLRDN-L2}(\boldsymbol{y}_{x,y}, \boldsymbol{t}_{x,y}), \tag{5.12}$$

with the loss adopted by CondInst. Because target mask $\boldsymbol{t}$ and predicted mask $\boldsymbol{y}$ in the regularization term have precisely the same forms as the input form for the mask loss term, the proposed regularizer is straightforward to implement.

5.3 Experiments

To evaluate the effectiveness of the proposed method, we have performed experiments on the COCO and Cityscapes instance segmentation datasets.

5.3.1 Implementation Details

ResNet-50 [26] is used as our backbone network, and weights pre-trained in ImageNet [12] are used for initialization. The weights of the newly added layers are initialized by the Kaiming initialization [27].

The network is trained by stochastic gradient descent (SGD) for optimization, with an initial learning rate of 0.01, 8 mini-batches, and 36 total epochs. The learning rate is reduced by a factor of 10 at the 28th and 34th epochs, respectively. The weight

decay and the momentum are set as 0.0001 and 0.9, respectively. The input image is resized to 640 or 800 on the short side as multi-scale data augmentation during training. The long side is set as 1333 or less. This scaling technique is followed in CondInst. The scale factor of the mask resolution is set as 4, which is the best value based on the experimental results of Tian et al. [56].

All hyperparameters for balancing the loss terms are λ_r, λ_k, λ_c, and λ_m are set as 1.0. For the main experiment, for λ_g, we adopt the parameter with the highest AP evaluated on the COCO val dataset in the model trained by COCO minitrain [49]. In addition, because learning with regularization from its initial stage is unstable, λ_g is set as 0.0 for up to two epochs to stabilize the learning. The comparison results for λ_g are summarized in TABLE.5.1.

We used a single GPU RTX3090 in experiments. A training time increased by about 2 minutes with the proposed regularization, compared to the baseline training time of about 3 hours and 6 minutes per epoch on the COCO dataset. It is a negligible small (about 1%) increase.

TABLE 5.1: Results of instance partitioning with varying number of λ_g in COCO val dataset. The model is trained with COCO-minitrain.

	AP		
λ_g	GLR	GLRDN-L1	GLRDN-L2
0.0	26.3	26.3	26.3
0.01	-	25.7	-
0.1	26.1	**26.5**	26.3
1.0	26.3	25.0	26.5
10.0	**26.5**	16.0	**27.0**
20.0	26.3	-	26.6

5.3.2 Results

Comparison on COCO Instance Segmentation

TABLE 5.2: Comparison of baseline and several regularization methods on COCO val dataset. “baseline” is CondInst without regularization, “GLR” is general graph Laplacian regularization, “GLRDN-L2” is proposed regularization, and ”GLRDN-L1” is a modification of proposed method to L1 norm.

regularization	*AP*	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
baseline	37.2	58.3	39.7	**18.7**	40.7	52.8
GLR	37.2	58.3	39.8	18.2	40.6	52.8
GLRDN-L1	37.1	58.2	39.6	18.4	40.5	52.6
GLRDN-L2	**37.7**	**58.7**	**40.1**	18.3	**41.1**	**54.0**

We compare the proposed method with different regularizations to show its effectiveness. TALBE.5.2 summarizes the comparison results, where the baseline is the original CondInst [56] without any regularizer, GLR is a simple Laplacian regularizer, GLRDN-L2 is our regularizer, and GLRDN-L1 is a variant of the proposed regularizer defined using L1 norm.

Comparing based on AP, we note that our GLRDN-L2 regularization outperforms the other methods, improving the performance by 0.005 points compared to the baseline.

The fact that the GLR does not improve its performance compared to the baseline shows that applying only a simple consistency penalty is lesser effective, and that learning with a spatial structure such as in GLRDN-L2 is more effective as a regularization. The fact that the performance of GLRDN-L1 is degraded compared to that of the baseline suggests that it is more suitable to measure the errors in the L2 norm than in the L1, norm for the COCO dataset.

The results for all scales - AP_S, AP_M, and AP_L - show that our regularization degrades the performance for small objects, denoted by AP_S, and significantly improves it for large objects, denoted by AP_L.

Comparison on Cityscapes Instance Segmentation

We demonstrate the effectiveness of our method on other datasets by showing the results of the experiments on the Cityscapes dataset. We train the model by changing λ_g, and the best results with the val-dataset are summarized in TABLE.5.3. The optimal values of λ_g for the GLR, GLRDN-L1, and GLRDN-L2 methods are 10.0, 0.1, and 10.0, respectively. As the table shows, GLRDN-L2 is effective even for the high-resolution images of the Cityscapes dataset.

5.3.3 Qualitative Results

We show the output results of the final mask in Fig. 5.5 and Fig. 5.6. Different instances are shown in different colors. The left column shows the results of the baseline method, and the right column those of our GLRDN-L2 method. Particularly

TABLE 5.3: Comparison of baseline and several regularization methods on Cityscapes val dataset. "baseline" is CondInst [56] without regularizer, GLR is general Laplacian regularizer, "GLRDN-L2" is proposed regularizer, and "GLRDN-L1" is the modification of proposed method to L1 norm.

regularization	AP
baseline	36.5
GLR	36.8
GLRDN-L1	36.1
GLRDN-L2	**37.1**

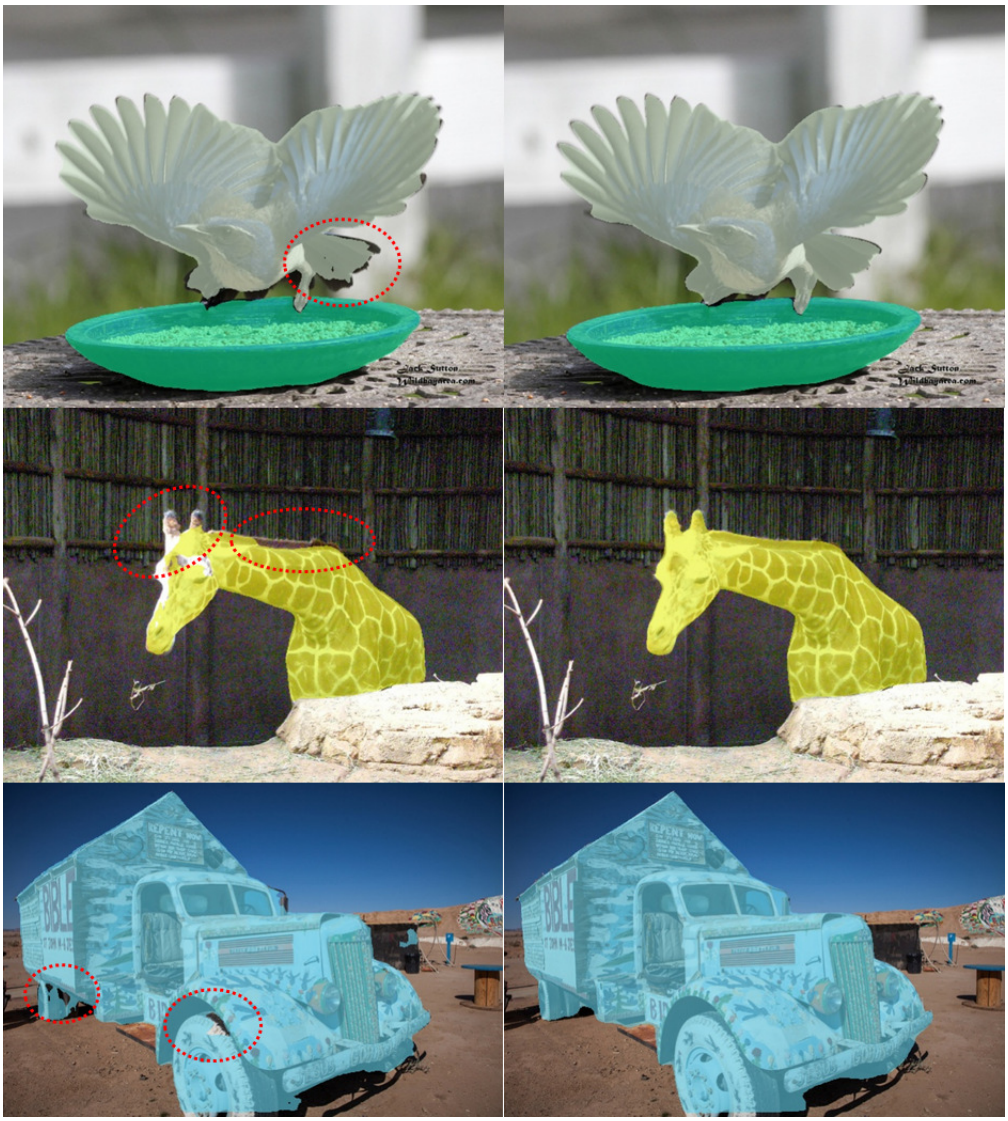


FIGURE 5.5: Successful examples of improvements in the foreground and background boundaries. Instance mask results of baseline (left) and our proposed regularization GLRDN-L2 (right) on COCO val dataset.

for large objects, it is notable that the hollowing is suppressed and the boundary details are segmented well by GLRDN-L2. Fig. 5.5 shows an example of a successful improvement in the masking of the boundary with the background, and Fig. 5.6 shows an example of an improved boundary mask between classes in an area overlapping with another class.

Also, we searched failure cases which are very few. The examples of such cases are shown in Fig.5.7 some samples that failed to predict the mask. Our GLRDN-L2 method does not work well when the predicted masks are far from the targets due to the lack of differences between the neighboring pixels.

FIGURE 5.6: Successful examples of improvements at the boundaries with other classes. Instance mask results of baseline (left) and our proposed regularization GLRDN-L2 (right) on COCO val dataset.

5.3.4 Ablation Study

We aggregated the output logit values of the mask head for all pixels in the bounding box of an instance to observe the changes in the discriminant space due to different regularizations. We did not use all images in the dataset for the observations, instead we selected 100 bounding boxes randomly in advance from the COCO val dataset as the observation targets.

The histogram of the aggregated data is shown in Fig. 5.8. There are three types of observation targets: baseline (without regularization), GLRDN-L1, and GLRDN-L2, which are displayed in red, green and blue, respectively.

In the case of the L2 norm, the variance within the positive class is smaller than in the baseline; therefore, we believe that the regularization provides the expected consistency of the labels. In the case of the L1 norm, the variance between the classes is larger than in the baseline, particularly for the negative class, suggesting that the L1 norm effect results in a sparse mask.

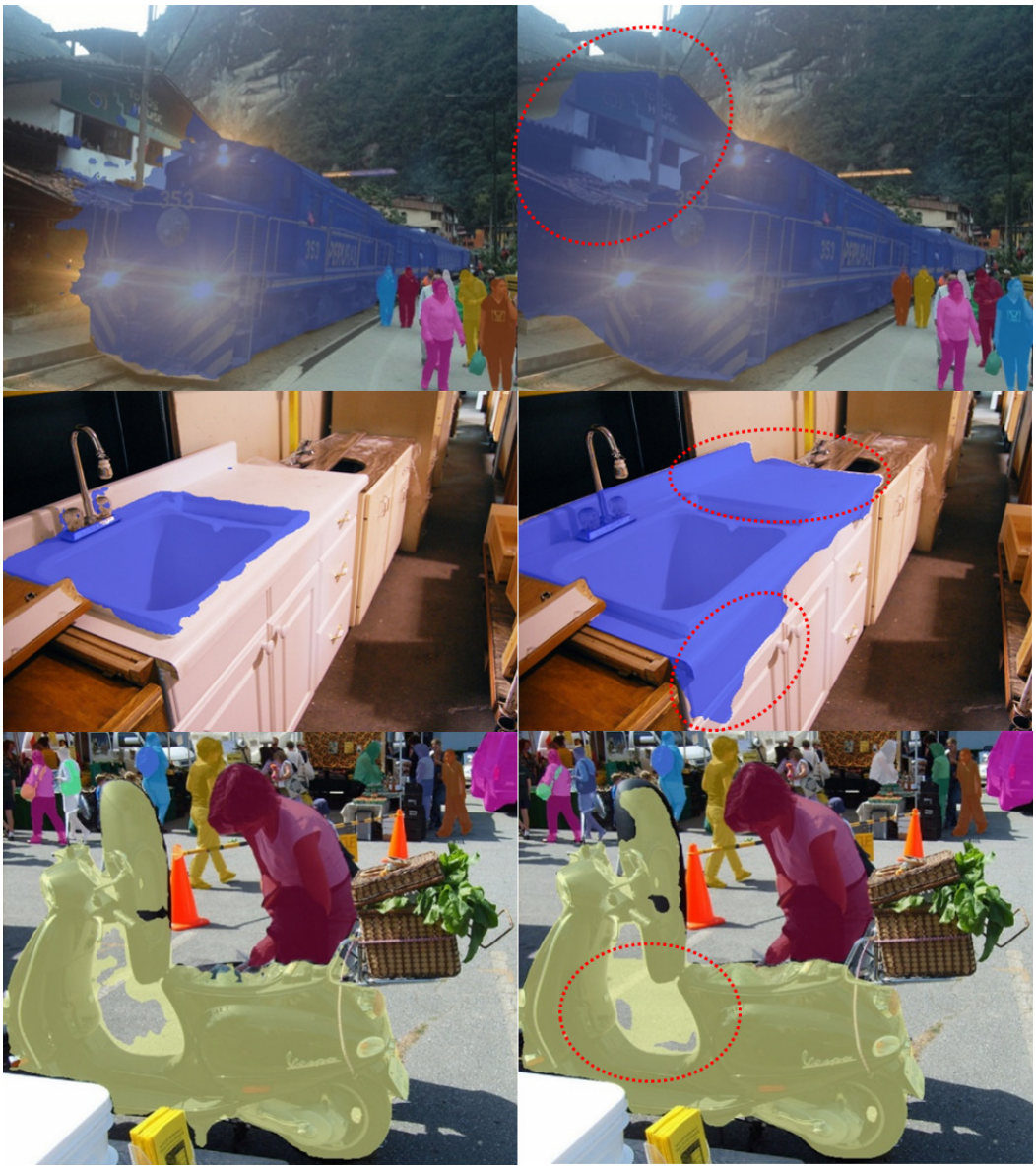


FIGURE 5.7: Failure examples where the proposed method has reduced the quality of the mask in the COCO val dataset. Instance mask results for the baseline (left) and our proposed regularized GLRDN-L2 (right).

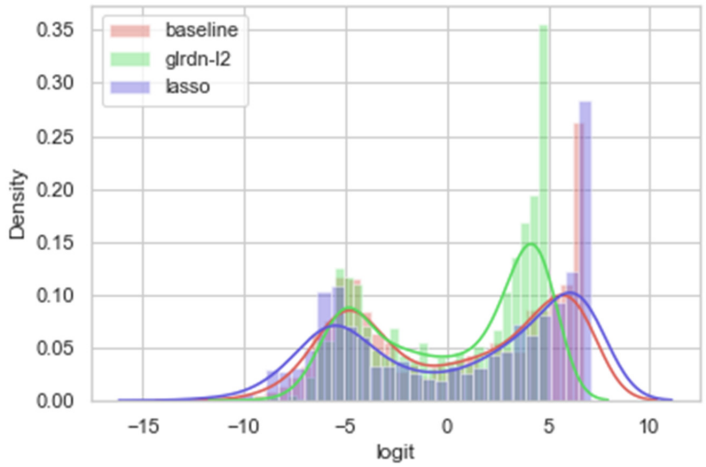


FIGURE 5.8: Histogram of posterior probabilities of mask prediction.

Chapter 6

Conclusion

6.1 Summary

In this research, our primary objective was to investigate the potential improvements in object detection and instance segmentation by introducing previously unavailable information as additional guidance. Our proposed methodologies revolved around three key aspects: "Grid Boundary Information," "Foreground and Background Information," and "Foreground and Background Boundary Information," all serving as prior knowledge for the models.

Chapter 1 showed that deep learning-based object detection and instance segmentation applications are already widely operational in the real world. We then discussed the impact of misrecognition and false positives that often occur in these applications on real-world operations. We also gave an overview of how our proposed approach reflects the challenges in different practical scenarios.

Chapter 2 delves into the history of object detection methods since the 1990s and traces the evolution of network architectures and loss functions, which have developed dramatically since the advent of deep learning. Moreover, we emphasize the intricate relationship between object detection and instance segmentation. We elucidate how these two tasks have nurtured each other's development, with innovations from one domain inspiring advancements in the other. This mutual influence has fostered a dynamic environment for the continual improvement of both tasks. Furthermore, we introduce benchmark datasets commonly adopted for evaluation in the field. These benchmark datasets serve as essential tools for assessing the performance and progress of object detection and instance segmentation methodologies, providing a standardized means for comparing and benchmarking various approaches.

Chapter 3 focused on the vulnerability of object detection models to minor translations in input images and identified grid boundaries and object position relationships as the underlying factors. To address this weakness, we introduced two modules: SGFEM, aimed at feature extraction from grid boundaries, and GADA, designed to align object centers with dropped grid boundaries. The combination of these methods demonstrated the efficacy of mitigating the model's translation susceptibility and enhancing its generalization performance.

Chapter 4 presented a novel framework, SODet, designed for two-step model learning. In the first step, the model was trained for instance segmentation, enabling the creation of a background mask-capable model. In the second step, we integrated the generated background mask into the input image to facilitate a feedback-driven retraining process, allowing us to incorporate foreground and background prior information. This innovative approach resulted in more detailed recognition capabilities and demonstrated performance improvements across multiple benchmarks.

In Chapter 5, we emphasized that conventional instance segmentation methods predominantly focus on pixel-wise errors, neglecting valuable spatial information related to boundaries with the background or other objects. We introduced a methodology that considered the spatial structure as information composed of relationships between neighboring pixels. This spatial structure information was formulated as a graph Laplacian regularization based on the differences between pixel pairs. This regularization provided the model with the capability to produce sharper masks at boundary regions, thus enhancing the overall quality of instance segmentation outputs.

6.2 Future Works

This research has addressed the challenging fields of object detection and instance segmentation, leveraging the recent advancements in deep learning. However, as outlined earlier, there remains a demand for further progress in practical applications. We firmly believe that harnessing untapped prior knowledge, as investigated in this work, is a vital perspective for the continuous advancement of these domains.

Looking ahead, several avenues for future research beckon us. First and foremost, we are considering the extension of our methods to video data. Incorporating temporal information, such as the lack of changes between adjacent frames or exploiting inertia in rigid objects, remains uncharted territory and holds promise for improved performance in dynamic settings.

Moreover, the incorporation of multi-scale features deserves attention. Existing approaches often introduce artificial structures to handle maps of varying resolutions, but there is still room for improvements in capturing the interplay between different scales by imposing resolution-related constraints.

In the context of the rapidly advancing technological landscape of contemporary society, I consider myself privileged to be engaged in research within this field. The pursuit of knowledge in object detection and instance segmentation has become my vocation, and I wholeheartedly commit to making further contributions to the ongoing advancements in these domains.

As the realm of computer vision continues to expand its practical applications in the real world, the demand for increasingly sophisticated and innovative technologies has grown exponentially. My proximity to the realms of development places me in a favorable position to readily incorporate research outcomes into tangible products.

It is my earnest aspiration to continue forging new frontiers, fostering collaborations with fellow researchers, and pushing the boundaries of object detection and instance segmentation.

Collectively, we aspire to help bridge the chasm between the current state of technology and the evolving requisites of practical applications, ultimately contributing to the creation of a safer and more secure world.

Bibliography

[1] Aharon Azulay and Yair Weiss. "Why do deep convolutional networks generalize so poorly to small image transformations?" In: *arXiv preprint arXiv:1805.12177* (2018).

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020).

[3] Daniel Bolya et al. "Yolact: Real-time instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9157–9166.

[4] Zhaowei Cai and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162.

[5] Anadi Chaman and Ivan Dokmanić. "Truly shift-equivariant convolutional neural networks with adaptive polyphase upsampling". In: *2021 55th Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2021, pp. 1113–1120.

[6] Hao Chen et al. "BlendMask: Top-down meets bottom-up for instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8573–8581.

[7] Kai Chen et al. "MMDetection: Open mmlab detection toolbox and benchmark". In: *arXiv preprint arXiv:1906.07155* (2019).

[8] Xinlei Chen et al. "Tensormask: A foundation for dense object segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2061–2069.

[9] Marius Cordts et al. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[10] Jifeng Dai et al. "Instance-sensitive fully convolutional networks". In: *European Conference on Computer Vision*. Springer. 2016, pp. 534–549.

[11] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[12] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[13] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[14] Kaiwen Duan et al. “Centernet: Keypoint triplets for object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6569–6578.

[15] Logan Engstrom et al. “Exploring the landscape of spatial robustness”. In: *International conference on machine learning*. PMLR. 2019, pp. 1802–1811.

[16] “Face matching through information theoretical attention points and its applications to face detection and classification”. In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE. 2000, pp. 34–39.

[17] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. “A discriminatively trained, multiscale, deformable part model”. In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee. 2008, pp. 1–8.

[18] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient graph-based image segmentation”. In: *International journal of computer vision* 59 (2004), pp. 167–181.

[19] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.

[20] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[21] Zheng Ge et al. “Yolox: Exceeding yolo series in 2021”. In: *arXiv preprint arXiv:2107.08430* (2021).

[22] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[23] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[24] Lukman Hakim, Huipeng Zheng, and Takio Kurita. “Improvement for Single Image Super-resolution and Image Segmentation by Graph Laplacian Regularizer based on Differences of Neighboring Pixels”. In: *International Journal of Intelligent Engineering and Systems* 15.1 (2022), pp. 95–105.

[25] David Hallac, Jure Leskovec, and Stephen Boyd. "Network lasso: Clustering and optimization in large graphs". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 387–396.

[26] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[27] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[28] Kaiming He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[29] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[30] Zhaojin Huang et al. "Mask scoring r-cnn". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6409–6418.

[31] Alexander Kirillov et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[32] Tao Kong et al. "Foveabox: Beyound anchor-based object detection". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 7389–7398.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[34] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[35] Youngwan Lee and Jongyoul Park. "Centermask: Real-time anchor-free instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13906–13915.

[36] Yi Li et al. "Fully convolutional instance-aware semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2359–2367.

[37] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[38] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[39] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[40] Shu Liu et al. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.

[41] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[42] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[43] Marco Manfredi and Yu Wang. “Shift equivariance in object detection”. In: *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer. 2020, pp. 32–45.

[44] Kemal Oksuz et al. “Imbalance problems in object detection: A review”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3388–3415.

[45] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[46] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[47] Shaoqing Ren et al. “Faster R-CNN: towards real-time object detection with region proposal networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016), pp. 1137–1149.

[48] Hamid Rezatofighi et al. “Generalized intersection over union: A metric and a loss for bounding box regression”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 658–666.

[49] Nermin Samet, Samet Hicsonmez, and Emre Akbas. “HoughNet: Integrating near and long-range evidence for bottom-up object detection”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 406–423.

[50] “Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image”. In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 1998, pp. 70–75.

[51] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[52] Bharat Singh and Larry S Davis. "An analysis of scale invariance in object detection snip". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3578–3587.

[53] Carole H Sudre et al. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[54] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[55] Mingxing Tan, Ruoming Pang, and Quoc V Le. "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

[56] Zhi Tian, Chunhua Shen, and Hao Chen. "Conditional convolutions for instance segmentation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 282–298.

[57] Zhi Tian et al. "Fcos: Fully convolutional one-stage object detection". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9627–9636.

[58] Robert Tibshirani et al. "Sparsity and smoothness via the fused lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1 (2005), pp. 91–108.

[59] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International journal of computer vision* 104 (2013), pp. 154–171.

[60] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[61] Paul Viola and Michael J Jones. "Robust real-time face detection". In: *International journal of computer vision* 57 (2004), pp. 137–154.

[62] Xinlong Wang et al. "Solo: Segmenting objects by locations". In: *European Conference on Computer Vision*. Springer. 2020, pp. 649–665.

[63] Xinlong Wang et al. "Solov2: Dynamic and fast instance segmentation". In: *Advances in Neural information processing systems* 33 (2020), pp. 17721–17732.

[64] Enze Xie et al. "Polarmask: Single shot instance segmentation with polar representation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12193–12202.

[65] Wenqiang Xu et al. "Explicit shape encoding for real-time instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5168–5177.

[66] Hongyi Zhang et al. “mixup: Beyond empirical risk minimization”. In: *arXiv preprint arXiv:1710.09412* (2017).

[67] Richard Zhang. “Making convolutional networks shift-invariant again”. In: *International conference on machine learning*. PMLR. 2019, pp. 7324–7334.

[68] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. “Bottom-up object detection by grouping extreme and center points”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 850–859.