Holistic, Instance-level Scene Parsing with Deep Neural Networks

Qizhu Li

St Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2020

To my dearest grandma, who I know is always watching over me

Declaration

I hereby declare that this thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

November 2020

Qizhu Li

Acknowledgements

My uttermost gratitude goes to my supervisor Prof. Phil Torr, for bestowing upon me the unparalleled opportunity to pursue a doctorate in one of the best research institutions around the world, for showing confidence in me always, and for filling me with strength and energy whenever they run low. This thesis would not have been possible without his unwavering support and ingenious creativity.

During my time in the Torr Vision Group, I have received much help and guidance from fellow group members, for which I am eternally grateful for. In particular, I would like to thank Dr Anurag Arnab, with whom I had the pleasure of working closely together for multiple years. His patient advice and counsel in the early years of my doctorate studies have been truly invaluable. His meticulous attitude towards research and burning passion for uncovering the unknown have deeply influenced my attitude and approach towards research, and are instrumental for making this thesis possible. When he is not absorbed in his work, his ability to come up with a joke anytime anywhere has also been a constant source of joy and amazement throughout my time here.

Besides, my heartfelt appreciation goes to another collaborator and mentor figure of mine, Dr Xiaojuan Qi. Her selfless devotion to the career of scientific research and her vast pool of knowledge have been inspiring and energising. Her approachable personality and willingness to share

have induced many intense but fruitful discussions and debates, which profoundly enriched this thesis.

I am also immensely grateful for my Transfer and Confirmation examiners, Prof. Andrea Vedaldi and Prof. Victor Prisacariu, whose constructive and insightful feedback along the way have provided the much needed fresh opinion, and propelled this thesis to reach new heights.

Without a doubt, my time here would not have been so enjoyable without the support and company of my friends. Both inside and outside the research office, I have been fortunate to meet and befriend many great people. I am deeply thankful to Namhoon, Arslan, Li, Yifu, Xiaonan, Ziqi, Xinlei, Xuanli, Xiaoyu, Honglie, Siying, and many others.

My sincere appreciation also goes to Ms Joanna Zapisek, the project manager, and Mrs Cassandra Warren, the group administrator, for always taking good care of us. I would also like to thank Mr Jeremy Daniel, our cluster manager, and the ARC Team, for working tirelessly to support the computing needs of the research group.

No words can describe my gratitude towards my parents, who have made many sacrifices and been extremely supportive throughout my studies. Without the constant love and care from them and the other family members, I would not be who I am today, nor would there be this thesis. My debt to them is beyond measure.

Last but not least, I am grateful to Huawei Technologies Co., Ltd. whose generous funding has made this thesis possible.

Abstract

Humans have long fantasised about the notion of a machine that sees and interprets a scene in ways that simulate the human visual system. As effortless as it appears to us, however, making sense of a scene is highly challenging for computers, due to the unbounded complexity and unlimited variations of a real-world scene. This thesis focuses on solving instance-aware pixel-level scene understanding with deep neural networks, which have potential applications in autonomous navigation, security systems, object counting, medical image analysis, amongst many others.

This thesis first proposes an instance-level human parsing network, which is capable of producing category-, instance-, and part-level segmentation of human in a single forward pass. The proposed network is trained end-to-end given detections, and exploits a differentiable conditional random field (CRF) defined over a dynamic number of part instances for every image. It is the first work to perform human parsing at an instance level and can be trivially adapted to parse other objects. At the time of publication, it achieves state-of-the-art performance on public leaderboards.

Prompted by the large amount of labour and cost required for annotating a pixel-level dataset, this thesis then seeks to reduce the reliance of instance-level scene parsing methods on fully labelled examples, by proposing a weakly-supervised training strategy that makes use of

image-level tags and bounding boxes as supervision. Using the proposed weak supervision scheme leads to massive reductions in required annotation time per image and attains a high percentage of the performance achieved by fully supervised oracles. This thesis also presents the first weakly-supervised panoptic segmentation model.

Finally, realising that a CRF-based instance segmentation pipeline has several limitations in terms of optimisation and capacity, this thesis presents a new panoptic segmentation pipeline which exploits a fully connected – and yet lightweight – instance affinity term. Unlike the prior art, this approach can directly supervise panoptic segmentation outputs and train end-to-end, thanks to a differentiable mechanism for propagating panoptic logits according to predicted instance affinities. At the time of publication, this method also obtains state-of-the-art results on public leaderboards.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans are genetically blessed with an innate ability to parse and make sense of the visual surrounding. With little conscious effort and high robustness, we recognise and track objects in a visual scene, discern the functionalities of features and elements, perceive three-dimensional shapes from stereoscopic vision, and make forecasts as to how the scene will likely evolve in the near future. Vision provides us with an essential source of information and serves as a major connection to our surrounding space.

It is thus no wonder that, with the proliferation of computers, the possibility and prospect of a machine that sees like us has intrigued countless minds of the scientific community and the general public. Contrary to the human experience, computers receive visual information in the form of images, which are discrete grids of pixels, each having one or multiple values representing the colour and intensity. Despite how naturally visual understanding happens for us, however, it has been difficult to distil our visual reasoning process into a comprehensive system of rules, following which a computer will be able to interpret visual inputs like us. Indeed, the high level of variability and complexity found in real-world scenes, in terms of the object size, orientation, lighting condition, pose, occlusion,

etc., have posed substantial challenges for realising the aspiration. To overcome these obstacles, the discipline of computer vision emerges at the crossroads of information technology and cognitive science, and carries huge significance for both theory and application.

1.1 Holistic, Instance-level Scene Parsing

This thesis focuses on holistic, instance-level scene parsing. In computer vision, to parse a scene is to gain a pixel-level understanding of its contents. There are multiple levels of information which can be extracted by parsing a scene. Starting from the coarse end of the spectrum, parsing an image at the *category level* involves classifying every pixel into a semantic category, and such a task is commonly referred to as *semantic segmentation*. The output of this task informs an agent of the kinds of objects present in a scene and can be exploited to enable applications such as drivable lane detection, computer-aided medical image analysis, and photo retouching.

While semantic segmentation makes distinctions between categories, it cannot differentiate between multiple objects of the same class, or count the number of objects in a scene. To achieve this, an *instance-level* understanding of the scene is required. Parsing a scene at the instance-level associates individual pixels to an object instance to which it belongs, in addition to a semantic class. Tasks that fall under the umbrella of instance-level scene parsing include instance segmentation and panoptic segmentation, with the key difference in how they handle "stuff" classes. Instance-level scene understanding has proven to be crucial for enabling a new class of applications that are concerned with individual objects in a scene, such as object counting, video surveillance, and autonomous driving.

While the task of semantic segmentation has been extensively studied, the field

of instance-level segmentation has seen relatively fewer research attempts, and has lagged behind the former in terms of maturity. This thesis seeks to address certain challenges in instance-aware scene parsing and aims to approach the task *holistically* to ensure coherent, non-overlapping segmentations.

1.2 Challenges in Scene Parsing with Computer Vision

From designing the architecture, training the network, to deploying the model, scene parsing with deep neural networks is a complex process that faces many challenges in its various aspects and phases. In this section, we will describe some of the pressing issues facing the field.

Variability in scene and objects. A scene can appear very different depending on the viewing position, lighting, weather conditions. Similarly, the look of an object can change significantly depending on how, where, and when an observation of it is made. When it is viewed at different distances from the camera, its apparent size (and hence the amount of details captured) in an image can vary massively. To further complicate the situation, objects of the same semantic class often manifest a multitude of appearance variation. For example, basketball, football, golf ball, and tennis ball are all in the "ball" class, but they display very different texture, colour, and size. Sometimes, the reverse is true: certain objects may have similar appearance traits, but fall under totally different categories. An orange, for example, appears ball-like, but no human annotators would label it as a "ball". Overall, humans are extremely good at abstracting away structures and extracting common characteristics (*e.g.*, "balls" are round objects for entertainment purposes), but achieving this level of robustness is extremely difficult for computers.

Curse of the dataset. Neural networks are said to be "data-hungry" – they demonstrate an appetite for massive amounts of labelled examples and an ability to learn rich representations by observing them. Recently, the emergence of large-scale datasets, together with increased compute powers, has given birth to the successful modern deep neural networks for computer vision today. However, access to large datasets has also shaped the current generation of deep learning methods in such a way that they rely on a massive amount of hand-labelled data to achieve state-of-the-art performance. The availability of datasets, or the lack thereof, not only dictates what tasks can be tackled (*e.g.*, to train instance segmentation, an instance segmentation dataset is needed) but also what deployment environments these algorithms can operate safely and reliably in, due to biases in data (*e.g.*, an autonomous driving algorithm trained on European road scenes may work poorly in Asian streets). This is especially problematic for scene parsing, as its pixel-level nature means that annotation has to be done at the pixel level too to fully supervise a neural network for scene parsing, requiring a huge amount of labour and cost.

Hierarchical scene understanding. Humans naturally interpret scenes in a hierarchy: from the broad categorisation of a scene, identifying the objects in it, comprehending the constituent components which form the observed objects, to finally inferring and registering the various attributes and functionalities of objects and their parts. A hierarchical understanding of a scene is crucial for carrying out even the simplest interactions – by human standards – with the world. For example, to travel across the living room and pick up a blue mug requires identifying the living room (*i.e.*, scene categorisation), locating the mug (*i.e.*, object recognition), checking its colour (*i.e.*, attributes), recognising the mug's handle (*i.e.*, parts), and working out how to pick it up (*i.e.*, functionalities). Most scene parsing methods only work at the category level and instance level, and are not designed with the

necessary architecture to produce a rich pyramid of information.

Unified and efficient network design. Much effort has gone into designing performant neural networks for the various types of scene parsing, and over the years, specialised architectures have been developed to tailor towards each task with optimal performance. However, as we progress and consider more advanced tasks, it is often the case that, to obtain a comprehensive understanding of a scene, one needs to gather predictions for tasks which are currently handled by very different networks. For example, the task of panoptic segmentation involves obtaining an instance-level understanding of countable objects, and a category-level understanding of other objects. While it can be done by employing a semantic segmentation network and an instance segmentation network, it is highly inefficient. On the other hand, naively training a neural network with a shared backbone and task-specific prediction modules often leads to reduced performances for several, if not all, tasks compared to single-task baselines [67]. Yet, it is reasonable that a network trained on multiple tasks should be able to discover and exploit synergies across tasks, and thus perform better than its specialised counterparts. The question thus arises on how we can design a unified and efficient architecture that performs well for a wide range of tasks, and what is the best strategy to train a model on multiple tasks without losing performance.

1.3 Approach

To address the aforementioned issues in scene parsing, we explore instance-level segmentation methods which are based upon deep neural networks.

We first work on achieving hierarchical scene parsing by focusing on the human as the subject. For complex robot-human interactions to take place safely and reliably, it is important that machines have an accurate and fine-grained un-

derstanding of the nearby humans. Up to this point, there have been algorithms developed to semantically parse human into body parts. However, they do not produce instance-level prediction, which is essential for performing interactions in a setting with multiple humans.

To achieve instance-level human parsing, we adopt an end-to-end neural network architecture, in which a recurrent neural network (RNN) component performs mean-field inference of CRFs in a differentiable manner. We develop what to our knowledge is the first instance-level human parsing algorithm, which has multiple advantages over alternative instance-level segmentation methods: it allows all segments in an image to be considered globally and jointly, assigns each pixel to a unique label by design, and produces coherent segmentation map without requiring additional processing.

We then turn our attention to the hunger for large-scale, densely labelled datasets exhibited by state-of-the-art instance segmentation methods. While the ability to learn rich representations from large datasets has been a strength of deep neural networks, the fact that such datasets are costly to obtain, however, has placed significant limitations on the quantity and quality of data available. With this in mind, we develop a novel training strategy for instance-level segmentation that, instead of using fully-labelled pixel-wise ground truth masks, exploit weak labels. We first extract the information in the weak labels to obtain coarse segmentation maps. Then, through a process based on Expectation Maximisation (EM) and self-training, we iteratively refine the coarse labels. The refined labels are used to train our final instance-level segmentation network.

Finally, as we become aware of the limitations of a CRF-based instance segmentation network, we focus on developing a new algorithm that has the benefits of a CRF model while being free from its shortcomings. Instead of using CRFs, we introduce a novel dense instance affinity operation, that is efficient, expressive,

and fully learnt from data. With this instance affinity operation at the heart of our network, we present a new method for performing panoptic segmentation – an instance-level task that involves fully segmenting an image to "stuff" regions, and "thing" instances.

1.4 Thesis outline

The remaining chapters of this thesis are structured as follows:

Chapter 2 Chapter 2 presents a review of relevant literature in the area of scene parsing, including semantic segmentation, instance segmentation, panoptic segmentation, and fully connected CRFs in scene parsing methods.

Chapter 3 Chapter 3 seeks to holistically address the task of human parsing at the category, instance, and part level, using a deep neural work that exploits conditional random fields (CRFs)-based probabilistic graphical modelling.

Chapter 4 Like many other works on instance-level segmentation, the method proposed in Chapter 3 requires fully labelled pixel-level annotation, which can be very expensive to acquire in terms of both time and cost. Aware of this, in Chapter 4, we present a training strategy for category- and instance-level segmentation that exploits weak labels in the form of image-level tags and bounding boxes. Furthermore, by extending an instance segmentation framework, we arrive at a pipeline that is capable of performing panoptic segmentation, which we then train with only weak labels.

Chapter 5 Both of the methods in previous chapters make use of two separately trained networks for detection and segmentation respectively, making the approach

not fully end-to-end and inefficient. Besides, their key component - the fully connected CRF-based graphical modelling - suffers from sensitivity to hyperparameters and limited learning capability. To address these issues, in Chapter 5 we propose a unified and efficient pipeline for panoptic segmentation by exploiting a densely connected instance affinity, eliminating the need for a post-processing merging step commonly used in state-of-the-art methods.

Chapter 6 Finally, in Chapter 6, we conclude this thesis by summarising the key contributions of each chapter and discussing questions that remain unanswered and potential future directions.

1.5 Contributions

Chapter 3: Holistic, Instance-level Humaning Parsing In traditional object parsing, only a category-level segmentation of object parts is performed. However, to comprehend a scene containing multiple objects and carry out advanced robotic interactions, having only category-level knowledge is often not sufficient. This Chapter presents, to our knowledge, the first work to perform *instance-level* human parsing. For each person in the image, not only do we produce semantic segments of various body parts, but we also predict his or her instance mask such that every body part is associated with a human instance. Furthermore, in contrast to other popular works on instance-level segmentation, we jointly consider all parts and instances in an image together, handle a varying number of people, and produce non-overlapping segments. Given detections, our network is trained end-to-end and can produce a hierarchy of different levels of segmentation with just a single forward pass. Finally, we benchmark our model on the challenging Pascal Person Parts dataset and achieve state-of-the-art results in instance-level part and human segmentation, and competitive results in category-level part segmentation.

Chapter 4: Weakly- and Semi-Supervised Panoptic Segmentation This chapter proposes a strategy to train an instance-level segmentation network with weak labels. Instead of requiring fully-labelled pixel-level ground truth, image-level tags (*i.e.*, binary labels that indicate presence or absence of an object category in an image) and bounding boxes are used to supervise the training of "stuff" and "thing" classes respectively. On the Cityscapes dataset, this weakly-supervised training strategy reduces the time required to annotation the ground truth by a factor of up to 30. Despite this huge reduction in annotation cost, our experiments show that this training strategy - based on the ideas of self-training and Expectation Maximisation (EM) - is able to achieve up to over 95% of fully-supervised performance on several popular segmentation benchmarks. Furthermore, our framework also readily allows for mixed-use of weak and full labels, *i.e.*, semi-supervised learning. Lastly, we extend the instance segmentation model proposed in [6] to perform panoptic segmentation and train the network with the proposed scheme of weak supervision. This is, to the best of our knowledge, the first weakly supervised panoptic segmentation model on Cityscapes.

Chapter 5: Unifying Training and Inference for Panoptic Segmentation This chapter proposes a lightweight and parametrised panoptic segmentation submodule that exploits an end-to-end learnt dense instance affinity. Most prior art requires a heuristic merger operation to combine semantic and instance segmentation results and produce panoptic predictions. As the network is not aware of this merger step during training, it may lead to sub-optimal performance. In contrast, our dense instance affinity – a predicted probability for any pair of pixels to belong in the same "thing" instance or "stuff" class – establishes a novel mechanism that can *differentiably* produce coherent panoptic segmentation without the need for any post-processing steps. This enables us to unify the training and inference archi-

tecture of the network, and directly optimise the panoptic segmentation during training, in an end-to-end manner. Furthermore, this fully connected instance affinity is computed efficiently, adding minimal computation burden to the full network. With this novel architecture, we are able to set new performance records for panoptic segmentation on Cityscapes and COCO. Finally, our flexible framework can readily work with or without an internal mask prediction branch. For applications which has a limited computation budget, this can be of interest.

1.6 Publications

The following publications form the individual chapters of this thesis:

Chapter 3

> Qizhu Li*, Anurag Arnab*, Philip H.S Torr. Holistic, Instance-level Human Parsing. *British Machine Vision Conference (BMVC)*, 2017.
> * Joint first authors

Chapter 4

> Qizhu Li*, Anurag Arnab*, Philip H.S Torr. Weakly- and Semi-Supervised Panoptic Segmentation. *European Conference on Computer Vision (ECCV)*, 2018.
> * Joint first authors

Chapter 5

> Qizhu Li, Xiaojuan Qi, Philip H.S Torr. Unifying Training and Inference for Panoptic Segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Chapter 2

Background

Early research on neural networks and the backpropagation mechanism have been conducted in the 1980s [131, 81]. However, it is only in recent years that, fuelled by the availability of large-scale datasets [36, 39] and great leaps in GPU-based compute capabilities, neural networks gain tremendous momentum [77, 139, 60] and become the de facto tool for approaching computer vision tasks [110, 127, 59].

In this chapter, we give a brief account of the advances and developments in scene parsing using deep neural networks. We start with methods for category-level understanding and then go on to instance-level methods including those for instance segmentation and panoptic segmentation. Finally, we describe the usage and mechanism of end-to-end CRF inference in semantic and instance segmentation networks, which is integral to the work in Chapter 3 and 4, and motivates the work in Chapter 5.

2.1 Semantic Segmentation Networks

Semantic segmentation is a pixel-level extension of the image classification task. It involves uniquely classifying every pixel in an input image into an object category. Several metrics exist for evaluating semantic segmentation performance, but the

most common metric by far is the mean intersection over union (mean IoU or mIoU). IoU for a single category is defined as the ratio between the number of correctly predicted pixels for this category (the region where prediction and ground truth agree, *i.e.*, their intersection) and the total number of pixels which are either annotated or predicted as the category of interest, or both (*i.e.*, their union). To produce the mean IoU, per-category IoU scores are computed and averaged across all categories.

In their 2015 work, Long *et al*. [110] revolutionalise the field by solving the problem with a fully convolutional neural network for the first time. The important finding that a classifier can be readily converted into a performant semantic segmentation network means that progress made in the classification field can directly benefit semantic segmentation, and researchers can focus on the unique challenges of semantic segmentation, rather re-inventing the wheel. Despite achieving state-of-the-art performance upon publication, however, it suffers from a few limitations. The aggressive downsampling operation coarsens the classification map and leads to blob-like predictions lacking fine details. Also, it does not have an effective mechanism for exploiting contextual information, either globally or locally.

Many works after [110] have sought to address these issues. Dilated convolution [162] is proposed as a drop-in replacement for normal convolutions situated downstream of a removed downsampling operation. Removing a downsampling operation increases the feature resolution. To apply the same convolution kernel on a larger feature map, dilation is needed to increase the absolute field of view (FoV) of the convolution, to keep the relative FoV unchanged, which is the ratio between the sliding-window size and the input feature resolution. Keeping this ratio unchanged is essential as it protects the validity of transfer learning from a pre-trained classifier to the segmentation network.

Another line of research has focused on global and local feature aggregation to

exploit contextual cues. It is first observed by Liu *et al*. [107] that the information contained in the mean global feature helps guide the network to make contextually sensible predictions. For example, knowing that an image is mostly "dining table" can guide the network towards predicting "dishes" rather than "vehicles". Later works extend the idea and propose multi-scale feature aggregation. Various implementations are proposed, mainly involving the use of convolutions with different dilation rates [21], and poolings at different window sizes [170].

More recently, the community has shifted towards the use of self-attention mechanism, popularised by works in the natural language processing (NLP) area [148], for propagating contextual information over long ranges. In [163, 45], the correlation between feature vectors at different spatial locations are taken as the pairwise attention strength, and features are propagated accordingly. A different but related approach is taken by Zhao *et al*. [171] which convolutionally predicts the pairwise attention strengths.

Despite their promising performance, a common limitation for these methods is the quadratic complexity for memory and computation with respect to the total number of pixels, caused by the fully connected pairwise attention. Several attempts are made to address the high complexity issue, from an approximate factorisation of the self-attention matrix [135], unrolling the self-attention mechanism approximately into a series of less costly operations [66], to a locally-connected self-attention mechanism [166].

Apart from the advances in terms of architecture, much research effort has also gone into finding a better loss. The widely used cross-entropy loss does not directly optimise the mean IoU metric, with the trained network tending to favour common and large objects over rare and small ones. However, the fact that the mean IoU is not differentiable and has to be computed over the entire dataset indicates that it is not amenable to direct optimisation. Several surrogate losses have been

proposed [15, 12, 82], with varying levels of success.

2.2 Instance Segmentation Networks

Before covering the past literature in the field of instance segmentation, it is important to make a distinction between "thing" and "stuff" – a concept which will also be useful later for panoptic segmentation. "Things" are *instantiable* objects, and can appear in different quantities. Examples of "things" are pedestrians, cars, and bicycles. Note how these objects all have well-defined shapes and can be unambiguously counted. On the other hand, "stuff" is uncountable and amorphous, and typically include classes such as the sky and ground.

Instance segmentation is solely concerned with "thing" classes, as it aims to find out if a pixel belongs to a "thing" object, and if so, associate it with the correct semantic category and object instance. Performance of instance segmentation is commonly measured by figures of mean average precision (mAP or mean AP), which is a ranking-based metric. The computation of this metric accepts an unlimited the number of predicted instances, and does not enforce unique prediction for each pixel. Most methods for this task can be broadly clustered into two groups, depending on whether they use region proposals. We will first introduce these two types of approaches, followed by a review of methods which cannot be aptly categorised in this way.

2.2.1 Proposal-based approaches

Proposal-based approaches are built upon the success attained by region-based detection algorithms [50, 127], and follows the same overarching framework where a region proposal network (RPN) first generates a set of candidate regions from an input image, and then each candidate region is independently processed to produce

detection predictions. Due to this structure, proposal-based methods have also been described as "two-stage" and "top-down". Based on a region-based detector, instance segmentation methods adds a further predictor to produce object masks for each candidate region. There has been discussion in the research community, however, over how the predictors should be structured, with some arguing for a cascade architecture from a performance standpoint [31, 19], and others favouring a parallel approach [59] citing its superior efficiency and simplicity.

A challenge for proposal-based approaches is to handle objects of varying sizes. Lin *et al*. [100] propose a feature pyramid network (FPN) which exploits a multi-scale pyramid of features and allows for the aforementioned two-stage process to be carried out at different feature resolutions, thereby capturing a wide range of object sizes. Liu *et al*. [104] further improve the FPN architecture by introducing multiple strategies for cross-scale feature fusion, achieving significant gains in performance.

While proposal-based methods have been leading other methods on several instance segmentation benchmarks [29, 101], they suffer from a few common shortcomings. As the predictions are made independently for each candidate region, there is no guarantee that a pixel would be assigned to a unique label. Furthermore, as the masks are predicted at a fixed resolution regardless of actual object size and resized to the correct scale, the outcome often lacks details for large objects (*e.g*., predicting a blob for fingers of a person). Taking notice of this issue, Kirillov *et al*. [73] draw inspiration from the rendering operation and devise an effective and efficient pipeline that iteratively upsamples and refines object masks, adding fine details where they are most needed, producing high-quality segments that rivals those from a bottom-up segmentation network.

2.2.2 Proposal-free approaches

Another stream of studies [72, 103, 108, 9, 34] tackle the task by performing instance grouping on top of an initial category-level segmentation. Most such "bottom-up" methods use a separate network to produce instance grouping cues, such as object edges [72], breakpoints [103], a discriminative embedding [34], and instance affinity maps [108]. Additionally, they also require a post-processing step [72, 103, 108] to combine all the information, and group pixels into instances. Due to the usage of multiple networks, these methods can be time-consuming to run, and the results can be sensitive to the choice of hyperparameters in the grouping strategies.

Our work in Chapter 5 shares a similar spirit with [108] as both employ pairwise instance affinities to aid instance discrimination. However, their affinity connections are sparse and heuristically tied to fixed locations around each central pixel. In addition, their network does not directly output instance segmentation and instead requires a separate, complex graph merging procedure to generate object instances based on the predicted affinities. In contrast, in Chapter 5, we formulate a fully connected dense instance affinity operation, and incorporate it into an end-to-end framework for panoptic segmentation, eliminating the need for any post-processing.

2.2.3 Other approaches

Here we introduce some methods that do not belong in either of the two categories.

Polygon-based approaches [17, 1, 90] aim to predict instances in the form of polygons, similar to how a human annotator would draw polygons around target instances [29]. In addition to an input image, these methods typically require localisation and/or shape cues as an input, such as bounding boxes [17, 1] or coarse instance masks [90], and subsequently outputs a set of polygon vertices for each instance. However, as each instance is processed independently like a

proposal-based approach, polygon-based methods are also prone to assigning a single pixel to multiple instances and have no mechanism to resolve such conflicts.

Arnab and Torr [6] propose an alternative instance segmentation network consisting of three components – a semantic segmentation submodule, an object detector, and an instance segmentation submodule. The category-level segmentation map and the localisation cues from the first two components are combined in the third to form coarse instance masks, which are then refined by an Instance CRF operation to produce the final instance segmentation. Compared to a purely top-down approach, its fully-fledged semantic segmentation network leads to superior segmentation quality; its design prohibits assigning multiple labels to a pixel by design, and can recover from imperfect localisation cues. On the other hand, unlike a purely bottom-up approach, it conveniently and directly benefits from the rich localisation cues from a state-of-the-art detector. One of its limitations is that, however, separating the object detector from the rest of the network leaves potential room to improve for efficiency and performance. There are also issues associated with its Instance CRF operation, which we will highlight in Sec. 2.4.

2.3 Panoptic Segmentation Networks

Formally described in [71], the panoptic segmentation task aims at unifying semantic segmentation and instance segmentation. It involves classifying both "stuff" and "thing" pixels and associating "thing" pixels with their respective instances. The most commonly used performance metric is the panoptic quality (PQ), which is in turn a product of segmentation quality (SQ) and recognition quality (RQ). In contrast to instance segmentation, predicting multiple conflicting labels for one pixel is not allowed in panoptic segmentation. In [71], a two-step baseline method is proposed. In the first step, it obtains semantic segmentation and instance seg-

mentation results from a PSPNet [170] and a Mask R-CNN [59] respectively. In the second step, fusing these semantic and instance segmentation predictions according to a set of heuristics, it attempts to assign each pixel with a unique "stuff" class or "thing" instance.

One line of follow-up research is dedicated to improving the first step in [71]. Such works [35, 70, 87, 83] design a unified architecture by augmenting an instance segmentation network with a semantic segmentation branch. On top of the above architecture, attempts have been made to encourage greater coherence between the semantic and instance predictions by passing attention masks or adding a consistency loss [87, 83]. Yang *et al*. [160] design a fully convolutional framework with multiple heads for unified semantic segmentation and instance segmentation. Albeit improving efficiency and consistency, the above approaches are not amenable to end-to-end training, are sensitive to errors in semantic and instance segmentation cues, and rely on ad hoc strategies to generate panoptic segmentation results like [71].

Another line of research seeks to advance toward an end-to-end framework. Xiong *et al*. [157] propose a unified network with a non-parametric panoptic head to fuse semantic and instance segmentation predictions as part of the network architecture. However, the effectiveness of their panoptic head highly depends on the semantic and instance segmentation quality, since it is built upon hand-crafted features with no learnable weights. Also, it still requires several heuristic strategies in the inference phase, *e.g.*, a complex voting mechanism to determine the semantic categories of predicted segments.

2.4 Conditional Random Fields as Recurrent Neural Network

Probabilistic graphical modelling with conditional random fields (CRFs) provides a principled path for injecting our prior knowledge into the prediction process. For pixel-level scene parsing, CRF models are defined over individual pixels or image patches, and consist of two terms – a unary potential for each pixel or image patch, and a pairwise potential which are defined over pairs of pixels or image patches.

CRFs have been widely studied in semantic segmentation. In early works, the pairwise potential is defined for only neighbouring pixels or patches [137, 145, 53, 46]. Without the ability to pass messages over long ranges, this leads to excessive local smoothing. Follow-up works seek to address the over-smoothing by introducing hierarchical connectivity and region-based higher-order terms, but are in turn bound by the accuracy of unsupervised image segmentation on which they rely [62, 78, 74, 79]. Fully connected CRFs, which have the pairwise potential defined over all pairs of pixels in an image, are made computationally practical [76] by exploiting a highly efficient implementation for high-dimensional Gaussian filtering [2]. Taking the idea further, Zheng *et al*. re-formulate the mean-field inference of fully-connect CRFs as a recurrent neural network (RNN), which can then be trained end-to-end as part of a deep convolutional neural network [173]. Given a softmax distribution of the unary potential, each iteration of the recurrent neural network consist of five operations:

1. **Message passing.** In this step, the post-softmax unary potentials are filtered with several Gaussian filters, thus passing the unary information across all pixels in the image. In the approach proposed in [76, 173], two Gaussian filters are used, with the first one being a bilateral Gaussian filter, and the second one being a spatial Gaussian filter. Two filtered outputs are thus produced,

which will be combined in the next step. Note that this step contains three hyperparameters that control the width of the two Gaussian kernels (two for the bilateral filter, and one for the second Gaussian filter), and they cannot be practically learnt via backpropagation. In practice, they are grid-searched.

2. **Weighting filter outputs.** This step takes a weighted sum of the filter outputs from the two Gaussian kernels in the previous step and produces a single potential. To increase the number of trainable parameters, this step is implemented as one 1×1 convolution per each filtered output, with the output number of channels equal to the input number of channels, followed by an element-wise addition. Therefore, for a semantic scene parsing task with C classes, this step has $2C^2$ trainable parameters.

3. **Compatibility transform.** In this step, the output from the previous step undergoes a compatibility transform, which penalises assignments of label pairs depending on their likelihood of co-appearance. This operation is implemented as another 1×1 convolution, with the same input and output number of channels. There are thus a total of C^2 trainable parameters in this step.

4. **Adding unary potentials.** This step combines the output of the compatibility transform and the original unary potential via an element-wise subtraction. This step is parameter-free.

5. **Normalising.** The output from the previous step is passed through a softmax function and serves as either the input to the next RNN iteration or the final output, depending on the current iteration count. This step is also parameter-free.

In summary, for semantic segmentation with C classes, there are a total of $3C^2$

trainable parameters, and four hyperparameters including three Gaussian kernel widths for message passing, and one iteration count for the RNN.

Arnab and Torr [6] further adapted the idea of end-to-end trained, fully connected CRFs to the task of instance segmentation. Their proposed Instance CRF module has a similar implementation to what is described above, but contains a few key modifications to work with instances instead of semantic classes:

- **Weighting filter outputs.** In the new instance-level task, each channel of the unary potential tensor represents an instance rather than a semantic class. As a result, channels no longer take on fixed semantic meanings, and the number of instances (and hence the number of channels) varies for different images. It is thus inappropriate and indeed impossible to implement the weighted summation of Gaussian outputs as 1×1 convolutions. Instead, in the Instance CRF, two trainable scalars are used as the weighting coefficients.

- **Compatibility transform.** By the same reason, the 1×1 convolution used for the compatibility transform in semantic segmentation [173] has to be replaced. Fixed penalty assignments using the Potts model is adopted. No trainable parameters exist after this modification.

Therefore, under the Instance CRF formulation, there are only two trainable parameters, and the same four hyperparameters as those found in its semantic segmentation counterpart.

The small number of trainable parameters present in the Instance CRF module implies a rather limited ability to learn from and adapt to data. Furthermore, it has been experimentally observed that the segmentation quality can be highly sensitive to the four hyperparameters, which must be searched [76]. For scenes with a wide range of object sizes, it can become even more challenging to find a good set of Gaussian kernel widths, as large kernels tend to smooth out small

objects and small kernels have limited effects for refining large objects. Finally, there can be problems of vanishing gradients due to the iterative nature of the algorithm [173]. Indeed, its difficulty of optimisation has prompted several works to first train the unary network (without the CRF module) until convergence (or equivalently, initialise the unary network with fully trained weights). In the last step, the model is finetuned with the CRF module enabled [173, 6].

In the next two chapters, we develop architectures that utilise the Instance CRF module [6]. Lastly, in Chapter 5, motivated by the aforementioned issues with the Instance CRF approach, we propose a new instance-level segmentation framework.

Chapter 3

Holistic, Instance-level Human Parsing

Qizhu Li*

University of Oxford

Anurag Arnab*

University of Oxford

Philip H. S. Torr

University of Oxford

Abstract

Object parsing – the task of decomposing an object into its semantic parts – has traditionally been formulated as a category-level segmentation problem. Consequently, when there are multiple objects in an image, current methods cannot count the number of objects in the scene,

* Equal contribution by the authors

nor can they determine which part belongs to which object. We address this problem by segmenting the parts of objects at an instance level, such that each pixel in the image is assigned a part label, as well as the identity of the object it belongs to. Moreover, we show how this approach benefits us in obtaining segmentations at coarser granularities as well. Our proposed network is trained end-to-end given detections, and begins with a category-level segmentation module. Thereafter, a differentiable Conditional Random Field, defined over a variable number of instances for every input image, reasons about the identity of each part by associating it with a human detection. In contrast to other approaches, our method can handle the varying number of people in each image and our holistic network produces state-of-the-art results in instance-level part and human segmentation, together with competitive results in category-level part segmentation, all achieved by a single forward-pass through our neural network.

3.1 Introduction

Object parsing, the segmentation of an object into semantic parts, is naturally performed by humans to obtain a more detailed understanding of the scene. When performed automatically by computers, it has many practical applications, such as in human-robot interaction, human behaviour analysis and image descriptions for the visually impaired. Furthermore, detailed part information has been shown to be beneficial in other visual recognition tasks such as fine-grained recognition [167], human pose estimation [37] and object detection [115]. In this paper, we focus on the application of parsing humans as it is more commonly studied, although our method makes no assumptions on the type of object it is segmenting.

(a) Input image (b) Category-level Part Segmentation (c) Instance-level Part Segmentation (d) Instance-level Human Segmentation

Figure 3.1: Our proposed approach segments human parts at an instance level (c) (which to our knowledge is the first work to do so) from category-level part segmentations produced earlier in the network (b). Moreover, we can easily obtain human instance segmentations (d) by taking the union of all pixels associated with a particular person. Therefore, our proposed end-to-end trained neural network parses humans into semantic parts at both category and instance level in a single forward pass. Best viewed in colour.

In contrast to existing human parsing approaches [94, 156, 52], we operate at an *instance level* (to our knowledge, we are the first work to do so). As shown in Fig. 3.1, not only do we segment the various body parts of humans (Fig. 3.1b), but we associate each of these parts to one of the humans in the scene (Fig. 3.1c), which is particularly important for understanding scenes with multiple people. In contrast to existing instance segmentation work [31, 96, 105], we operate at a more detailed part level, enabling us to extract more comprehensive information of the scene. Furthermore, with our part-level instance segmentation of humans, we can easily recover human-level instance segmentation (by taking the union of all parts assigned to a particular instance as shown in Fig. 3.1d), and we show significant improvement over previous state-of-the-art in human instance-segmentation when doing so.

Our approach is based on a deep Convolutional Neural Network (CNN), which consists of an initial category-level part segmentation module. Using the output of a human detector, we are then able to associate segmented parts with detected humans in the image using a differentiable Conditional Random Field (CRF), producing a part-level instance segmentation of the image. Our formulation is robust

to false-positive detections as well as imperfect bounding boxes which do not cover the entire human, in contrast to other instance segmentation methods based on object detectors [31, 105, 84, 56, 57]. Given object detections, our network is trained end-to-end, given detections, with a novel loss function which allows us to handle a variable number of human instances on every image.

We evaluate our approach on the Pascal Person-Parts [24] dataset, which contains humans in a diverse set of poses and occlusions. We achieve state-of-the-art results on instance-level segmentation of both body parts and humans. Moreover, our results on semantic part segmentation (which is not-instance aware) are also competitive with current state-of-the-art. All of these results are achieved with a holistic, end-to-end trained model which parses humans at both an instance and category level, and outputs a dynamic number of instances per image, all in a single forward-pass through the network.

3.2 Related Work

The problem of object parsing, which aims to decompose objects into their semantic parts, has been addressed by numerous works [94, 156, 150, 93, 116], most of which have concentrated on parsing humans. However, none of the aforementioned works have parsed objects at an instance level as shown in Fig. 3.1, but rather category level. In fact, a lot of work on human parsing has focussed on datasets such as Fashionista [158], ATR [93] and Deep Fashion [109] where images typically contain only one, centred person. The notion of instance-level segmentation only matters when more than one person is present in an image, motivating us to evaluate our method on the Pascal Person-Parts dataset [24] where multiple people can appear in unconstrained environments. Recent human parsing approaches have typically been similar to semantic segmentation works using fully convolutional networks

(FCNs) [110], but trained to label parts [20, 23, 21] instead of object classes. However, methods using only FCNs do not explicitly model the structure of a human body, and typically do not perform as well as methods which do [94]. Structural priors of the human body have been encoded using pictorial structures [41, 42], Conditional Random Fields (CRFs) [14, 80, 150, 69] and more recently, with LSTMs [95, 94]. The HAZN approach of [156] addressed the problem that some parts are often very small compared to other parts and difficult to segment with scale-variant CNNs. This scale variation was handled by a cascade of three separately-trained FCNs, each parsing different regions of the image at different scales.

An early instance segmentation work by Winn *et al*. [154] predicted the parts of an object, and then encouraged these parts to maintain a spatial ordering, characteristic of an instance, using asymmetric pairwise potentials in a CRF. However, subsequent work has not operated at a part level. Zhang *et al*. [169, 168] performed instance segmentation of vehicles using an MRF. However, this graphical model was not trained end-to-end as done by [172, 5, 99] and our approach. Furthermore, they assumed a maximum of 9 cars per image. Approaches using recurrent neural networks [129, 126] can handle a variable number of instances per image by segmenting an instance per time-step, but are currently restricted to only one object category. Our method, on the other hand, is able to handle both an arbitrary number of objects, and multiple object categories in the image with a single forward-pass through the network.

Various methods of instance segmentation have also involved modifying object detection systems to output segments instead of bounding boxes [56, 57, 31, 84]. However, these methods cannot produce a segmentation map of the image, as shown in Fig. 3.1, without post-processing as they consider each detection independently. Although our method also uses an object detector, it considers all detections in the image jointly with an initial category-level segmentation, and

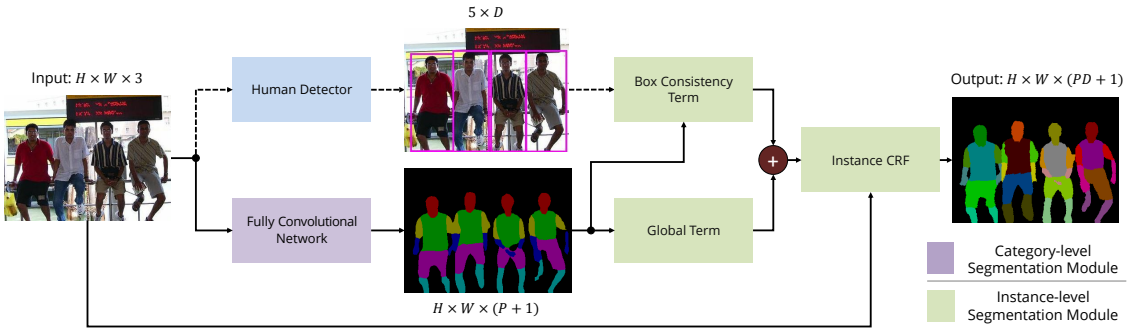


Figure 3.2: Our proposed approach. An $H \times W \times 3$ image is input to a human detection network and a body parts semantic segmentation network, producing D detections of human and an $H \times W \times (P+1)$ dimensional feature map respectively, where $(P + 1)$ is the size of the semantic label space including a background class. These results are used to form the unary potentials of an Instance CRF which performs instance segmentation by associating labelled pixels with human detections. In the above diagram, dotted lines represent forward only paths, and solid lines show routes where both features and gradients flow. The green boxes form the instance-level segmentation module (Sec. 3.3.2). Best viewed in colour.

produces segmentation maps naturally where one pixel cannot belong to multiple instances in contrast to the aforementioned approaches. The idea of combining the outputs of a category-level segmentation network and an object detector to reason about different instances was also presented by [7]. However, that system was not trained end-to-end, could not segment instances outside the detector's bounding box, and did not operate at a part level.

3.3 Proposed Approach

Our network (Fig. 3.2) consists of two components: a category-level part segmentation module, and an instance segmentation module. As both of these modules are differentiable, they can be integrated into a single network and trained jointly. The instance segmentation module (Sec. 3.3.2) uses the output of the first category-level segmentation module (Sec. 3.3.1) as well as the outputs of an object detector as its input. It associates each pixel in the category-level segmentation with an

object detection, resulting in an instance-level segmentation of the image. Given an $H \times W \times 3$ input image, $\mathbf{I}$, the category-level part segmentation module produces an $H \times W \times (P+1)$ dimensional output $\mathbf{Q}$ where P is the number of part classes in the dataset and one background class. There can be a variable number, D, of human detections per image, and the output of the instance segmentation module is an $H \times W \times (PD+1)$ tensor denoting the probabilities, at each pixel in the image, of each of the P part classes belonging to one of the D detections.

Two challenges of instance segmentation are the variable number of instances in every image, and the fact that permutations of instance labels lead to identical results (in Fig. 3.1, how we order the different people does not matter). Zhang *et al*. [169, 168] resolve these issues by assuming a maximum number of instances and using the ground truth depth ordering of instances respectively. Others have bypassed both of these issues by predicting each instance independently [31, 56, 57, 84], but this also allows a pixel to belong to multiple instances. Instead, we use a loss function (Sec. 3.3.3) that is based on "matching" the prediction to the ground truth, allowing us to handle permutations of the ground truth. Furthermore, weight-sharing in our instance segmentation module allows us to segment a variable number of instances per image. As a result, we do not assume a maximum number of instances, consider all instances jointly, and train our network end-to-end, given object detections.

3.3.1 Category-level part segmentation module

The part segmentation module is a fully convolutional network [110] based on ResNet-101 [60]. A common technique, presented in [23, 21], is to predict the image at three different scales (with network weights shared among all the scales), and combine predictions together with learned, image-dependent weights. We take a different approach of fusing information at multiple scales – we pool the

features after `res5c` [60] at five different resolutions (by varying the pooling stride), upsample the features to the resolution before pooling, and then concatenate these features before passing them to the final convolutional classifier, as proposed in [170]. As we show in Sec. 3.4.4, this approach achieves better semantic segmentation results than [21, 23]. We denote the output of this module by the tensor, $\mathbf{Q}$, where $Q_i(l)$ is the probability of pixel i being assigned label $l \in \{0, 1, 2, ..., P\}$. Further details of this module are included in Appendix 3.B.1.

3.3.2 Instance-level segmentation module

This module creates an instance-level segmentation of the image by associating each pixel in the input category-level segmentation, $\mathbf{Q}$, with one of the D input human-detections or the background label. Let there be D input human-detections for the image, where the i-th detection is represented by B_i, the set of pixels lying within the four corners of its bounding box, and $s_i \in [0, 1]$, the detection score. We assume that the 0-th detection refers to the background label. Furthermore, we define a multinomial random variable, V_i, at each of the N pixels in the image, and let $\mathbf{V} = [V_1, V_2, ..., V_N]^\top$. This variable can take on a label from the set $\{1, 2, ..., D\} \times \{1, 2, ..., P\} \cup \{(0, 0)\}$ since each of the P part labels can be associated with one of the D human detections, or that pixel could belong to the background label, $(0, 0)$.

We formulate a Conditional Random Field over these V variables, where the energy of the assignment $\mathbf{v}$ to all of the instance variables $\mathbf{V}$ consists of two unary terms, and one pairwise term (whose weighting coefficients are all learned via backpropagation):

$$E(\mathbf{V} = \mathbf{v}) = -\sum_{i}^{N} \ln \left[w_1 \psi_{Box}(v_i) + w_2 \psi_{Global}(v_i) + \epsilon \right] + \sum_{i<j}^{N} \psi_{Pairwise}(v_i, v_j). \quad (3.1)$$

The unary and pairwise potentials are computed within our neural network, dif-

ferentiable with respect to their input and parameters, and described in Sec. 3.3.2.1 through 3.3.2.3. The Maximum-a-Posteriori (MAP) estimate of our CRF (since the energy in Eq. 3.1 characterises a Gibbs distribution) is computed as the final labelling produced by our network. We perform the iterative mean-field inference algorithm to approximately compute the MAP solution by minimising Eq. 3.1. As shown by Zheng *et al*. [172], this can be formulated as a Recurrent Neural Network (RNN), allowing it to be trained end-to-end as part of a larger network. However, as our network is input a variable number of detections per image, D, the label space of the CRF is dynamic. Therefore, unlike [172], the parameters of our CRF are not class-specific to allow for this variable number of "channels".

3.3.2.1 Box Consistency Term

We observe that in most cases, a body part belonging to a person is located inside the bounding box of the person. Based on this observation, the box consistency term is employed to encourage pixel locations inside a human bounding box B_i to be associated with the i-th human detection. The box term potential at spatial location k for body part j of a human i is assigned either 0 for $k \notin B_i$, or the product of the detection score, s_i, and the category-level part segmentation confidence, $Q_k(j)$, for $k \in B_i$. For $(i, j) \in \{1, 2, \ldots, D\} \times \{1, 2, \ldots, P\}$,

$$\psi_{Box}(V_k = (i, j)) = \begin{cases} s_i Q_k(j) & \text{if } k \in B_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Note that this potential may be robust to false-positive detections when the category-level segmentation and human detection do not agree with each other, since $Q_k(l)$, the probability of a pixel k taking on body-part label l, is low. Furthermore, note that we use one human-detection to reason about the identity of all parts which

constitute that human.

3.3.2.2 Global Term

A possible shortcoming for the box consistency potential is that if some pixels belonging to a human instance fall outside the bounding box and are consequently assigned 0 for the box consistency term potential, they would be lost in the final instance segmentation prediction. Visually, the generated instance masks would appear truncated along the bounding box boundaries – a problem suffered by [7, 31, 84, 57]. To overcome this undesirable effect, we introduce the global potential: it complements the box consistency term by assuming that a pixel is equally likely to belong to any one of the detected humans. It is expressed as

$$\psi_{Global}(V_k = (i, j)) = Q_k(j), \tag{3.3}$$

for $(i, j) \in \{1, 2, ..., D\} \times \{1, 2, ..., P\} \cup \{(0, 0)\}$.

3.3.2.3 Pairwise Term

Our pairwise term is composed of densely connected Gaussian kernels [76] which are commonly used in segmentation literature [20, 172]. This pairwise potential encourages both spatial and appearance consistency, and we find these priors to be suitable in the case of instance-level segmentation as well. As in [172], the weighting parameters of these potentials are learned via backpropagation, though in our case, the weights are shared among all classes.

3.3.3 Loss function and network training

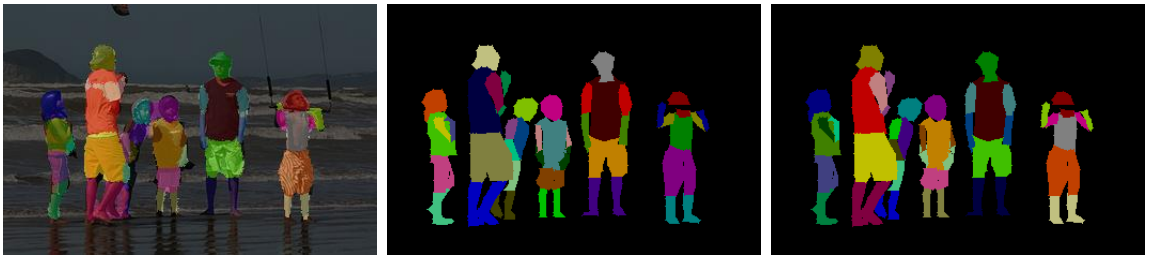
We first pre-train the category-level segmentation part of our network, as described in Appendix 3.B.2. Thereafter, we add the instance segmentation module, and train

with a permutation-invariant loss function which is backpropagated through both our instance- and category-level segmentation networks. Since all permutations of an instance segmentation have the same qualitative result, we "match" the original ground truth to our prediction before computing the loss, as shown in Fig. 3.3. This matching is based on the Intersection over Union (IoU) [39] of a predicted and ground truth instance, similar to [129]. Let $\mathcal{Y} = \{y_1, y_2, ..., y_m\}$, a set of m segments, denote the ground truth labelling of an image, where each segment is an instance and has a part label assigned to it. Similarly, let $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ denote our n predicted instances, each with an associated part label. Note that m and n need not be the same as we may predict greater or fewer instances than there actually are in the image. The "matched" ground truth, $\mathcal{Y}^*$ is the permutation of the original ground truth labelling which maximises the IoU between our prediction, $\mathcal{P}$ and ground truth

$$\mathcal{Y}^* = \underset{\mathcal{Z} \in \pi(\mathcal{Y})}{\arg\max}\, \text{IoU}(\mathcal{Z}, \mathcal{P}), \tag{3.4}$$

where $\pi(\mathcal{Y})$ denotes the set of all permutations of $\mathcal{Y}$. Note that we define the IoU between all segments of different labels to be 0. Eq. 3.4 can be solved efficiently using the Hungarian algorithm as it can be formulated as a bipartite graph matching problem, and once we have the "matched" ground truth, $\mathcal{Y}^*$, we can apply any loss function to it and train our network for segmentation.

In our case, we use the standard cross-entropy loss function on the "matched" ground truth. In addition, we employ Online Hard Example Mining (OHEM), and only compute our loss over the top K pixels with the highest loss in the training mini-batch. We found that during training, many pixels already had a high probability of being assigned to the correct class. By only selecting the top K pixels with the highest loss, we are able to encourage our network to improve on the pixels it is currently misclassifying, as opposed to increasing the probability of a pixel it is already classifying correctly. This approach was inspired by "bootstrap-



Prediction, $\mathcal{P}$ | Original ground truth, $\mathcal{Y}$ | "Matched" ground truth, $\mathcal{Y}^*$

Figure 3.3: As different permutations of the ground truth are equivalent in the case of instance segmentation, we "match" the original ground truth, $\mathcal{Y}$, to our network's prediction, $\mathcal{P}$, to obtain the "matched" ground truth which we use to compute our loss during training.

ping" [33, 142] or "hard-negative mining" [40] commonly used in training object detectors. However, these methods mined hard examples from the entire dataset. Our approach is most similar to [138], who mined hard examples online from each mini-batch in the context of detection. Similar to the aforementioned works, we found OHEM to improve our overall results, as shown in Sec. 3.4.2.

3.3.4 Obtaining segmentations at other granularities

Given the part instance prediction produced by our proposed network, we are able to easily obtain human instance segmentation and semantic part segmentation. In order to achieve human instance segmentation, we map the predicted part instance labels (i, j), i.e. part j of person i, to i. Whereas to obtain semantic part segmentation, we map predicted part instance labels (i, j) to j instead.

3.4 Experiments

We describe our dataset and experimental set-up in Sec. 3.4.1, before presenting results on instance-level part segmentation (Fig. 3.1c), instance-level human segmentation (Fig. 3.1d) and semantic part segmentation (Fig. 3.1b). Additional quantitative and qualitative results, failure cases and experimental details are included

in Appendix 3.A.

3.4.1 Experimental set-up

We evaluate our proposed method on the Pascal Person-Part dataset [37] which contains 1716 training images, and 1817 test images. This dataset contains multiple people per image in unconstrained poses and environments, and contains six human body part classes (Fig. 3.1b), as well as the background label. As described in Sec. 3.3.3, we initially pre-train our category-level segmentation module before training for instance-level segmentation. This module is first trained on the 21 classes of the Pascal VOC dataset [39], and then finetuned on the seven classes of the Pascal Part training set using category-level annotations. Finally, we train for instance segmentation with instance-level ground truth. Full details of our training process, including all hyperparameters such as learning rate, are in Appendix 3.B.2.

We use the standard AP^r metric [56] for evaluating instance-level segmentation: the mean Average Precision of our predictions is computed where a prediction is considered correct if its IoU with a ground truth instance is above a certain threshold. This is similar to the AP metric used in object detection. However, in detection, the IoU between ground truth and predicted bounding boxes is computed, whereas here, the IoU between regions is computed. Furthermore, in detection, an overlap threshold of 0.5 is used, whereas we vary this threshold. Finally, we define the AP^r_{vol} which is the mean of the AP^r score for overlap thresholds varying from 0.1 to 0.9 in increments of 0.1.

We use the publicly available R-FCN detection framework [32], and train a new model with data from VOC 2012 [39] that do not overlap with any of our test sets. We train with all object classes of VOC, and only use the output for the human class. Non-maximal suppression is performed on all detections before being fed into our network.

Table 3.1: Comparison of AP^r at various IoU thresholds for instance-level part segmentation on the Pascal Person-Parts dataset.

Method	IoU threshold 0.5	0.6	0.7	AP^r_{vol}
MNC [31]	38.8	28.1	**19.3**	36.7
Ours, piecewise trained, box term only*	38.0	27.4	16.7	36.6
Ours, piecewise trained	38.8	28.5	17.6	37.3
Ours, end-to-end trained	39.0	28.6	17.4	37.7
Ours, piecewise trained, box term only, OHEM	38.7	28.9	17.5	36.7
Ours, piecewise trained, OHEM	39.7	29.7	18.7	37.4
Ours, end-to-end trained, OHEM	**40.6**	**30.4**	19.1	**38.4**

*Model is equivalent to our re-implementation of [7]

3.4.2 Results on instance-level part segmentation

Table 3.1 shows our results on part-level instance segmentation on the Pascal Person-Part dataset. To our knowledge, we are the first work to do this, and hence we study the effects of various design choices on overall performance. We also use the publicly available code for MNC [31], which won the MS-COCO 2016 instance segmentation challenge, and finetune their public model trained on VOC 2011 [55] on Person-Part instances as a baseline.

We first train our model in a piecewise manner, by first optimising the parameters of the category-level segmentation module, and then "freezing" the weights of this module and only training the instance network. Initially, we only use the box consistency term (Sec. 3.3.2.1) in the Instance CRF, resulting in an AP^r at 0.5 of 38.0%. Note that this model is equivalent to our re-implementation of [7]. Adding in the global potential (Sec. 3.3.2.2) helps us cope with bounding boxes which do not cover the whole human, and we see an improvement at all IoU thresholds. Training our entire network end-to-end gives further benefits. We then train all variants of our model with OHEM, and observe consistent improvements across all IoU thresholds with respect to the corresponding baseline. Here, we set $K = 2^{15}$,

meaning that we computed our loss over 2^{15} or approximately 12% of the hardest pixels in each training image (since we train at full resolution). We also employ OHEM when pre-training the category-level segmentation module of our network, and observe minimal difference in the final result if we use OHEM when training the category-level segmentation module but not the instance segmentation module. Training end-to-end with OHEM achieves 2.6% higher in AP^r at 0.5, and 1.8% higher AP^r_{vol} over a piecewise-trained baseline model without OHEM and only the box term (second row), which is equivalent to the model of [7]. Furthermore, our AP^r_{vol} is 1.7% greater than the strong MNC [31] baseline. Note that although [57] also performed instance-level segmentation on the same dataset, their evaluation was only done using human instance labels, which is similar to our following experiment on human instance segmentation.

3.4.3 Results on human instance segmentation

We can trivially obtain instance-level segmentations of humans (Fig. 3.1d), as mentioned in Sec. 3.3.4. Table 3.2 shows our state-of-the-art instance segmentation results for humans on the VOC 2012 validation set [39]. We use the best model from the previous section as there is no overlap between the Pascal Person-Part training set, and the VOC 2012 validation set.

As Tab. 3.2 shows, our proposed approach outperforms previous state-of-the-art by a significant margin, particularly at high IoU thresholds. Our model receives extra supervision in its part labels, but the fact that our network can implicitly infer relationships between different parts whilst training may help it handle occluding instances better than other approaches, leading to better instance segmentation performance. The fact that our network is trained with part-level annotations may also help it identify small features of humans better, leading to more precise segmentations and thus improvements at high AP^r thresholds. Our AP^r at each

Table 3.2: Comparison of AP^r at various IoU thresholds for instance-level human segmentation on the VOC 2012 validation set.

Method	IoU threshold 0.5	0.6	0.7	0.8	0.9	AP^r_{vol}
SDS [56]	47.9	31.8	15.7	3.3	0.1	–
Chen *et al*. [25]	48.3	35.6	22.6	6.5	0.6	–
PFN [92]	48.4	38.0	26.5	16.5	5.9	41.3
Arnab *et al*. [7]*	58.6	52.6	41.1	30.4	10.7	51.8
R2-IOS [96]	60.4	51.2	33.2	–	–	–
Arnab *et al*. [6]*	65.6	58.0	46.7	33.0	14.6	57.4
Ours, piecewise	64.0	59.8	51.0	38.3	**20.1**	57.2
Ours, end-to-end	**70.2**	**63.1**	**54.1**	**41.0**	19.6	**61.0**

*Results obtained from supplementary material.

Table 3.3: Comparison of semantic part segmentation results on the Pascal Person-Parts test set.

Method	IoU [%]
DeepLab* [20]	53.0
Attention [23]	56.4
HAZN [156]	57.5
LG-LSTM [95]	58.0
Graph LSTM [94]	60.2
DeepLab v2 [21]	64.9
RefineNet [98]	68.6
Ours, pre-trained	65.9
Ours, final network	66.3

*Result reported in [156]

IoU threshold for human instance segmentation is higher than that for part instance segmentation (Tab. 3.1). This is because parts are smaller than entire humans, and thus more difficult to localise accurately. An alternate method of performing instance-level part segmentation may be to first obtain an instance-level human segmentation using another method from Tab. 3.2, and then partition it into the various body parts of a human. However, our approach, which groups parts into instances, is validated by the fact that it achieves state-of-the-art instance-level human segmentation performance.

3.4.4 Results on category-level part segmentation

Finally, our model is also able to produce category-level segmentations (as shown in Fig. 3.1b). This can be obtained from the output of the category-level segmentation module, or from our instance module as described in Sec. 3.3.4. As shown in Tab. 3.3, our semantic segmentation results are competitive with current state-of-the-art. By training our entire network consisting of the category-level and instance-

level segmentation modules jointly, and then obtaining the semantic segmentation from the final instance segmentation output by our network, we are able to obtain a small improvement of 0.4% in mean IoU over the output of the initial semantic segmentation module.

3.5 Conclusion

Our proposed, end-to-end trained network outputs instance-level body part and human segmentations, as well as category-level part segmentations in a single forward-pass. Moreover, we have shown how segmenting objects into their constituent parts helps us segment the object as a whole with our state-of-the-art results on instance-level segmentation of both body parts and entire humans. Furthermore, our category-level segmentations improve after training for instance-level segmentation. Our future work is to train the object detector end-to-end as well. Moreover, the improvement that we obtained in instance segmentation of humans as a result of first segmenting parts motivates us to explore weakly-supervised methods which do not require explicit object part annotations.

Acknowledgement We thank Stuart Golodetz for discussions and feedback. This work was supported by the EPSRC, Clarendon Fund, ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1.

Appendices

We present additional results of our proposed approach in Sec. 3.A, and provide additional training and implementation details in Sec. 3.B (both for our model, and the strong MNC baseline [31]).

3.A Additional Results

In the main sections, we reported our AP^r results averaged over all classes. Fig. 3.5 visualises the per-class results of our best model at different IoU thresholds. Fig. 3.6 displays the success cases of our method, while Fig. 3.7 shows examples of failure cases. Furthermore, we illustrate the strengths and weaknesses of our part instance segmentation method in comparison to MNC [31] in Fig. 3.8, and compare our instance-level human segmentation results, which we obtain by the simple mapping described in Sec. 3.3.4, to MNC in Fig. 3.9.

Finally, we attach an additional video[1]. We run our system offline, on a frame-by-frame basis on the entire music video, and show how our method is able to accurately parse humans at both category and instance level on internet data outside the Pascal dataset. Instance-level segmentation of videos requires data association. We use a simple, greedy method which operates on a frame-by-frame basis. Segments from one frame are associated to segments in the next frame based on the IoU, using the same method we use for our loss function as described in Sec. 3.3.3.

3.B Additional Information

We detail our initial category-level segmentation module and compare it to DeepLab-v2 [21] in Sec. 3.B.1, present our network training details in Sec. 3.B.2, and finally

[1]It can be viewed at `https://www.youtube.com/watch?v=XYb3_GnAgQo`

describe how we train the MNC model which serves as our baseline in Sec. 3.B.3.

3.B.1 Details of the category-level segmentation module

As shown in Fig 3.10b, the structure of our category-level segmentation module consists of a ResNet-101 backbone, and a classifier that extracts multi-scale features from the ResNet-101 output by using average pooling with different kernel sizes. While our category-level segmentation module and the Deeplab-v2 network (Fig. 3.10a) of Chen *et al*. [21] both attempt to exploit multi-scale information in the image, the approach of [21] entails executing three forward passes for each image, whereas we only need a single forward pass.

In the Deeplab-v2 architecture, a 513×513×3 input image is downsampled by two different ratios (0.75 and 0.5) to produce multi-scale input at three different resolutions. The three resolutions are independently processed by a ResNet-101-based network using shared weights (shown by the individually coloured paths in Fig. 3.10a). The output feature maps are then upsampled where appropriate, combined by taking the elementwise maximum, and finally upsampled back to 513×513.

In contrast, in this work, the category-level segmentation module forwards an input image of size 521×521×3 through a ResNet-101-based CNN, producing a feature map of resolution 66×66×2048 (Fig. 3.10b). This feature map is average-pooled with four different kernel sizes, giving us four feature maps with spatial resolutions 1×1, 2×2, 3×3, and 6×6 respectively. Each feature map undergoes convolution and upsampling, before being concatenated together with each other and the 66×66×2048 ResNet-101 output. This is followed by a convolution layer that reduces the dimension of the concatenated features to 512, and a convolutional classifier that maps the 512 channels to the size of label space in the dataset. Finally, the prediction is upsampled back to 521×521.

In comparison to Deeplab-v2, our network saves both memory and time, and achieves better performance. To carry out a single forward pass, our network uses 4.3GB of memory while Deeplab-v2 [21] needs 9.5GB, 120% more than ours. Speed-wise, our network runs forward passes at 0.255 seconds per image (3.9 fps), whereas Deeplab-v2 takes 55% longer, at 0.396 seconds per image (2.5 fps) on average. When Deeplab-v2 adds a CRF with 10 mean-field iterations to post-process the network output, it gains a small improvement in mean IoU by 0.54% [21], but it requires 11.2GB of memory to make a forward pass (140% of the total amount used by our full network including the instance-level segmentation module), and takes 0.960 seconds per image (1.0 fps), almost a quarter of our frame rate. Tests are done on a single GeForce GTX Titan X (Maxwell) card. Overall, we are able to achieve better segmentation accuracy (as shown in Tab. 3.3) and is more memory- and time-efficient than Deeplab-v2.

3.B.2 Training our proposed network

3.B.2.1 Training the category-level segmentation module

We initialise our semantic segmentation network with the COCO pre-trained ResNet-101 weights provided by [21]. Training is first performed on the Pascal VOC 2012 training set using the extra annotations from [55], which combine to a total of 9012 training images. Care is taken to ensure that all images from the Pascal Person-Parts test set are excluded from this training set. A polynomial learning rate policy is adopted such that the effective learning rate at iteration i is given by $l_i = l_0(1 - \frac{i}{i_{max}})^p$, where the base learning rate, l_0, is set to 6.25×10^{-4}, the total number of iterations, i_{max}, is set to 30k, and the power, p, is set to 0.9. A batch size of 16 is used. However, due to memory constraints, we simulate this batch size by "accumulating gradients": We carry out 16 forward and backward passes with one image per iteration, and only perform the weight update after completing

all 16 passes. We use a momentum of 0.9 and weight decay of 1×10^{-4} for these experiments. After 30k of iterations are completed, we take the best performing model and finetune on the Pascal Person-Parts training set using the same training scheme as described above. Note that the parameters of the batch normalisation modules are kept unchanged in the whole learning process.

Online data-augmentation is performed during training to regularise the model. The training images are randomly mirrored, scaled by a ratio between 0.5 and 2, rotated by an angle between -10 and 10 degrees, translated by a random amount in the HSV colour space, and blurred with a randomly-sized Gaussian kernel, all on-the-fly. We observe that these techniques are effective at reducing the accuracy gap between training and testing, leading to overall higher test accuracies.

3.B.2.2 Training the instance-level segmentation module

In our model, the pairwise term of the fully connected CRF takes the following form:

$$\psi_{Pairwise}(v_i, v_j) = \mu(v_i, v_j) k(\mathbf{f}_i, \mathbf{f}_j) \tag{3.5}$$

where $\mu(\cdot, \cdot)$ is a compatibility function, $k(\cdot, \cdot)$ is a kernel function, and $\mathbf{f}_i$ is a feature vector at spatial location i containing the 3-dimensional colour vector I_i and the 2-dimensional position vector p_i [76]. We further define the kernel as follows:

$$k(\mathbf{f}_i, \mathbf{f}_j) = w^{(1)} \exp\Big(- \frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2} \Big) + w^{(2)} \exp\Big(- \frac{|p_i - p_j|^2}{2\theta_\gamma^2} \Big) \tag{3.6}$$

where $w^{(1)}$ and $w^{(2)}$ are the linear combination weights for the bilateral term and the Gaussian term respectively. In order to determine the initial values for the parameters in the Instance CRF to train from, we carry out a random search. According to the search results, the best prediction accuracy is obtained by initialising $w^{(1)} = 8$, $w^{(2)} = 2$, $\theta_\alpha = 2$, $\theta_\beta = 8$, $\theta_\gamma = 2$. Furthermore, we use a fixed learning

rate of 1×10^{-6}, momentum of 0.9, and weight decay of 1×10^{-4} for training both the instance-level and category-level segmentation modules jointly. Although we previously use the polynomial learning rate policy, we find that for training the instance-level segmentation module, a fixed learning rate leads to better results. Furthermore, our experiments show that a batch size of one works best at this training stage. Using this scheme, we train for 175k iterations, or approximately 100 epochs.

3.B.3 Training Multi-task Network Cascades (MNC)

We use the publicly available Multi-task Network Cascades (MNC) framework [31], and train a new model for instance-level part segmentation using the Pascal Person-Parts dataset. The weights are initialised with the officially released MNC model[2] which has been trained on Pascal VOC 2011/SBD [55]. The base learning rate is set to 1×10^{-3}, which is reduced by 10 times after 20k iterations. A total of 25k training iterations are carried out. A batch size of 8, momentum of 0.9 and weight decay of 5×10^{-4} are used. These settings are identical to the ones used in training the original MNC and provided in their public source code. Using these settings, we are also able to reproduce the experimental results obtained in the original MNC paper [31], and hence we believe that the MNC model we have trained acts as a strong baseline for our proposed approach.

[2] https://github.com/daijifeng001/MNC

Figure 3.4: Some results of our system. The first column shows the input image and the input detections we obtained from training the R-FCN detector [32]. The second and third columns show our final semantic segmentation (Sec. 3.3.4) and instance-level part segmentation. *First row*: our network can deal with poor bounding box localisation, as it manages to segment the third person from the left although the bounding box only partially covers her. *Second row*: our method is robust against false positive detections because of the box term. Observe that the bowl of the rightmost person in the bottom row is falsely detected as a person, but rejected in the final prediction. *Following rows*: we are able to handle overlapping bounding boxes by reasoning globally using the Instance CRF.

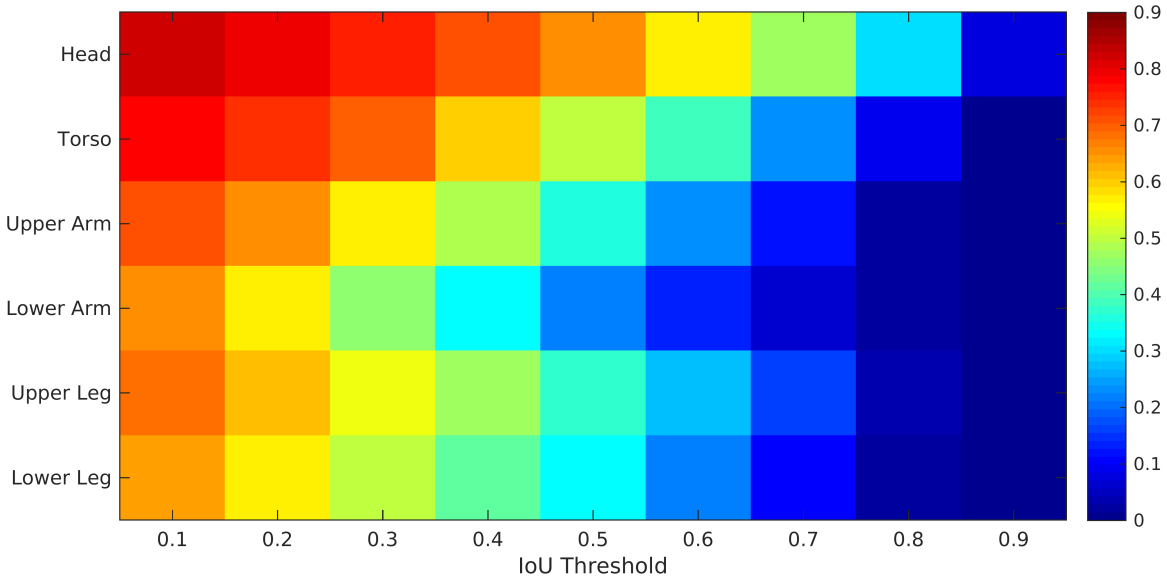


Figure 3.5: Visualisation of per-class results for different IoU thresholds on the Pascal Person-Parts test set. The heatmap shows the per-class AP^r of our best model at IoU thresholds from 0.1 to 0.9 in increments of 0.1 on the Pascal Person-Parts test set. It shows that our method achieves best instance accuracy for the head category, and finds lower arms and lower legs most challenging to segment correctly. This is likely because of the thin shape of the lower limbs which is known to pose difficulty for semantic segmentation.

Figure 3.6: Success cases of our method. The first column shows the input image and the input detections we obtained from training the R-FCN detector [32]. The second column shows our final semantic segmentation (as described in Sec. 3.3.4). Our proposed method is able to leverage an initial category-level segmentation network and human detections to produce accurate instance-level part segmentation as shown in the third column.

Figure 3.7: Failure cases of our method. *First two rows*: a missing human detection confuses the instance-level segmentation module. *Third and fourth row*: overlapping detection bounding boxes lead to incorrect instance label assignment when the overlapping region are visually similar. *Fifth row*: although our method is robust against false positive detections, two small regions on the leftmost person's left arm and left knee are assigned to the false positive detection.

Figure 3.8: Comparison to MNC on the Pascal Person-Parts [24] test set. *First row*: unlike MNC which predicts for each part instance independently, our method reasons globally and jointly. As a result, MNC predicts two instances of lower legs for the same lower leg of the second and third person from the left. Furthermore, with a dedicated category-level segmentation module, we are less prone to false negatives, whereas MNC misses the legs of the rightmost person, and the lower arm of the second person from the right. *Second row*: while we can handle poor bounding box localisation because of our global potential term, MNC is unable to segment regions outside the bounding boxes it generates. Consequently, only one lower arm of the person on the left is segmented as the other one is outside the bounding box. The square corners of the segmented lower arm correspond to the limits imposed by the bounding box which MNC internally uses. *Third row*: By analysing an image globally and employing a differentiable CRF, our method can produce more precise boundaries. As MNC does not perform category-level segmentation over the entire image, it has no incentive to produce a coherent and continuous prediction. Visually, this is reflected in the gaps of "background" between body parts of the same person. *Fourth row*: MNC predicts two instances of the lower leg for the second person from the right, and fails to segment any lower arms for all four people due to the aforementioned problems.

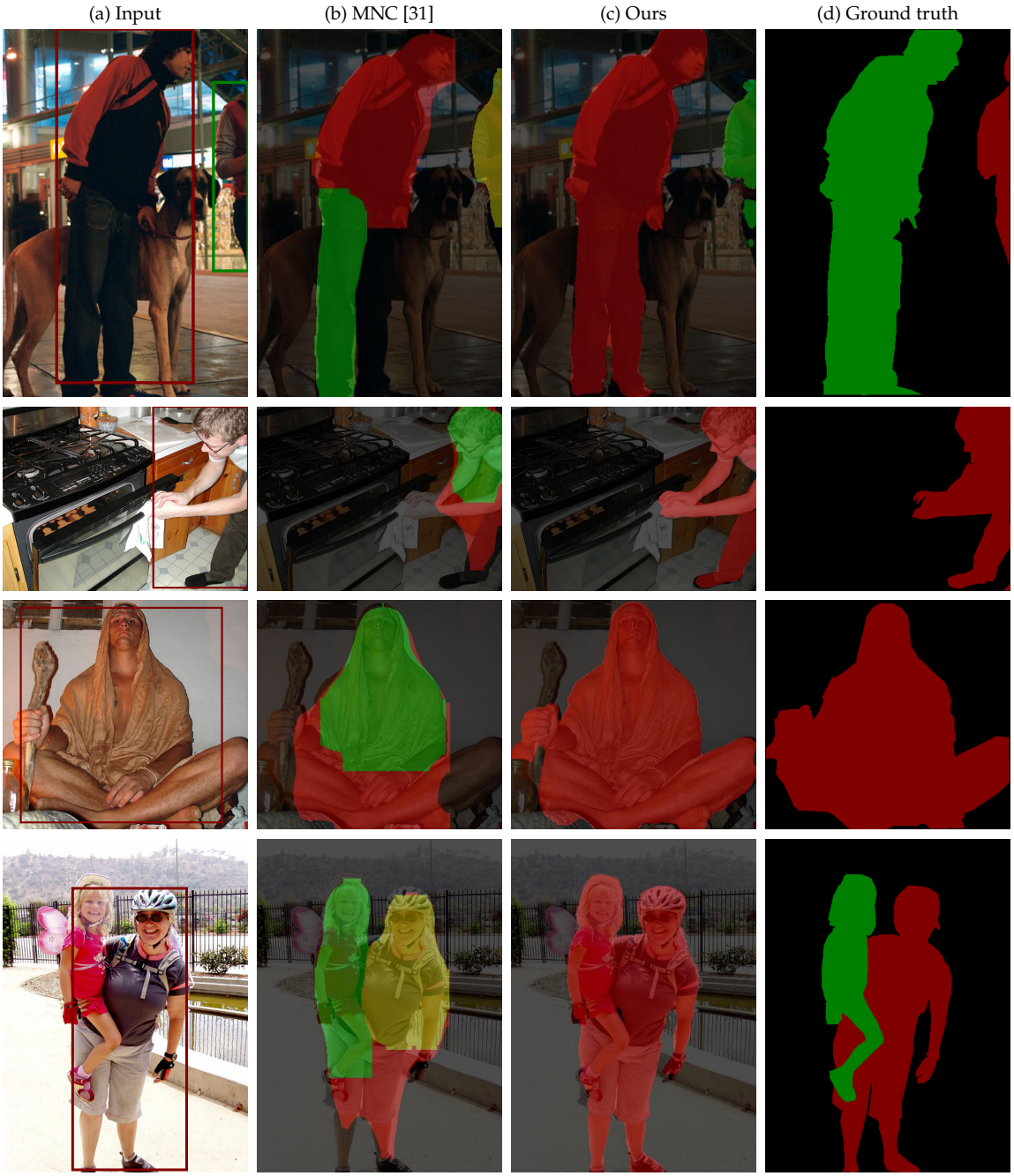
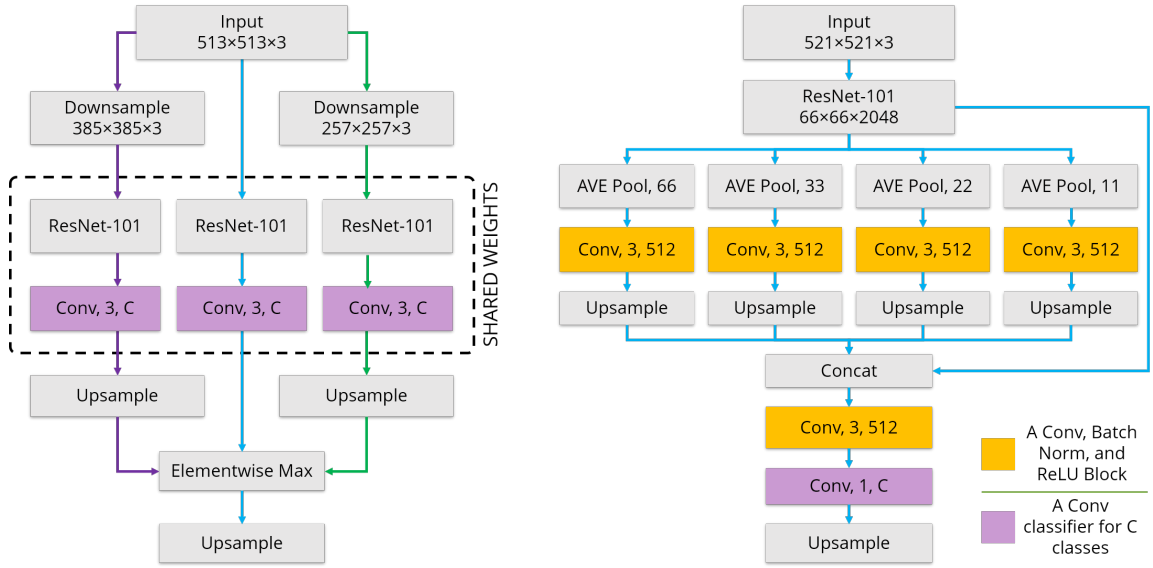


Figure 3.9: Comparison to MNC on the Pascal Person-Parts [24] test set for instance-level human segmentation. To generate the results in the second column, we run the public MNC model trained on VOC 2011/SBD [55] and extract only its human instance predictions. *First and second row*: since MNC predicts instances independently, it is prone to predicting multiple instances for a single person. *Third row*: due to the global potential term, we can segment regions outside of a non-ideal detection bounding box, whereas MNC is unable to recover from such imperfect bounding boxes.*Fourth row*: a case where MNC and our method show different failure modes. MNC predicts three people where there are only two, and our method can only predict one instance due to a missing detection.

Figure 3.9: *continued*. Comparison to MNC on the Pascal Person-Parts [24] test set for instance-level human segmentation. *First and second row*: MNC is unable to recover from a false positive detection and predicts two people. *Third row*: MNC performs better in this case as it is able to segment the infant, whereas we miss her completely due to a false negative person detection.



(a) Deeplab-v2 (b) Ours

Figure 3.10: Comparison between the Deeplab-v2 network structure which achieves 64.9% IoU on the Pascal Person-Parts dataset [21] and our network structure. The numbers following the layer type denote the kernel size and number of filters. For pooling layers, only their kernel sizes are shown. The upsampling ratios can be inferred and are therefore omitted for clarity. In both designs, the ResNet-101 backbone uses dilated convolution such that its output at `res5c` is at $1/8$ of the input resolution. The convolutional classifiers (coloured in purple) output C channels, corresponding to the number of classes in the dataset including a background class. For the Pascal Person-Parts Dataset, C is 7. Best viewed in colour.

Chapter 4

Weakly- and Semi-Supervised Panoptic Segmentation

Qizhu Li*

University of Oxford

Anurag Arnab*

University of Oxford

Philip H. S. Torr

University of Oxford

Abstract

We present a weakly supervised model that jointly performs both semantic- and instance-segmentation – a particularly relevant problem given the substantial cost of obtaining pixel-perfect annotation for these tasks. In contrast to many popular instance segmentation approaches based

* Equal contribution by the authors

on object detectors, our method does not predict any overlapping instances. Moreover, we are able to segment both "thing" and "stuff" classes, and thus explain all the pixels in the image. "Thing" classes are weakly-supervised with bounding boxes, and "stuff" with image-level tags. We obtain state-of-the-art results on Pascal VOC, for both full and weak supervision (which achieves about 95% of fully-supervised performance). Furthermore, we present the first weakly-supervised results on Cityscapes for both semantic- and instance-segmentation. Finally, we use our weakly-supervised framework to analyse the relationship between annotation quality and predictive performance, which is of interest to dataset creators.

4.1 Introduction

Convolutional Neural Networks (CNNs) excel at a wide array of image recognition tasks [60, 139, 127]. However, their ability to learn effective representations of images requires large amounts of labelled training data [132, 141]. Annotating training data is a particular bottleneck in the case of segmentation, where labelling each pixel in the image by hand is particularly time-consuming. This is illustrated by the Cityscapes dataset where finely annotating a single image took "more than 1.5h on average" [29]. In this paper, we address the problems of semantic- and instance-segmentation using only weak annotations in the form of bounding boxes and image-level tags. Bounding boxes take only 7 seconds to draw using the labelling method of [120], and image-level tags an average of 1 second per class [119]. Using only these weak annotations would correspond to a reduction factor of 30 in labelling a Cityscapes image which emphasises the importance of cost-effective, weak annotation strategies.

Our work differs from the prior art on weakly-supervised segmentation [75, 152, 121, 30, 11] in two primary ways: Firstly, our model jointly produces semantic- and instance-segmentations of the image, whereas the aforementioned works only output instance-agnostic semantic segmentations. Secondly, we consider the segmentation of both "thing" and "stuff" classes [44, 3], in contrast to most existing work in both semantic- and instance-segmentation which only consider "things".

We define the problem of instance segmentation as labelling every pixel in an image with both its object class and an instance identifier [6, 7, 169]. It is thus an extension of semantic segmentation, which only assigns each pixel an object class label. "Thing" classes (such as "person" and "car") are countable and are also studied extensively in object detection [39, 101]. This is because their finite extent makes it possible to annotate tight, well-defined bounding boxes around them. "Stuff" classes (such as "sky" and "vegetation"), on the other hand, are amorphous regions of homogeneous or repetitive textures [44]. As these classes have ambiguous boundaries and no well-defined shape they are not appropriate to annotate with bounding boxes [97]. Since "stuff" classes are not countable, we assume that all pixels of a stuff category belong to the same, single instance. Recently, this task of jointly segmenting "things" and "stuff" at an instance level has also been named "Panoptic Segmentation" by [71].

Note that many popular instance segmentation algorithms which are based on object detection architectures [59, 31, 88, 104, 106] are not suitable for this task, as also noted by [71]. These methods output a ranked list of proposed instances, where the different proposals are allowed to overlap each other as each proposal is processed independently of the other. Consequently, these architectures are not suitable where each pixel in the image has to be explained, and assigned a unique label of either a "thing" or "stuff" class as shown in Fig. 4.1. This is in contrast to other instance segmentation methods such as [6, 9, 34, 72, 103].

Figure 4.1: We propose a method to train an instance segmentation network from weak annotations in the form of bounding-boxes and image-level tags. Our network can explain both "thing" and "stuff" classes in the image, and does not produce overlapping instances as common detector-based approaches [59, 31, 88].

In this work, we use weak bounding box annotations for "thing" classes, and image-level tags for "stuff" classes. Whilst there are many previous works on semantic segmentation from image-level labels, the best performing ones [152, 153, 118, 18] used a saliency prior. The salient parts of an image are "thing" classes in popular saliency datasets [27, 159, 136] and this prior therefore does not help at all in segmenting "stuff" as in our case. We also consider the "semi-supervised" case where we have a mixture of weak- and fully-labelled annotations.

To our knowledge, this is the first work which performs weakly-supervised, non-overlapping instance segmentation, allowing our model to explain all "thing" and "stuff" pixels in the image (Fig. 4.1). Furthermore, our model jointly produces semantic- and instance-segmentations of the image, which to our knowledge is the first time such a model has been trained in a weakly-supervised manner. Moreover, to our knowledge, this is the first work to perform either weakly supervised semantic- or instance-segmentation on the Cityscapes dataset. On Pascal VOC, our method achieves about 95% of fully-supervised accuracy on both semantic- and instance-segmentation. Furthermore, we surpass the state-of-the-art on fully-supervised instance segmentation as well. Finally, we use our weakly- and semi-

supervised framework to examine how model performance varies with the number of examples in the training set and the annotation quality of each example, with the aim of helping dataset creators better understand the trade-offs they face in this context.

4.2 Related Work

Instance segmentation is a popular area of scene understanding research. Most top-performing algorithms modify object detection networks to output a ranked list of segments instead of boxes [59, 31, 88, 104, 106, 56]. However, all of these methods process each instance independently and thus overlapping instances are produced – one pixel can be assigned to multiple instances simultaneously. Additionally, object detection based architectures are not suitable for labelling "stuff" classes which cannot be described well by bounding boxes [97]. These limitations, common to all of these methods, have also recently been raised by Kirillov *et al*. [71]. We observe, however, that there are other instance segmentation approaches based on initial semantic segmentation networks [6, 9, 34, 72] which do not produce overlapping instances and can naturally handle "stuff" classes. Our proposed approach extends methods of this type to work with weaker supervision.

Although prior work on weakly-supervised instance segmentation is limited, there are many previous papers on weak semantic segmentation, which is also relevant to our task. Early work in weakly-supervised semantic segmentation considered cases where images were only partially labelled using methods based on Conditional Random Fields (CRFs) [149, 61]. Subsequently, many approaches have achieved high accuracy using only image-level labels [75, 152, 123, 122], bounding boxes [68, 121, 30], scribbles [97] and points [11]. A popular paradigm for these works is "self-training" [133]: a model is trained in a fully-supervised manner

by generating the necessary ground truth with the model itself in an iterative, Expectation-Maximisation (EM)-like procedure [121, 30, 97, 122]. Such approaches are sensitive to the initial, approximate ground truth which is used to bootstrap training of the model. To this end, Khoreva *et al*. [68] showed how, given bounding box annotations, carefully chosen unsupervised foreground-background and segmentation-proposal algorithms could be used to generate high-quality approximate ground truth such that iterative updates to it were not required thereafter.

Our work builds on the "self-training" approach to perform instance segmentation. To our knowledge, only Khoreva *et al*. [68] have published results on weakly-supervised instance segmentation. However, the model used by [68] was not competitive with the existing instance segmentation literature in a fully-supervised setting. Moreover, [68] only considered bounding-box supervision, whilst we consider image-level labels as well. Recent work by [64] modifies Mask-RCNN [59] to train it using fully-labelled examples of some classes, and only bounding box annotations of others. Our proposed method can also be used in a semi-supervised scenario (with a mixture of fully- and weakly-labelled training examples), but unlike [64], our approach works with only weak supervision as well. Furthermore, in contrast to [68] and [64], our method does not produce overlapping instances, handles "stuff" classes and can thus explain every pixel in an image as shown in Fig. 4.1.

4.3 Proposed Approach

We first describe how we generate approximate ground truth data to train semantic- and instance-segmentation models with in Sec. 4.3.1 through 4.3.4. Thereafter, in Sec. 4.3.5, we discuss the network architecture that we use. To demonstrate our method and ensure the reproducibility of our results, we release our approximate

ground truth and the code to generate it[1].

4.3.1 Training with weaker supervision

In a fully-supervised setting, semantic segmentation models are typically trained by performing multinomial logistic regression independently for each pixel in the image. The loss function, the cross-entropy between the ground truth distribution and the prediction, can be written as

$$L = -\sum_{i \in \Omega} \log p(l_i|\mathbf{I}) \tag{4.1}$$

where l_i is the ground truth label at pixel i, $p(l_i|\mathbf{I})$ is the probability (obtained from a softmax activation) predicted by the neural network for the correct label at pixel i of an image $\mathbf{I}$ and Ω is the set of pixels in the image.

In the weakly-supervised scenarios considered in this paper, we do not have reliable annotations for all pixels in Ω. Following recent work [68, 75, 11, 122], we use our weak supervision and image priors to approximate the ground truth for a subset $\Omega' \subset \Omega$ of the pixels in the image. We then train our network using the estimated labels of this smaller subset of pixels. Section 4.3.2 describes how we estimate Ω' and the corresponding labels for images with only bounding-box annotations, and Sec. 4.3.3 for image-level tags.

Our approach to approximating the ground truth is based on the principle of only assigning labels to pixels which we are confident about, and marking the remaining set of pixels, $\Omega \setminus \Omega'$, as "ignore" regions over which the loss is not computed. This is motivated by Bansal *et al*. [10] who observed that sampling only 4% of the pixels in the image for computing the loss during fully-supervised training yielded about the same results as sampling all pixels, as traditionally done.

[1] https://github.com/qizhuli/Weakly-Supervised-Panoptic-Segmentation

(a) Input image (b) Semantic segmentation approximate ground truth (c) Instance segmentation approximate ground truth

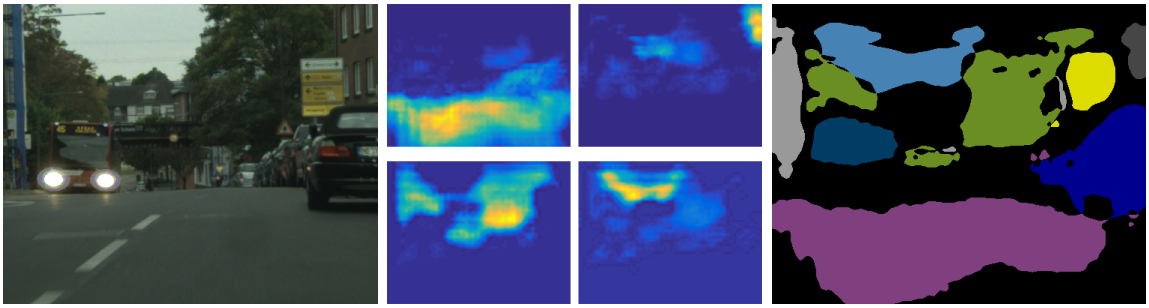
Figure 4.2: An example of generating approximate ground truth from bounding box annotations for an image (a). A pixel is labelled with the bounding-box label if it belongs to the foreground masks of both GrabCut [130] and MCG [4] (b). Approximate instance segmentation ground truth is generated using the fact that each bounding box corresponds to an instance (c). Grey regions are "ignore" labels over which the loss is not computed due to ambiguities in label assignment.

This supported their hypothesis that most of the training data for a pixel-level task are statistically correlated within an image, and that randomly sampling a much smaller set of pixels is sufficient. Moreover, [124] and [85] showed improved results by respectively sampling only 6% and 12% of the hardest pixels, instead of all of them, in fully-supervised training.

4.3.2 Approximate ground truth from bounding box annotations

We use GrabCut [130] (a classic foreground segmentation technique given a bounding-box prior) and MCG [4] (a segment-proposal algorithm) to obtain a foreground mask from a bounding-box annotation, following [68]. To achieve high precision in this approximate labelling, a pixel is only assigned to the object class represented by the bounding box if both GrabCut and MCG agree (Fig. 4.2).

Note that the final stage of MCG uses a random forest trained with pixel-level supervision on Pascal VOC to rank all the proposed segments. We do not perform this ranking step, and obtain a foreground mask from MCG by selecting the proposal that has the highest Intersection over Union (IoU) with the bounding



Input image | Localisation heatmaps for road, building, vegetation and sky | Approximate ground truth generated from image tags

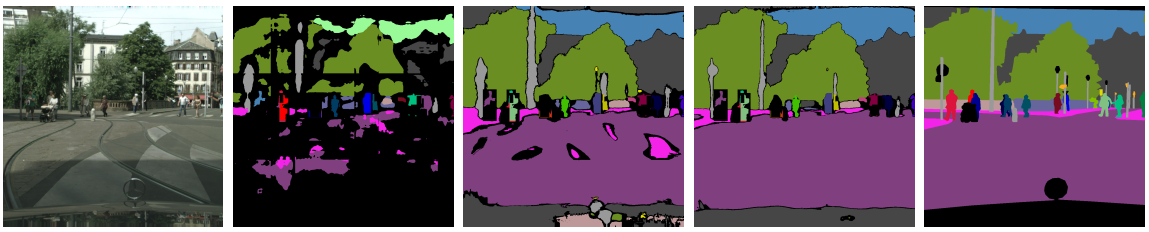
Figure 4.3: Approximate ground truth generated from image-level tags using weak localisation cues from a multi-label classification network. Cluttered scenes from Cityscapes with full "stuff" annotations makes weak localisation more challenging than Pascal VOC and ImageNet that only have "things" labels. Black regions are labelled "ignore". Colours follow Cityscapes convention.

box annotation.

This approach is used to obtain labels for both semantic- and instance-segmentation as shown in Fig. 4.2. As each bounding box corresponds to an instance, the foreground for each box is the annotation for that instance. If the foreground of two bounding boxes of the same class overlap, the region is marked as "ignore" as we do not have enough information to attribute it to either instance.

4.3.3 Approximate ground truth from image-level annotations

When only image-level tags are available, we leverage the fact that CNNs trained for image classification still have localisation information present in their convolutional layers [174]. Consequently, when presented with a dataset of only images and their tags, we first train a network to perform multi-label classification. Thereafter, we extract weak localisation cues for all the object classes that are present in the image (according to the image-level tags). These localisation heatmaps (as shown in Fig. 4.3) are thresholded to obtain the approximate ground truth for a particular class. It is possible for localisation heatmaps for different classes to overlap. In this case, thresholded heatmaps occupying a smaller area are given precedence. We



Input Image | Iteration 0 | Iteration 2 | Iteration 5 | Ground truth

Figure 4.4: By using the output of the trained network, the initial approximate ground truth produced according to Sec. 4.3.2 and 4.3.3 (Iteration 0) can be improved. Black regions are "ignore" labels over which the loss is not computed in training. Note for instance segmentation, permutations of instance labels of the same class are equivalent.

found this rule, like [75], to be effective in preventing small or thin objects from being missed.

Though this approach is independent of the weak localisation method used, we used Grad-CAM [134]. Grad-CAM is agnostic to the network architecture unlike CAM [174] and also achieves better performance than Excitation BP [165] on the ImageNet localisation task [132].

We cannot differentiate different instances of the same class from only image tags as the number of instances is unknown. This form of weak supervision is thus appropriate for "stuff" classes which cannot have multiple instances. Note that saliency priors, used by many works such as [152, 153, 118] on Pascal VOC, are not suitable for "stuff" classes as popular saliency datasets [27, 159, 136] only consider "things" to be salient.

4.3.4 Iterative ground truth approximation

The ground truth approximated in Sec. 4.3.2 and 4.3.3 can be used to train a network from random initialisation. However, the ground truth can subsequently be iteratively refined by using the outputs of the network on the training set as the new approximate ground truth as shown in Fig 4.4. The network's output is

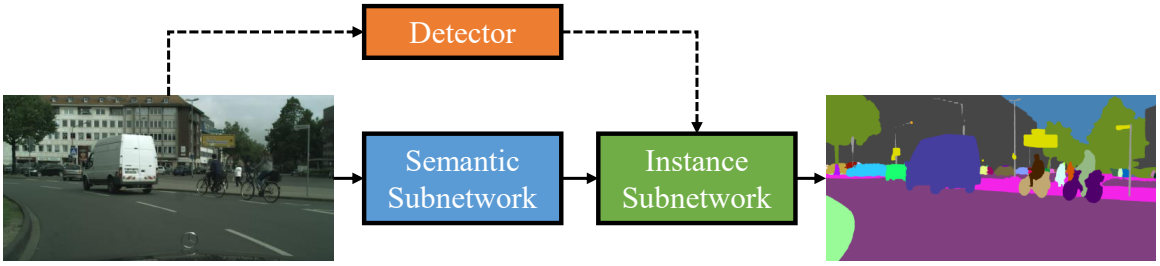


Figure 4.5: Overview of the network architecture. An initial semantic segmentation is partitioned into an instance segmentation, using the output of an object detector as a cue. Dashed lines indicate paths which are not backpropagated through during training.

also post-processed with DenseCRF [76] using the parameters of Deeplab [20] (as also done by [75, 68]) to improve the predictions at boundaries. Moreover, any pixel labelled a "thing" class that is outside the bounding-box of the "thing" class is set to "ignore" as we are certain that a pixel for a thing class cannot be outside its bounding box. For a dataset such as Pascal VOC, we can set these pixels to be "background" rather than "ignore". This is because "background" is the only "stuff" class in the dataset.

4.3.5 Network architecture

Using the approximate ground truth generation method described in this section, we can train a variety of segmentation models. Moreover, we can trivially combine this with full human-annotations to operate in a semi-supervised setting. We use the architecture of Arnab *et al*. [6] as it produces both semantic- and instance-segmentations, and can be trained end-to-end, given object detections. This network consists of a semantic segmentation subnetwork, followed by an instance subnetwork which partitions the initial semantic segmentation into an instance segmentation with the aid of object detections, as shown in Fig. 4.5.

We denote the output of the first module, which can be any semantic segmentation network, as $\mathbf{Q}$ where $Q_i(l)$ is the probability of pixel i of being assigned

semantic label l. The instance subnetwork has two inputs – $\mathbf{Q}$ and a set of object detections for the image. There are D detections, each of the form (l_d, s_d, B_d) where l_d is the detected class label, $s_d \in [0, 1]$ the score and B_d the set of pixels lying within the bounding box of the d^{th} detection. This model assumes that each object detection represents a possible instance, and it assigns every pixel in the initial semantic segmentation an instance label using a Conditional Random Field (CRF). This is done by defining a multinomial random variable, X_i, at each of the N pixels in the image, with $\mathbf{X} = [X_1, X_2 \ldots, X_N]^\top$. This variable takes on a label from the set $\{1, \ldots, D\}$ where D is the number of detections. This formulation ensures that each pixel can only be assigned one label. The energy of the assignment $\mathbf{x}$ to all instance variables $\mathbf{X}$ is then defined as

$$E(\mathbf{X} = \mathbf{x}) = -\sum_{i}^{N} \ln \left(w_1 \psi_{Box}(x_i) + w_2 \psi_{Global}(x_i) + \epsilon \right) + \sum_{i<j}^{N} \psi_{Pairwise}(x_i, x_j). \quad (4.2)$$

The first unary term, the box term, encourages a pixel to be assigned to the instance represented by a detection if it falls within its bounding box,

$$\psi_{Box}(X_i = k) = \begin{cases} s_k Q_i(l_k) & \text{if } i \in B_k \\ 0 & \text{otherwise.} \end{cases} \quad (4.3)$$

Note that this term is robust to false-positive detections [6] since it is low if the semantic segmentation at pixel i, $Q_i(l_k)$ does not agree with the detected label, l_k. The global term,

$$\psi_{Global}(X_i = k) = Q_i(l_k), \quad (4.4)$$

is independent of bounding boxes and can thus overcome errors in mislocalised bounding boxes not covering the whole instance. Finally, the pairwise term is the common densely connected Gaussian and bilateral filter [76] encouraging appear-

ance and spatial consistency.

In contrast to [6], we also consider stuff classes (which object detectors are not trained for), by simply adding "dummy" detections covering the whole image with a score of 1 for all stuff classes in the dataset. This allows our network to jointly segment all "things" and "stuff" classes at an instance level. As mentioned before, the box and global unary terms are not affected by false-positive detections arising from detections for classes that do not correspond to the initial semantic segmentation $\mathbf{Q}$. The Maximum-a-Posteriori (MAP) estimate of the CRF is the final labelling, and this is obtained by using mean-field inference, which is formulated as a differentiable, recurrent network [172, 8].

We first train the semantic segmentation subnetwork using a standard cross-entropy loss with the approximate ground truth described in Sec 4.3.2 and 4.3.3. Thereafter, we append the instance subnetwork and finetune the entire network end-to-end. For the instance subnetwork, the loss function must take into account that different permutations of the same instance labelling are equivalent. As a result, the ground truth is "matched" to the prediction before the cross-entropy loss is computed as described in [6].

4.4 Experimental Evaluation

4.4.1 Experimental set-up

Datasets and weak supervision We evaluate on two standard segmentation datasets, Pascal VOC [39] and Cityscapes [29]. Our weakly- and fully-supervised experiments are trained with the same images, but in the former case, pixel-level ground truth is approximated as described in Sec. 4.3.1 through 4.3.4.

Pascal VOC has 20 "thing" classes annotated, for which we use bounding box supervision. There is a single "background" class for all other object classes.

Following common practice on this dataset, we utilise additional images from the SBD dataset [55] to obtain a training set of 10582 images. In some of our experiments, we also use 54000 images from Microsoft COCO [101] only for the initial pretraining of the semantic subnetwork. We evaluate on the validation set, of 1449 images, as the evaluation server is not available for instance segmentation.

Cityscapes has 8 "thing" classes, for which we use bounding box annotations, and 11 "stuff" class labels for which we use image-level tags. We train our initial semantic segmentation model with the images for which 19998 coarse and 2975 fine annotations are available. Thereafter, we train our instance segmentation network using the 2975 images with fine annotations available as these have instance ground truth labelled. Details of the multi-label classification network we trained in order to obtain weak localisation cues from image-level tags (Sec. 4.3.3) are described in Appendix 4.B.2. When using Grad-CAM, the original authors originally used a threshold of 15% of the maximum value for weak localisation on ImageNet. However, we increased the threshold to 50% to obtain higher precision on this more cluttered dataset.

Network training Our underlying segmentation network is a re-implementation of PSPNet [170]. For fair comparison to our weakly-supervised model, we train a fully-supervised model ourselves, using the same training hyperparameters (detailed in Appendix 4.B) instead of using the authors' public, fully-supervised model. The original PSPNet implementation [170] used a large batch size synchronised over 16 GPUs, as larger batch sizes give better estimates of batch statistics used for batch normalisation [170, 22]. In contrast, our experiments are performed on a single GPU with a batch size of one 521×521 image crop. As a small batch size gives noisy estimates of batch statistics, our batch statistics are "frozen" to the values from the ImageNet-pretrained model as common practice [21, 65]. Our instance

subnetwork requires object detections, and we train Faster-RCNN [127] for this task. All our networks use a ResNet-101 [60] backbone.

Evaluation Metrics We use the AP^r metric [56], commonly used in evaluating instance segmentation. It extends the AP, a ranking metric used in object detection [39], to segmentation where a predicted instance is considered correct if its Intersection over Union (IoU) with the ground truth instance is more than a certain threshold. We also report the AP^r_{vol} which is the mean AP^r across a range of IoU thresholds. Following the literature, we use a range of 0.1 to 0.9 in increments of 0.1 on VOC, and 0.5 to 0.95 in increments of 0.05 on Cityscapes.

However, as noted by several authors [161, 6, 9, 71], the AP^r is a ranking metric that does not penalise methods which predict more instances than there actually are in the image as long as they are ranked correctly. Moreover, as it considers each instance independently, it does not penalise overlapping instances. As a result, we also report the Panoptic Quality (PQ) recently proposed by [71],

$$\text{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \text{IoU}(p,g)}{|TP|}}_{\text{Segmentation Quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Recognition Quality (RQ)}}, \tag{4.5}$$

where p and g are the predicted and ground truth segments, and TP, FP and FN respectively denote the set of true positives, false positives and false negatives.

4.4.2 Results on Pascal VOC

Tables 4.1 and 4.2 show the state-of-art results of our method for semantic- and instance-segmentation respectively. For both semantic- and instance-segmentation, our weakly-supervised model obtains about 95% of the performance of its fully-supervised counterpart, emphasising that accurate models can be learned from only bounding box annotations, which are significantly quicker and cheaper to obtain than pixel-wise annotations. Table 4.2 also shows that our weakly-supervised

Table 4.1: Comparison of semantic segmentation performance to recent methods using only weak, bounding-box supervision on Pascal VOC. Note that [30] and [121] use the less accurate VGG network, whilst we and [68] use ResNet-101. "FS%" denotes the percentage of fully-supervised performance.

Method	Validation set			Test set		
	IoU (weak)	IoU (full)	FS%	IoU (weak)	IoU (full)	FS%
Without COCO annotations						
BoxSup [30]	62.0	63.8	**97.2**	64.6	–	–
Deeplab WSSL [121]	60.6	67.6	89.6	62.2	70.3	88.5
SDI [68]	69.4	74.5	93.2	–	–	–
Ours	**74.3**	**77.3**	96.1	**75.5**	**78.6**	**96.3**
With COCO annotations						
SDI [68]	74.2	77.7	95.5	–	–	–
Ours	**75.7**	**79.0**	**95.8**	**76.7**	**79.4**	**96.6**

model outperforms some recent fully supervised instance segmentation methods such as [7] and [92]. Moreover, our fully-supervised instance segmentation model outperforms all previous work on this dataset. The main difference of our model to [6] is that our network is based on the PSPNet architecture using ResNet-101, whilst [6] used the network of [5] based on VGG [139].

We can obtain semantic segmentations from the output of our semantic subnetwork, or from the final instance segmentation (as we produce non-overlapping instances) by taking the union of all instances which have the same semantic label. We find that the IoU obtained from the final instance segmentation, and the initial pretrained semantic subnetwork to be very similar, and report the latter in Tab. 4.1. Further qualitative and quantitative results, including success and failure cases, are included in Appendix 4.A.

End-to-end training of instance subnetwork Our instance subnetwork can be trained in a piecewise fashion, or the entire network including the semantic subnetwork can be trained end-to-end. End-to-end training was shown to obtain higher performance by [6] for full supervision. We also observe this effect for weak

Table 4.2: Comparison of instance segmentation performance to recent (fully- and weakly-supervised) methods on the VOC 2012 validation set.

Method	AP^r 0.5	0.6	0.7	0.8	0.9	AP^r_{vol}	PQ
Weakly supervised without COCO							
SDI [68]	44.8	–	–	–	–	–	–
Ours	**60.5**	**55.2**	**47.8**	**37.6**	**21.6**	**55.6**	**59.0**
Fully supervised without COCO							
SDS [56]	43.8	34.5	21.3	8.7	0.9	–	–
Chen *et al*. [25]	46.3	38.2	27.0	13.5	2.6	–	–
PFN [92]	58.7	51.3	42.5	31.2	15.7	52.3	–
Ours (fully supervised)	**63.6**	**59.5**	**53.8**	**44.7**	**30.2**	**59.2**	**62.7**
Weakly supervised with COCO							
SDI [68]	46.4	–	–	–	–	–	–
Ours	**60.9**	**55.9**	**48.0**	**37.2**	**21.7**	**55.5**	**59.5**
Fully supervised with COCO							
Arnab *et al*. [7]	58.3	52.4	45.4	34.9	20.1	53.1	–
MPA [106]	62.1	56.6	47.4	36.1	18.5	56.5	–
Arnab *et al*. [6]	61.7	55.5	48.6	39.5	25.1	57.5	–
SGN [103]	61.4	55.9	49.9	42.1	26.9	–	–
Ours (fully supervised)	**63.9**	**59.3**	**54.3**	**45.4**	**30.2**	**59.5**	**63.1**

supervision from bounding box annotations. A weakly-supervised model, trained with COCO annotations improves from an AP^r_{vol} of 53.3 to 55.5. When not using COCO for training the initial semantic subnetwork, a slightly higher increase by 3.9 from 51.7 is observed. This emphasises that our training strategy (Sec. 4.3.1) is effective for both semantic- and instance-segmentation.

Iterative training The approximate ground truth used to train our model can also be generated iteratively, as discussed in Sec. 4.3.4. However, as the results from a single iteration (Tab. 4.1 and 4.2) are already very close to the fully-supervised performance, this offers negligible benefit. Iterative training is, however, crucial for obtaining good results on Cityscapes as discussed in Sec. 4.4.3.

Semi-Supervision We also consider the case where we have a combination of weak and full annotations. As shown in Tab. 4.3, we consider all combinations of weak- and full-supervision of the training data from Pascal VOC and COCO. Table 4.3 shows that training with fully-supervised data from COCO and weakly-supervised data from VOC performs about the same as weak supervision from both datasets for both semantic- and instance-segmentation. Furthermore, training with fully annotated VOC data and weakly labelled COCO data obtains similar results to full supervision from both datasets. We have qualitatively observed that the annotations in Pascal VOC are of higher quality than those of Microsoft COCO (random samples from both datasets are shown in Appendix 4.C). And this intuition is evident in the fact that there is not much difference between training with weak or full annotations from COCO. This suggests that in the case of segmentation, per-pixel labelling of additional images is not particularly useful if they are not labelled to a high standard, and that labelling fewer images at a higher quality (Pascal VOC) is more beneficial than labelling many images at a lower quality (COCO). This is because Tab. 4.3 demonstrates how both semantic- and instance-segmentation networks can be trained to achieve similar performance by using only bounding box labels instead of low-quality segmentation masks. The average annotation time can be considered a proxy for segmentation quality. While a COCO instance took an average of 79 seconds to segment [101], this figure is not mentioned for Pascal VOC [39, 38].

4.4.3 Results on Cityscapes

Tables 4.4 and 4.5 present, what to our knowledge is, the first weakly-supervised results for either semantic or instance segmentation on Cityscapes. Table 4.4 shows that, as expected for semantic segmentation, our weakly-supervised model performs better, relative to the fully-supervised model, for "thing" classes compared

Table 4.3: Semantic- and instance-segmentation performance on Pascal VOC with varying levels of supervision from the Pascal and COCO datasets. The former is measured by the IoU, and latter by the AP^r_{vol} and PQ.

Dataset		IoU	AP^r_{vol}	PQ
VOC	COCO			
Weak	Weak	75.7	55.5	59.5
Weak	Full	75.8	56.1	59.8
Full	Weak	77.5	58.9	62.7
Full	Full	79.0	59.5	63.1

Table 4.4: Semantic segmentation performance on the Cityscapes validation set. We use more informative, bounding-box annotations for "thing" classes, and this is evident from the higher IoU than on "stuff" classes for which we only have image-level tags.

Method	IoU (weak)	IoU (full)	FS%
Ours (thing classes)	68.2	70.4	96.9
Ours (stuff classes)	60.2	72.4	83.1
Ours (overall)	63.6	71.6	88.8

to "stuff" classes. This is because we have more informative bounding box labels for "things", compared to only image-level tags for "stuff". For semantic segmentation, we obtain about 97% of the fully-supervised performance for "things" (similar to our results on Pascal VOC) and 83% for "stuff". Note that we evaluate images at a single-scale, and higher absolute scores could be obtained by multi-scale ensembling [170, 21].

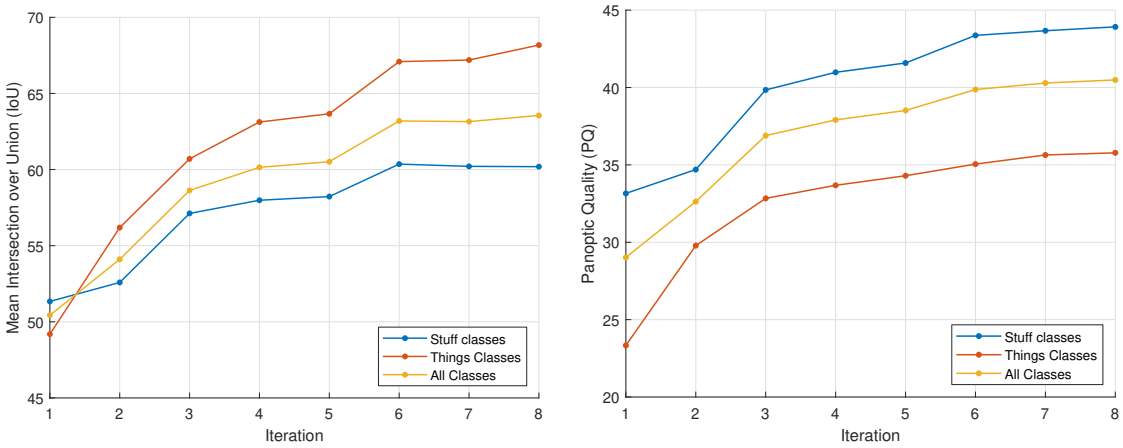
For instance-level segmentation, the fully-supervised ratios for the PQ are similar to the IoU ratio for semantic segmentation. In Tab. 4.5, we report the AP^r_{vol} and PQ for both thing and stuff classes, assuming that there is only one instance of a "stuff" class in the image if it is present. Here, the AP^r_{vol} for "stuff" classes is higher than that for "things". This is because there can only be one instance of a "stuff" class, which makes instances easier to detect, particularly for classes such as "road" which typically occupy a large portion of the image. The Cityscapes evaluation server, and previous work on this dataset, only report the AP^r_{vol} for "thing" classes.

Table 4.5: Instance-level segmentation results on Cityscapes validation and test sets. On the validation set, we report results for both "thing" (th.) and "stuff" (st.) classes. The online server, which evaluates the test set, only computes the AP^r for "thing" classes. We compare to other fully-supervised methods which produce non-overlapping instances. To our knowledge, no published work has evaluated on both "thing" and "stuff" classes. Our fully supervised model, initialised from the public PSPNet model [170] is equivalent to our previous work [6], and competitive with the state-of-art. Note that we cannot use the public PSPNet pretrained model in a weakly-supervised setting.

	Validation									Test
	AP^r_{vol}			PQ			IoU			AP^r_{vol}
Method	th.	st.	all	th.	st.	all	th.	st.	all	th.
Ours (weak, ImageNet init.)	17.0	33.1	26.3	35.8	43.9	40.5	68.2	60.2	63.6	12.8
Ours (full, ImageNet init.)	24.3	42.6	34.9	39.6	52.9	47.3	70.4	72.4	71.6	18.8
Ours (full, PSPNet init.) [6]	28.6	52.6	42.5	42.5	62.1	53.8	80.1	79.5	79.8	23.4
Pixel Encoding [147]	9.9	–	–	–	–	–	–	–	–	8.9
RecAttend [126]	–	–	–	–	–	–	–	–	–	9.5
InstanceCut [72]	–	–	–	–	–	–	–	–	–	13.0
DWT [9]	21.2	–	–	–	–	–	–	–	–	19.4
SGN [103]	29.2	–	–	–	–	–	–	–	–	25.0

As a result, we report results for "stuff" classes only on the validation set. Table 4.5 also compares our results to existing work which produces non-overlapping instances on this dataset, and shows that both our fully- and weakly-supervised models are competitive with recently published work on this dataset. We also include the results of our fully-supervised model, initialised from the public PSPNet model [170] released by the authors, and show that this is competitive with the state-of-art [103] among methods producing non-overlapping segmentations (note that [103] also uses the same PSPNet model). Figure 7 shows some predictions of our weakly-supervised model; further quantitative and qualitative results can be found in Appendix 4.A.

Iterative training Iteratively refining our approximate ground truth during training, as described in Sec. 4.3.4, greatly improves our performance on both semantic- and instance-segmentation as shown in Fig. 4.6. We trained the network for 150 000



(a) Semantic segmentation (IoU) (b) Instance segmentation (PQ)

Figure 4.6: Iteratively refining our approximate ground truth during training improves both semantic and instance segmentation on the Cityscapes validation set.

iterations before regenerating the approximate ground truth using the network's own output on the training set. Unlike on Pascal VOC, iterative training is necessary to obtain good performance on Cityscapes as the approximate ground truth generated on the first iteration is not sufficient to obtain high accuracy. This was expected for "stuff" classes, since we began from weak localisation cues derived from the image-level tags. However, as shown in Fig. 4.6, "thing" classes also improved substantially with iterative training, unlike on Pascal VOC where there was no difference. Compared to VOC, Cityscapes is a more cluttered dataset, and has large scale variations as the distance of an object from the car-mounted camera changes. These dataset differences may explain why the image priors employed by the methods we used (GrabCut [130] and MCG [4]) to obtain approximate ground truth annotations from bounding boxes are less effective. Furthermore, in contrast to Pascal VOC, Cityscapes has frequent co-occurrences of the same objects in many different images, making it more challenging for weakly-supervised methods.

Effect of ranking methods on the AP^r The AP^r metric is a ranking metric derived from object detection. It thus requires predicted instances to be scored such that

Ranking Method	AP^r_{vol} th.	AP^r_{vol} st.	PQ all
Detection score	17.0	26.7	40.5
Mean seg. confidence	14.6	33.1	40.5
Oracle	21.6	37.0	40.5

Table 4.6: The effect of different instance ranking methods on the AP^r_{vol} of our weakly supervised model computed on the Cityscapes validation set.

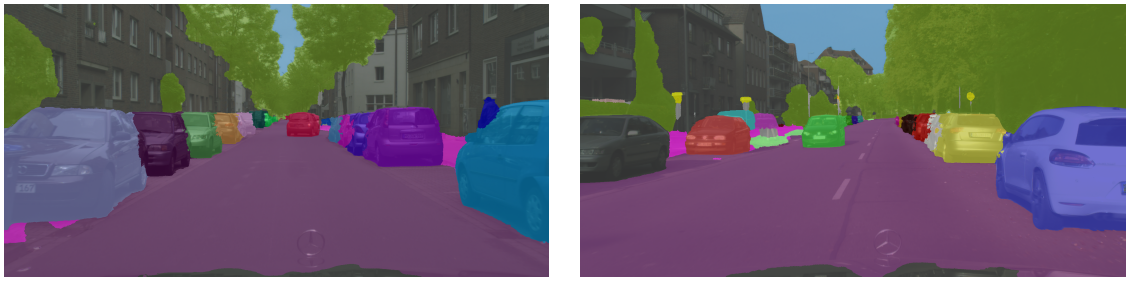


Figure 4.7: Example results on Cityscapes of our weakly supervised model.

they are ranked in the correct relative order. As our network uses object detections as an additional input and each detection represents a possible instance, we set the score of a predicted instance to be equal to the object detection score. For the case of stuff classes, which object detectors are not trained for, we use a constant detection score of 1 as described in Sec. 4.3.5. An alternative to using a constant score for "stuff" classes is to take the mean of the softmax-probability of all pixels within the segmentation mask. Table 4.6 shows that this latter method improves the AP^r for stuff classes. For "things", ranking with the detection score performs better and comes closer to oracle performance which is the maximum AP^r that could be obtained with the predicted instances.

Changing the score of a segmented instance does not change the quality of the actual segmentation, but does impact the AP^r greatly as shown in Tab. 4.6. The PQ, which does not use scores, is unaffected by different ranking methods, and this suggests that it is a better metric for evaluating non-overlapping instance segmentation where each pixel in the image is explained.

4.5 Conclusion

We have presented, to our knowledge, the first weakly-supervised method that jointly produces non-overlapping instance and semantic segmentation for both "thing" and "stuff" classes. Using only bounding boxes, we are able to achieve 95% of state-of-art fully-supervised performance on Pascal VOC. On Cityscapes, we use image-level annotations for "stuff" classes and obtain 88.8% of the fully-supervised performance for semantic segmentation and 85.6% for instance segmentation (measured with the PQ). Crucially, the weak annotations we use incur only about 3% of the time of full labelling. As annotating pixel-level segmentation is time consuming, there is a dilemma between labelling few images with high quality or many images with low quality. Our semi-supervised experiment suggests that the latter is not an effective use of annotation budgets as similar performance can be obtained from only bounding-box annotations.

Future work is to perform instance segmentation using only image-level tags and the number of instances of each object present in the image as supervision. This will require a network architecture that does not use object detections as an additional input.

Acknowledgements This work was supported by Huawei Technologies Co., Ltd., the EPSRC, Clarendon Fund, ERC grant ERC-2012-AdG 321162-HELIOS, EPRSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1.

Appendices

Section 4.A presents further qualitative and quantitative results of our experiments on Cityscapes and Pascal VOC. Section 4.B describes the training of the networks described in Sec. 4.3. In Sec. 4.4.2, it was mentioned that the annotation quality of Pascal VOC [39] is better than COCO [101]. Some randomly drawn images from these datasets are presented to illustrate this point in Sec. 4.C. Finally, Sec. 4.D shows our calculation of how much the overall annotation time is reduced by using weak annotations, in comparison to full annotations, on the Cityscapes dataset.

4.A Additional Qualitative and Quantitative Results

Figure 4.8 and Tab. 4.7 present additional qualitative and quantitative results on the Cityscapes dataset. Similarly, Fig. 4.9 and Tab. 4.8 show additional results on the Pascal VOC dataset.

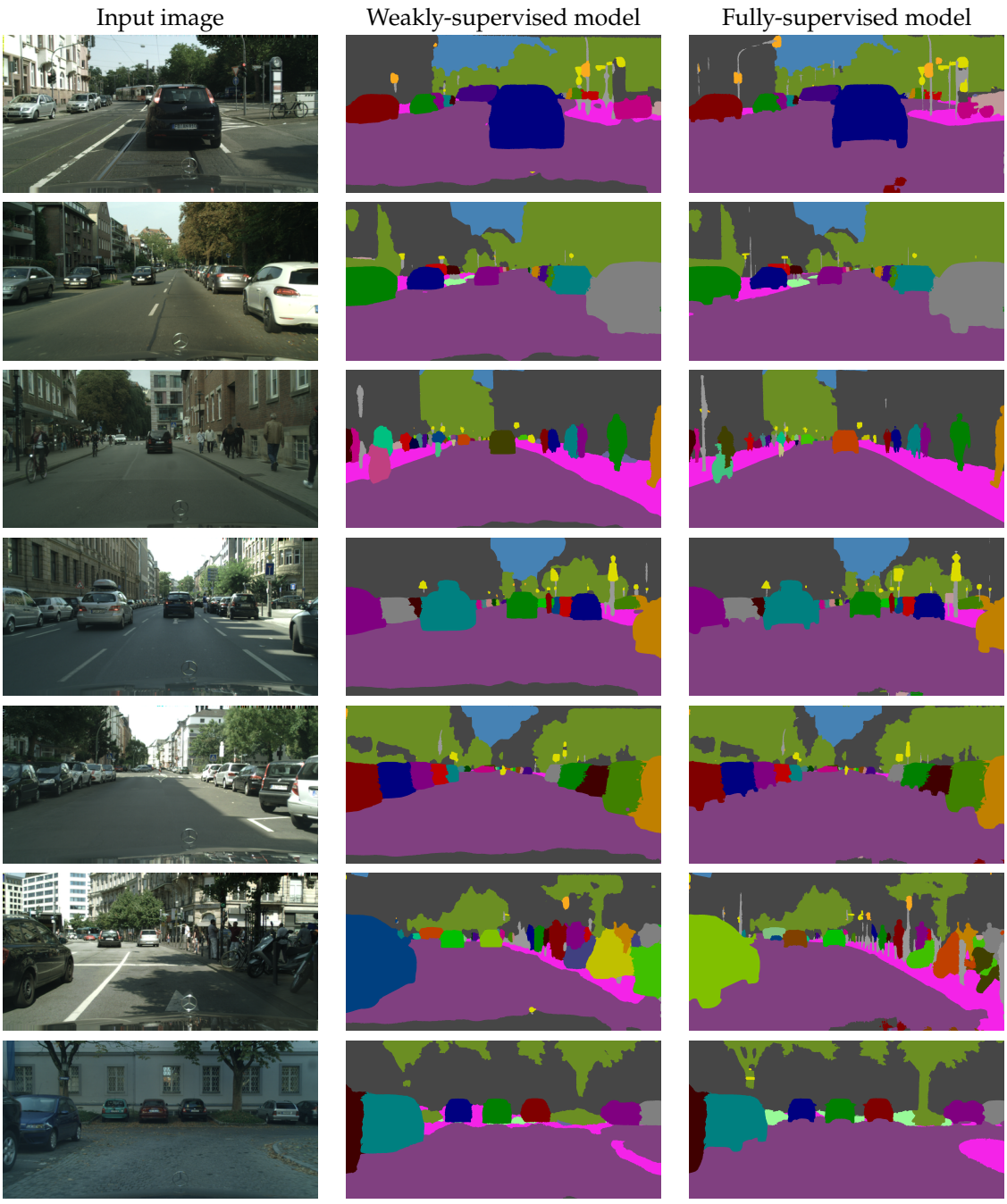


Figure 4.8: Comparison of our weakly- and fully-supervised instance segmentation models on the Cityscapes dataset. The fully-supervised model produces more precise segmentation, as seen by its sharper boundaries. The last row also shows how the fully-supervised model segments "stuff" classes such as "vegetation" and "sidewalk" more accurately. Both of these were expected, as the weakly-supervised model is trained only with bounding box and image tag annotations. Rows 3 and 6 also show some instances with different colouring. Each colour represents an instance ID, and a discrepancy between the two indicates that a different number of instances were segmented.

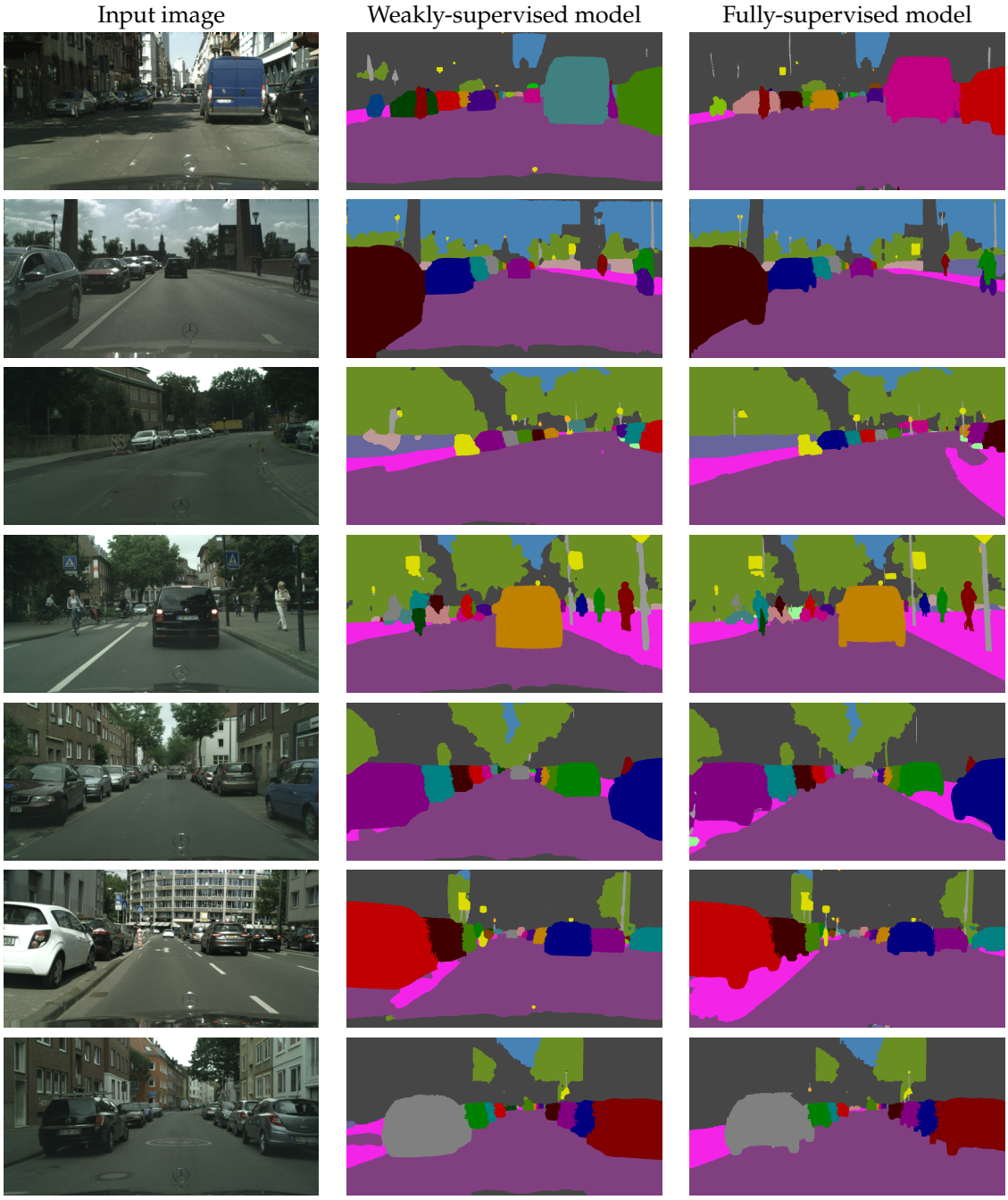


Figure 4.8: *continued*. The last three rows show how the fully-supervised model is also able to segment "stuff" classes such as "sidewalk" more accurately. This was expected since the weakly-supervised model is only trained with image-level tags for "stuff" classes, which provides very little localisation information.

Figure 4.9: Comparison of our weakly- and fully-supervised instance segmentation models on the Pascal VOC validation set. The weakly-supervised model typically obtains results similar to its state-of-the-art, fully-supervised counterpart. However, the fully-supervised model produces more accurate and precise segmentations, as seen in the last two rows.

Figure 4.9: *continued*. The first and second rows show examples where the results of the two models are similar. In the third row, the weakly-supervised model does not segment the "green person" as well as the fully-supervised model. In the last row, both weakly- and fully-supervised models have made an error in not completely segmenting each of the bottles.

Table 4.7: Per-class results of our weakly- and fully-supervised models for both semantic and instance segmentation on the Cityscapes validation set. The IoU measures semantic segmentation performance, whilst the AP^r_{vol} and PQ measure instance segmentation performance.

Metric	Mean	road	side-walk	build-ing	wall	fence	pole	traffic light	traffic sign	vege-tation	terrain	sky	person	rider	car	truck	bus	train	motor-cycle	bi-cycle
Weakly supervised model																				
IoU	63.6	93.3	59.3	86.6	38.7	29.6	32.0	44.0	59.2	88.7	39.1	91.7	69.4	48.4	87.4	68.0	80.7	68.0	56.0	67.5
AP^r_{vol}	26.3	82.7	27.6	68.1	5.9	5.2	0.6	3.0	16.6	74.1	4.7	76.1	11.7	5.0	27.7	17.4	36.3	23.0	9.0	5.9
PQ	40.5	91.2	47.0	79.6	14.8	12.7	5.5	13.2	37.3	83.3	16.2	82.3	30.6	25.7	46.9	33.7	55.5	37.0	31.8	24.9
Fully supervised model																				
IoU	71.6	97.6	81.9	90.4	42.2	52.3	54.5	61.1	71.8	90.5	61.1	93.5	76.6	53.2	93.4	68.3	77.8	70.6	50.7	72.3
AP^r_{vol}	34.9	94.8	56.2	73.6	10.5	7.4	11.9	10.7	31.9	77.3	16.2	78.2	21.2	15.0	32.6	25.5	41.4	30.5	15.3	12.6
PQ	47.3	95.5	67.9	83.4	17.2	15.5	38.0	22.2	54.7	84.7	21.7	80.4	40.4	37.1	49.8	31.8	54.1	36.4	34.3	32.5

Table 4.8: Per-class results of our weakly- and fully-supervised models for both semantic and instance segmentation on the Pascal VOC validation set. The IoU measures semantic segmentation performance, whilst the AP^r_{vol} and PQ measure instance segmentation performance.

Metric	Mean	aero-plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor-bike	per-son	plant	sheep	sofa	train	tv
Weakly supervised model																					
IoU	75.7	85.0	35.9	88.6	70.3	77.9	91.9	83.6	90.5	39.2	84.5	59.4	86.5	82.4	81.5	84.3	57.0	85.9	55.8	85.8	70.4
AP^r_{vol}	55.5	68.8	26.4	74.4	50.4	37.9	70.0	49.4	78.6	22.0	57.1	37.4	78.7	61.6	61.7	50.8	42.2	54.6	46.9	74.9	66.5
PQ	59.5	69.7	18.0	76.8	55.1	48.2	75.4	54.9	77.8	26.4	65.8	43.6	73.8	62.9	68.9	60.8	48.7	62.9	53.7	75.9	71.4
Fully supervised model																					
IoU	79.0	92.0	42.2	90.6	71.1	80.7	95.0	88.5	91.9	41.5	90.6	60.3	86.5	88.3	85.4	86.9	61.7	91.6	53.3	89.2	76.8
AP^r_{vol}	59.5	77.1	31.7	78.1	50.9	40.2	72.4	52.6	82.9	27.0	60.3	35.4	83.1	65.4	72.3	57.3	45.6	56.4	49.7	80.1	71.3
PQ	63.1	77.8	29.1	79.0	57.2	48.9	75.5	59.8	81.7	31.8	67.3	46.2	77.3	69.0	75.3	64.8	52.2	62.0	54.6	79.8	73.7

4.B Experimental Details

4.B.1 Network architecture and training

The underlying semantic segmentation network is a re-implementation of PSPNet [170] as described in Sec. 4.3.5, using a ResNet-101 backbone. This network has an output stride of 8, meaning that the result of the network has to be upsampled by a factor of 8 to obtain the final prediction at the original resolution.

We used most of the same training hyperparameters for training both our fully- and weakly-supervised networks. A batch size of a single 521×521 image crop, momentum of 0.9, and a weight decay of 5×10^{-4} were used in all our experiments.

We trained the semantic segmentation module first, and finetuned the entire instance segmentation network afterwards. For training the semantic segmentation module, the fully supervised models were trained with an initial learning rate of 1×10^{-4}, which was then reduced to 1×10^{-5} when the training loss converged. We used the same learning rate schedule for our weakly-supervised model on Pascal VOC where we did not do any iterative training. In total, about 400k iterations of training were performed. When training our weakly-supervised model iteratively on Cityscapes, we used an initial learning rate of 1×10^{-4} which was then halved for each subsequent stage of iterative training. Each of these iterative training stages was 150k iterations long. Both of the weakly- and fully-supervised models were initialised with ImageNet-pretrained weights and batch normalisation statistics.

In the instance training stage, we fixed the learning rate to 1×10^{-5} for both weakly- and fully-supervised experiments on the VOC and Cityscapes datasets. We observed that a total of 400k iterations were required for the models' training losses to converge.

When training the Faster-RCNN object detector [127], we used all the default training hyperparameters in the publicly available code.

4.B.2 Multi-label classification network

We obtained weak localisation cues, as described in Sec. 4.3.3, by first training a network to perform multi-label classification on the Cityscapes dataset.

We adapted the same PSPNet [170] architecture for segmentation for the classification task: The output of the last convolutional layer (`conv5_4`) is followed by a global average pooling layer to aggregate all the spatial information. Thereafter, a fully connected layer with 19 outputs (the number of classes in the Cityscapes dataset) is appended. This network was then trained with a binary cross-entropy loss for each of the 19 labels in the dataset. The loss for a single image is

$$L = \frac{1}{N}\sum_{i=1}^{N} -y_i \log(\text{sigmoid}(z_i)) - (1 - y_i)\log(1 - \text{sigmoid}(z_i)), \tag{4.6}$$

where $\mathbf{y}$ is the ground truth image-level label vector and $y_i = 1$ if the i^{th} class is present in the image and 0 otherwise. z_i is the logit for the i^{th} class output by the final fully connected layer in the network.

It is not possible to fit an entire 2048×1024 Cityscapes image in memory to perform multi-label classification. Using the PSPNet architecture described above (with an output stride of 8), it would take 48.8 GB of memory to train a network with a batch size of 1. Even the standard ResNet-101 architecture [60] (which has a higher output stride of 32, and thus sixteen times less spatial resolution) would take 21.7 GB of memory, which is still almost double the 12GB available in our Titan X GPU. Consequently, we took 15 fixed crops of size 500×400 from the original 2048×1024 image and trained with these crops instead. We were careful not to take random crops during training, as this could be a form of extra supervision. Instead, as we took 15 fixed crops which tile the image and derived image-level labels from them, it effectively means that in a real-world scenario annotators would be asked to annotate image-level labels for fifteen 500×400 images rather than a

single 2048×1024 image.

This multi-label classification network was trained with a batch size of 1 and a fixed learning rate of 1×10^{-4} until the training loss converged. We found that this occurred after 50k iterations of training. At this point, the mean Average Precision (mAP) on the validation set was 78.8. The mAP is also used by the Pascal VOC dataset to benchmark multi-label classification [39].

4.C Comparison of Pascal VOC and Microsoft COCO Annotation Quality

Section 4.4.2 mentioned that images in Pascal VOC [39] are annotated at a higher quality than those in Microsoft COCO [101]. Figure 4.10 illustrates this observation. Images were randomly drawn from Microsoft COCO, and then images from Pascal VOC with the same semantic classes present are shown alongside for comparison. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect.

4.D Calculation of Reduction Factor in Annotation Time if Only Weak Labels Are Used

The Cityscapes dataset has 11 "stuff" classes, and 8 "thing" classes annotated. Over the training and validation sets, there are an average of 17.9 instances of "thing" classes per full-resolution, 2048×1024 image.

For the calculation in Sec. 4.1, we assumed that each instance of a "thing" class is labelled with a bounding box, and that image-level tags are annotated for all present "stuff" classes. We assumed that a bounding box takes 7 seconds per instance to draw [120] and that an image-level tag takes 1 second to label [119].

Therefore the average time to annotate "thing" classes with a bounding-box is $17.9 \times 7 = 125.3$ seconds. As we took 15 fixed crops per image (as described in Sec. 4.B.2) and there are an average of 3.8 "stuff" tags per crop, the average time to annotate stuff classes is $15 \times 3.8 = 57$ seconds. This totals 182.3 seconds $= 3.0$ minutes per image. Thus the annotation time is reduced by a factor of 29.6 (since the images originally required 90 minutes to label at a pixel level by hand [29]) if weak annotations in the form of bounding boxes and image-level tags are used.

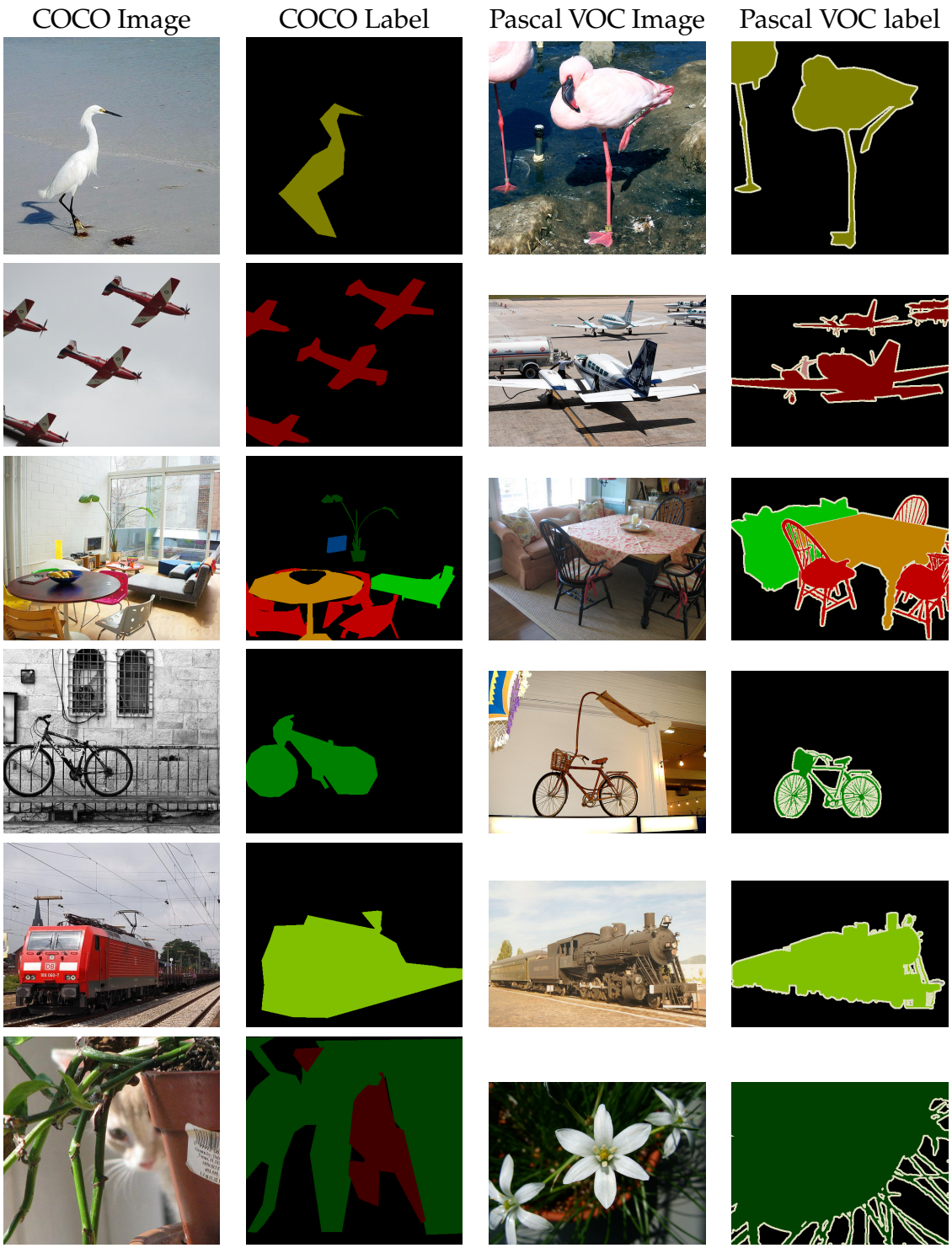


Figure 4.10: Comparison of the annotation quality of images in the Microsoft COCO and Pascal VOC datasets. An image was randomly drawn from COCO, and an image from Pascal VOC with similar content is shown alongside it. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect. Grey regions in the Pascal images indicate "void" regions where the annotator was unsure of the correct label.

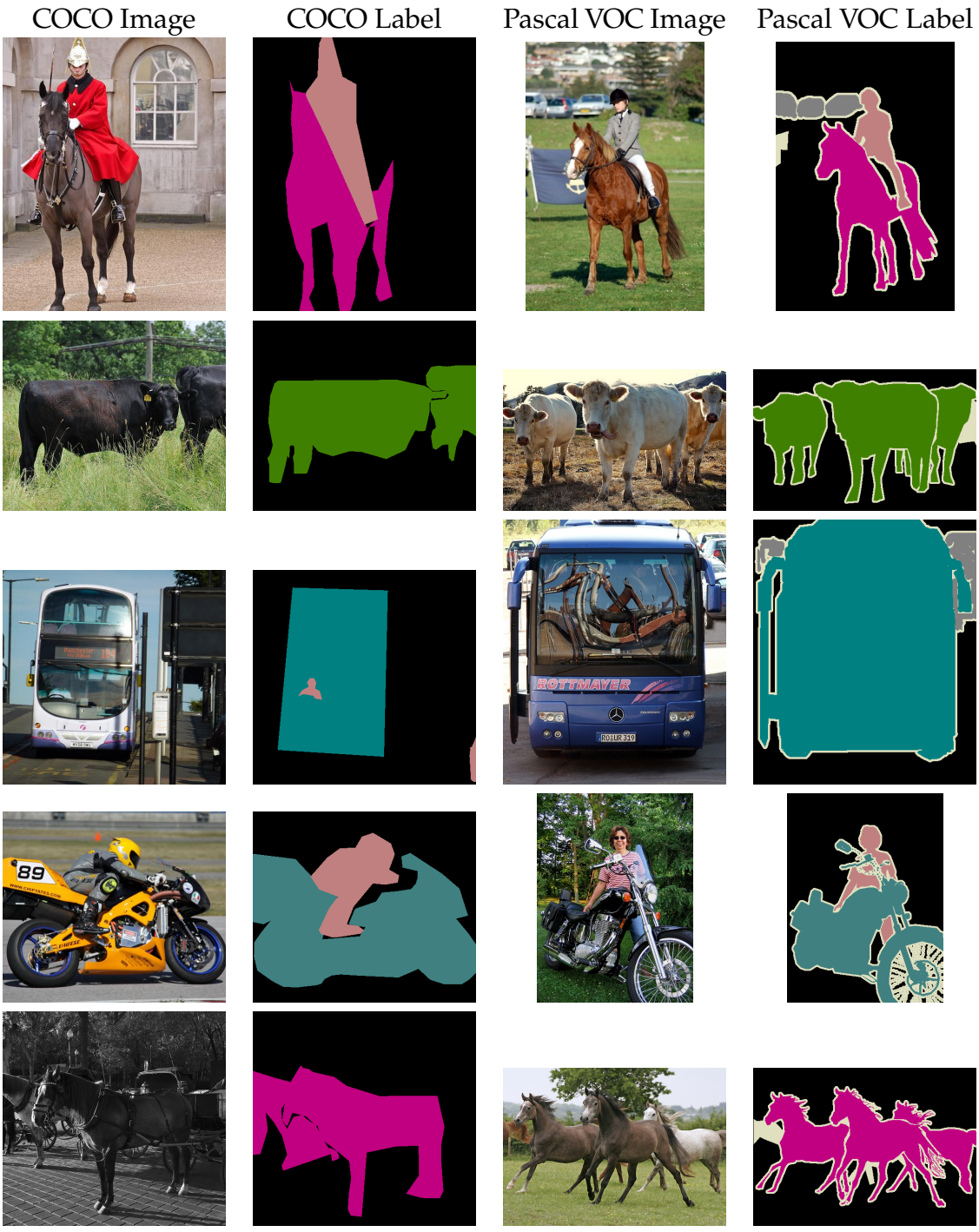


Figure 4.10: *continued*. Comparison of the annotation quality of images in the Microsoft COCO and Pascal VOC datasets. An image was randomly drawn from COCO, and an image from Pascal VOC with similar content is shown alongside it. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect. Grey regions in the Pascal images indicate "void" regions where the annotator was unsure of the correct label.

Chapter 5

Unifying Training and Inference for Panoptic Segmentation

Qizhu Li

University of Oxford

Xiaojuan Qi

University of Oxford

Philip H. S. Torr

University of Oxford

Abstract

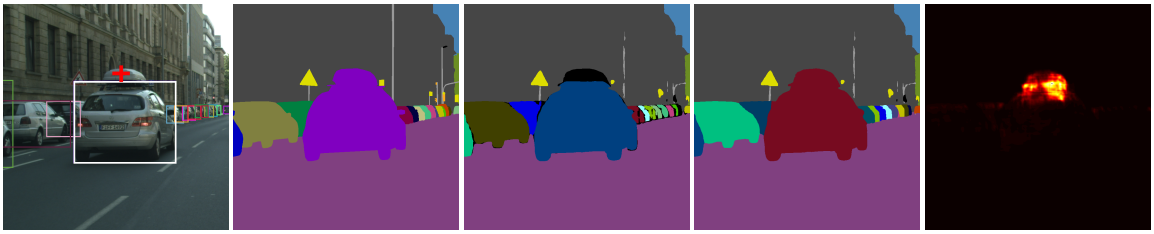
We present an end-to-end network to bridge the gap between training and inference pipeline for panoptic segmentation, a task that seeks to partition an image into semantic regions for "stuff" and object instances for "things". In contrast to recent works, our network exploits a parametrised, yet lightweight panoptic segmentation submodule, powered by an end-to-end learnt dense instance affinity, to capture the

probability that any pair of pixels belong to the same instance. This panoptic submodule gives rise to a novel propagation mechanism for panoptic logits and enables the network to output a coherent panoptic segmentation map for both "stuff" and "thing" classes, without any post-processing. Reaping the benefits of end-to-end training, our full system sets new records on the popular street scene dataset, Cityscapes, achieving 61.4 PQ with a ResNet-50 backbone using only the `fine` annotations. On the challenging COCO dataset, our ResNet-50-based network also delivers state-of-the-art accuracy of 43.4 PQ. Moreover, our network flexibly works with and without object mask cues, performing competitively under both settings, which may be of interest for applications with computation budgets.

5.1 Introduction

As a pixel-wise classification task, panoptic segmentation aims to achieve a seamless semantic understanding of all countable and uncountable objects in a scene - *a.k.a.* "things" and "stuff" respectively, and delineate the instance boundaries of objects where semantically possible.

While early attempts at tackling panoptic segmentation often resort to two separate networks for instance and semantic segmentation, recent works [87, 83, 70, 157, 160] are able to improve the overall efficiency by constructing the two branches on a single, shared feature extractor, and training the multi-head, multi-task network jointly. However, these works have stopped short of devising an end-to-end pipeline for panoptic segmentation, as they all adopt a post-processing stage with heuristics to combine the different outputs of their multi-task networks, following [71, 70]. Such pipelines suffer from several shortcomings. Firstly, post-



(a) Image with boxes (b) Ground truth (c) Heuristic fusion (d) Our approach (e) Instance affinities

Figure 5.1: Comparison of our approach versus the heuristic rule-based method of [70]. We overlay the predicted bounding boxes on the input images for visualisation. For the cross-marked pixel in (a) which falls outside its bounding box, we show its instance affinities in (e). Heuristics-based fusion [70] produces truncated objects when localisation is not accurate, while our instance affinity enables the network to recover the full object, by propagating information between pixels with strong instance affinities. Best viewed in colour.

processing often requires a time-consuming trial-and-error procedure to mine a good set of hyperparameters, which may need to be repeated for each image domain. As the performance of an algorithm can be quite sensitive to the choice of hyperparameters, how well a method performs can quickly degenerate to a function of the amount of computation resources at its disposal [76, 70]. Secondly, methods without an explicit loss function for panoptic segmentation [87, 83, 70, 160] cannot directly optimise for the ultimate goal. Even with expert knowledge, it is difficult to design an exhaustive set of rules and remedies for all failure modes. An example is shown in Fig. 5.1 (c): after the heuristic post-processing, the missing part of the car cannot be recovered.

To achieve an end-to-end system, we reckon three challenging steps need to be taken: (1) unify the training and inference, enabling the network to *differentiably* produce panoptic segmentation during training; (2) embed a data-driven mechanism in the multi-task network whereby imperfect and coarse cues can be cleaned and corrected; (3) design an appropriate loss function to directly optimise the global objective for panoptic segmentation.

To achieve (1) and (2), we propose a novel pipeline using segmentation and

localisation cues to predict a coherent panoptic segmentation in an end-to-end manner. At the heart of this pipeline lie a *dynamic potential head* – a parameter-free stage that represents a dynamic number of panoptic instances, and a *dense instance affinity head* – a parametrised, efficient, and data-driven module that predicts and utilises the likelihood for any pair of pixels to belong to the same "thing" instance or "stuff" class. These two differentiable heads produce full panoptic segmentation during training and inference, eradicating the train-test logic discrepancy.

Furthermore, to fulfil (3), we propose a *panoptic matching loss* which computes loss directly on panoptic segmentation. This objective function, together with the differentiable nature of our proposed panoptic head, enables the network to learn in an end-to-end manner. To our best knowledge, our loss is the first to perform online segment matching before computing a cross-entropy loss in an end-to-end panoptic segmentation system. The matching step allows training the network with *predicted* detections, thereby incentivising it to handle imperfect localisation cues. While the idea is not convoluted, our ablation studies (Tab. 5.8) show that doing so – as opposed to training with ground truth detections – yields performance gains.

By closing the gap between training and inference, the network enjoys improved accuracy in challenging scenarios. As illustrated in Fig. 5.1, by aggregating panoptic logits across the whole image according to the predicted affinity strengths (Fig. 5.1e), our parametrised panoptic head is able to fix inaccurate predictions from a previous stage - truncated objects due to imperfect bounding box localisations (Fig. 5.1c).

Last but not least, thanks to its power of improving coarse panoptic logits, our network achieves competitive performance even without using object mask cues, which are required in most recent approaches [87, 83, 70, 157]. This means our method can offer an additional degree of flexibility in terms of network design, a trait desirable for applications with a limited computation and time budget.

On the challenging Cityscapes and COCO datasets, our models set new records for ResNet-50-based networks, achieving panoptic qualities (PQ) of 61.4 and 43.4 respectively.

5.2 Related Work

Arguably, the problem of panoptic segmentation can be viewed as a combination of instance and semantic segmentation. Indeed, this interpretation has guided many recent works on panoptic segmentation [71, 70], where it is largely approached as a bi-task problem, and the focus is placed on solving both sub-problems simultaneously and efficiently. Shared features of these works include the use of networks with multiple specialised subnets for each sub-task, and the lack of an explicit objective on panoptic segmentation.

In addition to the inclusion of "stuff" classes, another major difference between panoptic and instance segmentation is that the former requires all pixels to be given a unique label, whereas the latter does not. As a result, "thing" predictions from an off-the-shelf detection-driven instance segmentation network – *e.g.*, Mask-RCNN [59] – cannot be readily inserted into the panoptic prediction, as pixels need to have their conflicting instance labels resolved. Moreover, contradictions between the semantic and instance branch must also be carefully resolved. This prompted recent works to adopt an offline postprocessing step first described in [71] to perform conflict resolution and merger of instance and semantic predictions, based on a set of carefully tuned heuristics. A number of works have also attempted to encourage consistency between semantic and instance predictions by adding a communication mechanism between the two subnets [83, 87]. However, as these proposed changes do not modify the output format of the network, they still rely on postprocessing to produce panoptic predictions. In addition, Liu *et al*. proposes

to directly learn the ordering of "thing" instances for conflict resolution [102]. However, this approach does not handle overlapping instances pixel-by-pixel – as it predicts a single ranking score for each instance – and does not reconcile conflicts between "stuff" and "thing".

A small number of works have attempted to advance towards an end-to-end network with a unified train-test logic. We observe that [86] extends a dynamically instantiated instance segmentation network described in [6] to solve the panoptic segmentation problem. It produces non-overlapping segments by design, and is trained end-to-end, given detections. However, it is prone to failures when objects of the same class are nearby and similarly coloured. Moreover, its Instance CRF suffers from the very small number of trainable parameters (since the compatibility transforms are frozen as the Potts model), and is made less attractive by the need to grid search good kernel variances for the bilateral filters in the message passing step.

Recently, Xiong *et al*. [157] modifies the unary terms of [6, 86] and proposes a parameter-free, differentiable panoptic head to fuse semantic and instance segmentation predictions during training. Similar to [86], it allows a panoptic loss to be directly applied on the fused probabilities. However, in the inference phase, it still resorts to several heuristic strategies (*e.g*., overlap-based instance mask pruning) and relies on a complex voting mechanism to determine the semantic categories of predicted segments, deviating from a unified training and inference pipeline. Furthermore, the effectiveness of their parameter-free panoptic head heavily depends on the quality of semantic and instance predictions it receives, since it arguably functions as an online heuristic merger due to the absence of learnable weights.

Also pertinent to this work is the extensive research carried out around the techniques of long-range contextual aggregation. Aside from CRF-driven methods [76, 173, 6], Bertasius *et al*. proposes a semantic segmentation method based on

random walks to learn and predict inter-pixel affinity graphs, and iteratively multiply the learnt affinity with an initial segmentation to achieve convergence [13]. Lately, another technique, self-attention, has been successful in several vision tasks [151, 163, 45]. However, its quadratic memory and computation complexity has cast doubt over its practicality. To mitigate this problem, Shen *et al*. [135] suggest to invoke the associativity of matrix multiplication and avoid the explicit production of expensive attention maps. This approach effectively reduces the complexity to a linear one, $O(HW)$, making it suitable for pixel-level labelling tasks.

Albeit sharing certain operational similarities with self-attention and non-local methods [163, 66, 151], our proposed dense instance affinity head serves a different purpose, and cannot be substituted by directly inserting these operations in the backbone. The aforementioned methods work by enhancing the expressiveness of extracted features, as reflected in the fact that these actions are performed in the feature space, and can generally lead to performance gains for many tasks. In contrast, our proposed instance affinity is not a generic feature enhancer. It is specifically designed and tasked to model the pairwise probability for any two pixels to belong in the same "thing" instance or "stuff" category. This relationship in turn enables our network to revise and resolve. With this purpose in mind, we incorporate insights from [135] to construct a module that is lightweight, learnable, and agnostic to the number of channels, allowing us to model a dynamic number of instances across different images.

5.3 Proposed Approach

Our proposed network (Fig. 5.2) consists of four blocks. A shared *fully convolutional backbone* extracts a set of features. Operating on these features, a *semantic*

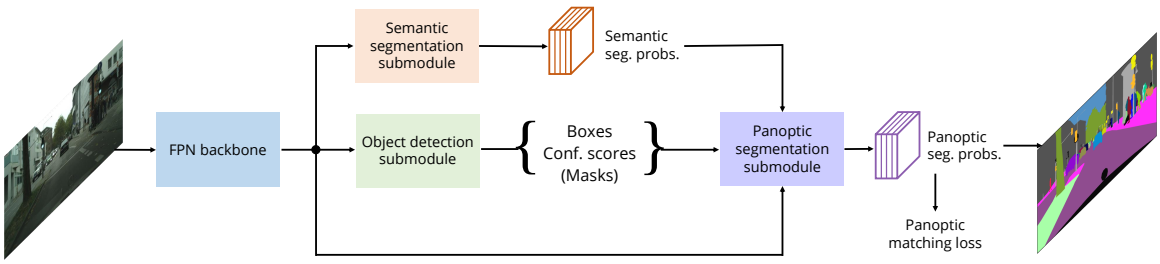


Figure 5.2: Overview of the network architecture. Semantic segmentation and object detections are fed into the proposed panoptic segmentation submodule – including a dynamic potential head and a dense instance affinity head – to produce panoptic segmentation predictions without requiring post-processing. All components are differentiable, and the network is trained end-to-end.

segmentation submodule and an *object detection submodule* produce segmentation and localisation cues, which are fused and revised by the proposed *panoptic segmentation submodule*. All components are differentiable and trained jointly, end-to-end.

5.3.1 Backbone

The pipeline starts with a shared fully convolutional backbone, which takes an input image of spatial dimension $H \times W$, and generates a set of features $\boldsymbol{F}$. In our experiments, we adopt a simple ResNet-FPN backbone that outputs four multi-scale feature maps [100], following a common practice in prior works [70, 157]. To encourage global consistency, we carry out a squeeze-and-excitation operation [63] on the top-level ResNet feature before producing the first FPN feature. A similar strategy is used in [157].

5.3.2 Semantic segmentation submodule

The backbone features $\boldsymbol{F}$ are fed into the semantic segmentation submodule to produce a $\frac{H}{d} \times \frac{W}{d} \times (N_{st} + N_{th})$ tensor $\boldsymbol{V}$, where N_{st} and N_{th} are the numbers of "stuff" and "thing" classes respectively. $V_i(l)$ denotes the probability that pixel p_i belongs to semantic class l. The spatial dimension is downsampled d times to strike

a balance between resolution and complexity. We choose d as 4 in the experiments.

Multiple implementations for this submodule have been proposed in the literature, all showing decent performance [70, 157]. In this work, we modify the design in [157] by inserting a Group Normalisation operation [155] after each convolution, which has been observed to help stabilise training. Please refer to Appendix 5.A.1 for further details.

5.3.3 Object detection submodule

In parallel, the features $\boldsymbol{F}$ are also passed to an object detection submodule, which generates D object detections, consisting of bounding boxes $\boldsymbol{B} = \{B_1, B_2, B_3, ..., B_D\}$, confidence scores $\boldsymbol{s} = \{s_1, s_2, s_3, ..., s_D\}$, and predicted classes $\boldsymbol{c} = \{c_1, c_2, c_3, ..., c_D\}$. Additionally, we add a whole image bounding box for each "stuff" class to the object detection predictions, raising the total number of detections to $D + N_{st}$. Doing so allows the panoptic submodule to process "things" and "stuff" with a unified architecture.

Notably, the versatility of the panoptic submodule allows our network to work with or without object masks. When the object detection submodule has the capability to predict instance masks for "things" $\boldsymbol{M} = \{M_1, M_2, M_3, ..., M_D\}$, they are easily incorporated into the dynamic potential $\boldsymbol{\Psi}$. Details will be given in Sec. 5.3.4.1.

5.3.4 Panoptic segmentation submodule

This submodule serves as the mastermind of the pipeline. Receiving cues from the two prior submodules, the panoptic segmentation submodule combines them into a dynamic potential $\boldsymbol{\Psi}$ (Sec. 5.3.4.1) and revises it according to predicted pairwise instance affinities (Sec. 5.3.4.2), producing the final panoptic segmentation with the same logic in training and inference. This pipeline is illustrated in Fig. 5.3.

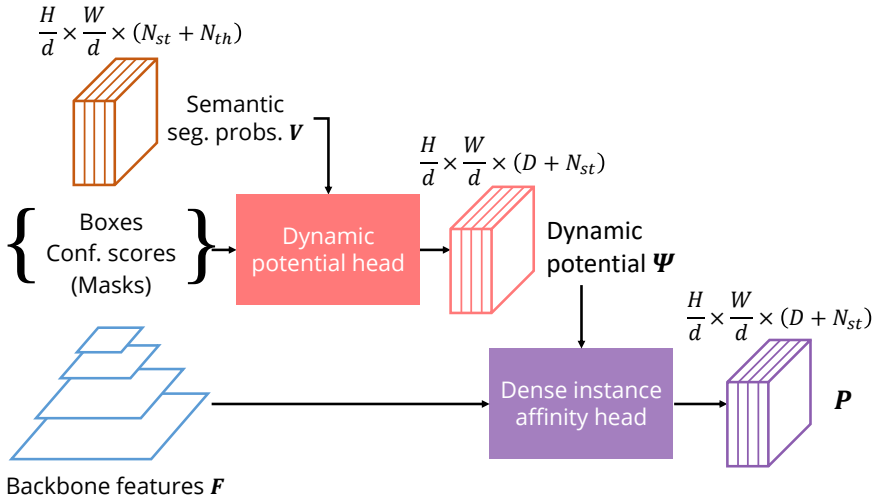


Figure 5.3: The panoptic segmentation submodule. Details on the dynamic potential head and dense instance affinity head are further clarified in Fig. 5.4 and 5.5 respectively.

5.3.4.1 Dynamic potential head

The dynamic potential head functions as an assembly node for segmentation and localisation cues from prior submodules. This head is capable of representing varying numbers of instances as it outputs a *dynamic* number of channels, one for each object instance or "stuff" class. We present three variants of dynamic head design, as illustrated in Fig. 5.4. Variant A is proposed in [157], whereas the mask-free parent of B and C is first described in [6] as the box consistency term. A major difference between variant A and the rest is the absence of detection score in A. We argue that leveraging detection scores can suppress false positives in the final output, as unconfident detections will be attenuated by its score. Thus, we will describe variant B and C in more details.

Given $(D+N_{st})$ bounding boxes $\boldsymbol{B}$ and box classes $\boldsymbol{c}$ (including the dummy full-image "stuff" boxes), it populates each box region with a combination of semantic segmentation probabilities $\boldsymbol{V}$ and box confidence scores $\boldsymbol{s}$ to produce a *dynamic potential* $\boldsymbol{\Psi}$ with $(D + N_{st})$ channels:

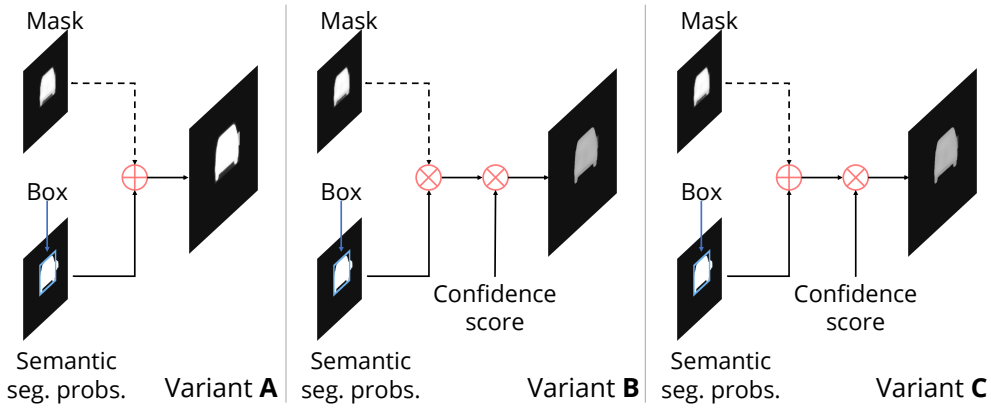


Figure 5.4: Three variants of the dynamic potential head. For clarity, we only show one instance in each diagram. In practice, the same operation is extended to all detections and "stuff". Note that the dotted path is only activated when masks are provided to the head. When no masks are given, variant B and C are equivalent.

$$\Psi_i(k) = \begin{cases} s_k V_i(c_k) & \text{for } i \in B_i \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Optionally, if provided with object masks $\boldsymbol{M}$, the dynamic potential head can also incorporate them into $\boldsymbol{\Psi}$. Defining $\boldsymbol{M}$ to be image-resolution instance masks where the raw masks have been resized to their actual dimensions and pasted to appropriate spatial locations in the image, the dynamic potential with object masks can be summarised as:

$$\Psi_i(k) = \begin{cases} s_k \left[V_i(c_k) \odot M_i(k) \right] & \text{for } i \in B_i \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

In variant B and C, operator $\odot$ is multiplication and summation respectively. More analysis of the variant B and C are included in Appendix 5.A.3.

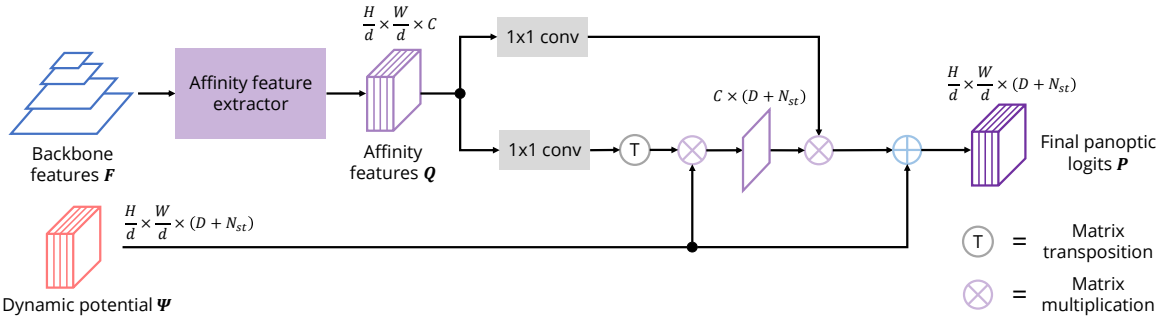


Figure 5.5: The dense instance affinity head. It is parametrised, expressive, lightweight, and fully differentiable.

5.3.4.2 Dense instance affinity head

We observe that the dynamic potential Ψ often carries conflicts and errors due to imperfect cues from semantic segmentation and object localisation. This motivates the design of this parametrised head, with the aim to enable a data-driven mechanism that resolves and revises the output of the dynamic potential head. The main difficulty with injecting parameters into an instance-level head is the varying number of instances across images, which practically translates to a dynamic number of channels in the input tensor. On the other hand, the fundamental building block of a convolutional neural network – convolution – is designed to handle a fixed number of input channels. This apparent incompatibility has led prior works on panoptic segmentation to use either no parameter at all [157], or only single scaling factors for entire tensors [86] providing limited modelling capacity.

This conundrum can be tackled by driving this head with a pairwise dense instance affinity, which is predicted from data, fully differentiable, and compatible with a dynamic number of input channels. By integrating global information according to the pairwise affinities, it produces the final panoptic segmentation probabilities, from which inference can be trivially made with an `argmax` operation along the channel dimension. Thus, it is amenable to a direct panoptic loss, an ingredient of an end-to-end network.

To construct the dense instance affinity, this head first extracts from the backbone features $\boldsymbol{F}$ a single feature tensor $\boldsymbol{Q}$ of dimension $\frac{H}{d} \times \frac{W}{d} \times C$, where C is the number of feature channels, and d is a downsampling factor. This corresponds to the affinity feature extractor in Fig. 5.5. The spatial dimensions of $\boldsymbol{Q}$ can be easily collapsed to produce an $\frac{HW}{d^2} \times C$ feature matrix.

Normally, the pairwise instance affinities $\boldsymbol{A}$ – a large $\frac{HW}{d^2} \times \frac{HW}{d^2}$ matrix – would then be produced by performing a matrix multiplication $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{Q}^T$. This would be followed by multiplying $\boldsymbol{A}$ with an $\frac{HW}{d^2} \times C'$ input tensor to complete the process. It is, however, prohibitively expensive due to the quadratic complexity with respect to HW. In a typical training step, where $(H, W) = (800, 1300)$ and $d = 4$, a single-precision matrix with the size of $\boldsymbol{A}$ would occupy 15.7GB of GPU memory, making this approach unpractical.

Drawing from the insight of [135], we design a lightweight pipeline for computing and applying the dense instance affinities (Fig. 5.5). Instead of sequentially computing $\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{\Psi}$ which explicitly produces $\boldsymbol{A}$, we compute $\boldsymbol{Q}\big(\boldsymbol{Q}^T\boldsymbol{\Psi}\big)$, since:

$$\big(\boldsymbol{Q}\boldsymbol{Q}^T\big)\boldsymbol{\Psi} = \boldsymbol{Q}\big(\boldsymbol{Q}^T\boldsymbol{\Psi}\big) \tag{5.3}$$

The result of $\boldsymbol{Q}^T\boldsymbol{\Psi}$ is a very small $C \times (D + N_{st})$ tensor, taking only tens of kilobytes. In terms of computation, using the same H, W, d as the example above and $(C, D, N_{st} = 128, 100, 53)$ as typically used in experiments, the efficient implementation reduces the total number of multiply-adds by 99.8% to 5 billion FLOPS. For reference, a ResNet-50-FPN backbone at the same input resolution requires 140 billion FLOPS.

Finally, we add the product back to the input, forming a residual connection to ease the learning task. As such, the full action of our dense instance affinity applier

can be summarised with the following expression:

$$\boldsymbol{P} = \boldsymbol{\Psi} + \phi_0(\boldsymbol{Q})\big(\phi_1(\boldsymbol{Q}^T)\boldsymbol{\Psi}\big) \tag{5.4}$$

where ϕ_0 and ϕ_1 are each a 1×1 convolution followed by an activation. From this formulation, inference is straight forward and does not require any post-processing, as an `argmax` operation on $\boldsymbol{P}$ along the channel direction readily produces the panoptic segmentation prediction.

Note that we do not compute a loss directly over $\boldsymbol{Q}$; instead, the instance affinities are implicitly trained by supervision from the panoptic matching loss described in the next section. In the preliminary experiments, we tried directly supervising $\boldsymbol{Q}$ with a contrastive loss, but did not observe performance gains. This shows that our end-to-end training scheme with the panoptic matching loss is already able to guide the model to learn effectively. A detailed discussion of the dense instance affinity operation, with ablation studies and visualisations, is provided in Sec. 5.4.1.

For simplicity, the affinity feature extractor adopts the same architecture as our semantic segmentation submodule. We use $C = 128$ in all experiments.

5.3.5 Panoptic matching loss

For instance-level segmentation, different permutations of the indices in the segmentation map are qualitatively equivalent, since the indices merely act to distinguish between each other, and do not carry actual semantic meanings.

During training, we feed predicted object detections into the panoptic segmentation submodule. As a result, the indices of the instances are not fixed or known beforehand. To compute loss, we first match the ground truth segmentation to the predicted detections by maximising the intersection over union between their bounding boxes (box IoU). Given a set of α ground truth segments $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, ..., \mathcal{T}_\alpha\}$,

and a set of β predicted bounding boxes $\boldsymbol{B} = \{B_1, B_2, B_3, ..., B_\beta\}$, we find the "matched" ground truth $\boldsymbol{\mathcal{T}}^\star$ which satisfies:

$$\boldsymbol{\mathcal{T}}^\star = \underset{\boldsymbol{\mathcal{Z}} \in \pi(\boldsymbol{\mathcal{T}})}{\arg\max} \, \mathrm{IoU}_t(\mathrm{box}(\boldsymbol{\mathcal{Z}}), \boldsymbol{B}) \tag{5.5}$$

where $\mathrm{box}(.)$ extracts tight bounding boxes from segments, $\pi(\boldsymbol{\mathcal{T}})$ refers to all permutations of $\boldsymbol{\mathcal{T}}$, and t sets the minimum match threshold for a match to qualify as valid. Note that the box IoU between different semantic classes are taken to be 0, and α and β need not be the same. Ground truth segments without matched predictions are set to the "ignore" label, and detections matching to the same ground truth segment are all removed except the top match, before being fed into the panoptic submodule. Both cases do not contribute any gradients. With the "matched" ground truth segmentation $\boldsymbol{\mathcal{T}}^\star$, we can compute the loss on the predicted panoptic segmentation probabilities $\boldsymbol{P}$ as per normal with a cross-entropy loss. Our experiments use 0.5 for t.

Unlike ours, the panoptic loss used by [157] does not have the matching stage and its panoptic head is trained with ground truth detections instead. As a result, the models of [157] are not trained to handle imperfect localisations. In addition, our loss differs from [102] as the loss used by their *spatial ranking module* does not directly supervise panoptic segmentation, does not take "stuff" into account, and thus does not globally optimise in an end-to-end way.

5.4 Experimental Evaluation

Cityscapes. The Cityscapes dataset features high-resolution road scenes with 11 "stuff" and 8 "thing" classes. There are 2,975 training images, 500 validation images, and 1,525 test images. We report on its validation set and test set.

COCO. The COCO panoptic dataset has a greater number of images and categories. It features 118k training images, 5k validation images, and 20k *test-dev* images. There are 133 semantic classes, including 53 "stuff" and 80 "thing" categories. We report on its validation set and *test-dev* set.

Evaluation metric. Our main evaluation metric is the panoptic quality (PQ), which is the product of segmentation quality (SQ) and recognition quality (RQ) [71]. SQ captures the average segmentation quality of matched segments, whereas RQ measures the ability of an algorithm to correctly detect objects.

We also report the mean Intersection over Union (IoU) score of our initial category-level segmentation $\boldsymbol{V}$, and the box Average Precision (AP_{box}) of our predicted bounding boxes $\boldsymbol{B}$. Additionally, for models which predict object instance masks $\boldsymbol{M}$ in the object detection submodule, we report its mask Average Preicision (AP_{mask}) as well. Both AP_{box} and AP_{mask} are averaged across IoU thresholds between 0.5 and 0.95, at increments of 0.05.

Cityscapes training. We follow most of the learning settings described in [70]. We distribute the 32 crops in a minibatch over 4 GPUs instead. The weights for the detection, semantic segmentation, and panoptic segmentation losses are set to 0.25, 1.0, and 1.0 respectively.

COCO training. We follow most of the learning settings for COCO experiments in [70]. For the learning schedule, we train for 200k iterations with a base learning rate of 0.02, and reduce it by a factor of 10 at 150k and 190k iterations. While this learning schedule differs from that used in [70], we found that our panoptic submodule with its additional parameters benefits from the new schedule. In terms of loss weights, we use 1.0, 0.2, and 0.1 for the object detection, semantic segmentation, and panoptic segmentation losses.

Model	Settings msk.	aff.	e2e.	heu.	amx.	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP box
$\mathbb{A}$					✓	54.6	46.0	60.9	77.9	68.4	75.0	–	36.9
$\mathbb{B}$		✓	✓		✓	59.0	50.2	65.3	80.1	72.4	77.8	–	38.1
$\mathbb{C}1$	✓			✓		59.3	51.4	65.0	79.8	73.2	78.1	33.8	38.1
$\mathbb{C}2$	✓				✓	59.6	52.4	64.8	80.4	72.9	78.1	33.8	38.1
$\mathbb{D}1$	✓	✓	✓	✓		60.6	52.4	66.5	80.4	74.2	79.5	33.7	38.8
$\mathbb{D}2$	✓	✓	✓		✓	61.4	54.7	66.3	81.1	74.7	79.5	33.7	38.8

Table 5.1: Ablation studies on Cityscapes validation set. Settings include two architecture variations: whether to utilise object masks (msk.), and whether to utilise the proposed instance affinity (aff.); one training option: whether to train end-to-end with the panoptic matching loss (e2e.); and two inference strategies: whether to directly take `argmax` (amx.) of the panoptic logits (which is either $\boldsymbol{\Psi}$ for $\mathbb{A}$ and $\mathbb{C}2$, or $\boldsymbol{P}$ for $\mathbb{B}$ and $\mathbb{D}2$) or use the heuristic merging strategy [70] (heu.).

5.4.1 Ablation studies

We conduct detailed ablation studies for five different settings, including two architecture choices (msk. and aff.), one training strategy (e2e.), and two inference options (heu. and amx.). We report the results in Table 5.1. Explanations for the abbreviations can be found in the table caption. For clarity, we provide a brief description of the ablation models:

- Model $\mathbb{A}$ uses a Faster-RCNN head as its object detection submodule, and has neither the dense instance affinity head nor the panoptic matching loss. The dynamic potential $\boldsymbol{\Psi}$ is used as the final output $\boldsymbol{P}$.

- Model $\mathbb{B}$ differs from $\mathbb{A}$ by employing the dense instance affinity head and the panoptic matching loss.

- In $\mathbb{C}1$ and $\mathbb{C}2$, the model uses a Mask-RCNN head as its object detection submodule, and has neither the dense instance affinity head nor the panoptic matching loss. During inference, $\mathbb{C}1$ merges the semantic and instance segmentation predictions using heuristics [71], whereas $\mathbb{C}2$ outputs the dynamic potential $\boldsymbol{\Psi}$ as $\boldsymbol{P}$.

- The pair ($\mathbb{D}1$, $\mathbb{D}2$) differs from ($\mathbb{C}1$, $\mathbb{C}2$) by employing the instance affinity and the panoptic matching loss.

Note that model $\mathbb{A}$ and $\mathbb{B}$ do not produce nor use object mask predictions, and are therefore not possible to test with the heuristic merger strategy [70]. In addition, the pair $\mathbb{C}1$ and $\mathbb{C}2$, as well as $\mathbb{D}1$ and $\mathbb{D}2$, are identical models using different inference methods.

Dense instance affinity. Comparing across model $\mathbb{A}$ and $\mathbb{B}$, it is evident that training and testing with the proposed dense instance affinity leads to significant performance boosts. Increased performances are seen across all metrics, with the largest rises in PQ (+4.4 for all, +4.2 for "things" and +4.4 for "stuff") and RQ (+4.0). This testifies to the effectiveness of the dense instance affinity, even with only box predictions. A similar trend is also evident with object masks enabled, between model $\mathbb{C}2$ with $\mathbb{D}2$, recording a 1.8 rise in overall PQ. Fig. 5.6 visualises some examples of instance affinities, with more in Appendix 5.E.

End-to-end training with panoptic matching loss. While $\mathbb{C}1$ and $\mathbb{D}1$ are trained differently – with the former being trained jointly, and the latter being trained end-to-end with the panoptic matching loss – they are tested using the same heuristic strategy [70]. Therefore, the 1.3 increase in PQ of $\mathbb{D}1$ over $\mathbb{C}1$ solely stems from the fact that $\mathbb{D}1$ undergoes end-to-end training, and shows that our end-to-end training strategy with the panoptic matching loss is effective.

Unified training and inference pipeline. For $\mathbb{D}1$, we test a model trained end-to-end with the panoptic matching loss using the heuristic merger strategies. In contrast, for $\mathbb{D}2$, we take the same model and take `argmax` from the final panoptic logits. We can see that the $\mathbb{D}2$ still outperforms $\mathbb{D}1$ by 0.8 PQ, giving proof for the benefit of having a unified training and testing pipeline.

5.4.2 Comparison with state-of-the-art

Cityscapes. We compare our results with other methods on Cityscapes validation set in Table 5.2. All entries are ResNet-50 [60] based except [86, 160]. We sort prior works into two tracks, depending on whether the network performs instance segmentation internally. For both tracks, our method achieves the state-of-art. The most telling comparison is between our model and UPSNet, as these methods have a similar network architecture other than our proposed panoptic segmentation submodule. Our network is able to outperform UPSNet by 2.1 PQ. On the other hand, among methods that do not rely on instance segmentation [86, 160], our system outperforms the previous state-of-art by 3.5 PQ, even though they utilise stronger backbones (Xception-71 [28] and ResNet-101 [60]) than ours (ResNet-50).

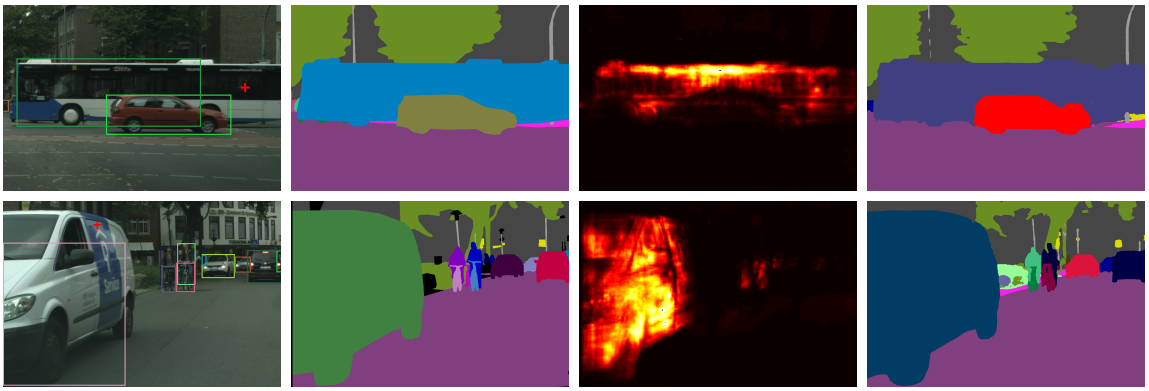
Speed-wise, our design compares favourably with other state-of-the-art models. On Cityscapes, inference takes 386ms[1] and 201ms[2] per image for [70] and [157], whereas our full model runs at 197ms per image. All models are ResNet-50 based and timed on a single RTX 2080Ti card.

COCO. Results on the COCO panoptic validation set are reported in Table 5.3. Due to the disentangling power of our proposed pipeline and unified train-test logic, we are able to outperform the previous state-of-art method by 0.9 in terms of overall PQ, and 2.1 in terms of PQ for "stuff".

Results on the Cityscapes test set and COCO *test-dev* set are reported in Table 5.4 and 5.5. We perform *single-scale* inference, without any test-time augmentation. For a fair comparison, only methods that are ResNe(X)t-based are reported. Our method achieves the state-of-art performance on both datasets with a PQ of 63.3 and 47.2 respectively.

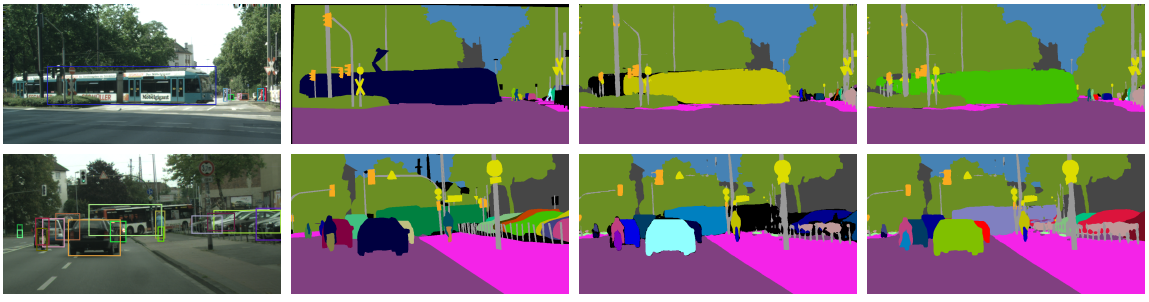
Qualitative results are shown in Fig 5.7 where we compare with our re-imple-

[1]Obtained by running our re-implementation.
[2]Obtained by running its publicly released code.



(a) Image with boxes (b) Ground truth (c) Instance affinities (d) Panoptic prediction

Figure 5.6: Examples of predicted instance affinities. The instance affinities shown in (c) are for the cross-marked pixels in (a). Observe that the predicted bounding boxes (shown in (a)) for the bus in Row 1 and the frontal car in the Row 2 fail to enclose the full object. Rule-based fusion in [71, 70] cannot recover from such localisation errors as their segments are constrained to pixels inside bounding boxes. In contrast, our model is able to still segment full objects by predicting strong affinities between the marked locations with the rest of the instance.



(a) Image with boxes (b) Ground truth (c) Heuristic merger [70] (d) Our prediction

Figure 5.7: Qualitative results on Cityscapes Dataset. The input images are shown with the predicted bounding boxes overlaid above. In column (c), swathes of "void" region are clearly visible for pixels where assignment cannot be made by heuristics. In contrast, our panoptic segmentation results are robust to incoherence in segmentation and localisation cues, and can explain more pixels in an image.

Method	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP bbox
Li *et al*. [86]	53.8	42.5	62.1	–	–	**79.8**	–	–
DeeperLab [160]	56.5	–	–	–	–	–	–	–
SSAP [48]	58.4	**50.6**	–	–	–	–	–	–
Ours (w/o mask)	**59.0**	50.2	**65.3**	**80.1**	**72.4**	77.8	–	**38.1**
TASCNet [83]†	55.9	50.5	59.8	–	–	–	–	–
Attention [87]†	56.4	52.7	59.0	–	–	73.6	33.6	–
Pan. FPN [70]†	57.7	51.6	62.2	–	–	75.0	32.0	–
UPSNet [157]†	59.3	54.6	62.7	79.7	73.0	75.2	33.3	**39.1**
Panoptic Deeplab [26]†	59.7	–	–	–	–	–	–	–
Seamless [125]†	60.3	**56.1**	63.3	–	–	77.5	33.6	–
Ours (w/ mask)†	**61.4**	54.7	**66.3**	**81.1**	**74.7**	**79.5**	**33.7**	38.8

Table 5.2: Panoptic segmentation results on Cityscapes validation set. Models that run instance segmentation internally are marked with †. Other than [86, 160], all works are ResNet-50 [60] based. For fairness, we only include numbers obtained via *single-scale* inference.

Method	PQ all	PQ th.	PQ st.	SQ all	RQ all	IoU all	AP mask	AP bbox
JSIS-Net [35]	26.9	29.3	23.3	72.4	35.7	–	–	–
Panoptic Deeplab [26]	35.1	–	–	–	–	–	–	–
Panoptic FPN [70]	39.0	45.9	28.7	–	–	–	33.3	–
UPSNet [157]	42.5	**48.6**	33.4	78.0	52.5	**54.3**	34.3	37.8
Ours (w/ mask)	**43.4**	**48.6**	**35.5**	**79.6**	**53.0**	53.7	**36.4**	**40.5**

Table 5.3: Panoptic segmentation results on COCO 2017 validation set. All methods are based on a ResNet-50 backbone.

mentation of Panoptic FPN. As the instance affinity operation integrates information from pixels locally and globally, our method can resolve errors in the detection stage by propagating meaningful information from other pixels. The "void" region (displayed in black) shown in Fig 5.7c is typically present in results produced by the heuristic merging process popularised by [71]. They are due to the method's inability to resolve inconsistencies between semantic and instance predictions. In contrast, our method successfully handles such cases, as evident in Fig. 5.7d.

Method	Bb.	PQ all	PQ th.	PQ st.	SQ all	SQ th.	SQ st.	RQ all	RQ th.	RQ st.
P. Deeplab [26]	R-50	58.0	–	–	–	–	–	–	–	–
Ours (w/ mask)	R-50	**61.0**	**52.7**	**67.1**	**81.4**	**79.6**	**82.8**	**73.9**	**66.2**	**79.6**
Li *et al*. [86, 6]	R-101	55.4	44.0	63.6	79.7	77.3	81.5	68.1	57.0	76.1
SSAP [48]	R-101	58.9	48.4	66.5	**82.4**	**82.9**	82.0	70.6	58.3	79.6
TASCNet [83]†	X-101	60.7	53.4	66.0	81.0	79.7	82.0	73.8	67.0	78.8
Ours (w/ mask)†	R-101	**63.3**	**56.0**	**68.5**	**82.4**	81.0	**83.4**	**75.9**	**69.1**	**80.9**

Table 5.4: Panoptic segmentation performance on the Cityscapes test set. Models pretrained on the COCO dataset are marked with †. Bb.: backbone, R: ResNet, X: ResNeXt.

Method	Bb.	PQ all	PQ th.	PQ st.	SQ all	SQ th.	SQ st.	RQ all	RQ th.	RQ st.
JSIS-Net [35]	R-50	27.2	29.6	23.4	71.9	71.6	72.3	35.9	39.4	30.6
P. Deeplab [26]	R-50	35.2	–	–	–	–	–	–	–	–
SSAP [48]	R-50	36.9	40.1	32.0	**80.7**	**81.6**	**79.4**	44.8	48.5	39.3
TASCNet [83]	R-50	40.7	47.0	31.0	78.5	80.6	75.3	50.1	57.1	39.6
Ours (w/ mask)	R-50	**43.6**	**48.9**	**35.6**	80.1	81.3	78.3	**53.3**	**59.5**	**44.0**
Attention [87]	X-152	46.5	**55.9**	32.5	81.0	**83.7**	77.0	56.1	**66.3**	40.7
UPSNet [157]	R-101	46.6	53.2	36.7	80.5	81.5	78.9	56.9	64.6	45.3
Ours (w/ mask)	R-101	**47.2**	53.5	**37.7**	**81.1**	82.3	**79.2**	**57.2**	64.3	**46.3**

Table 5.5: Panoptic segmentation performance on the COCO *test-dev* set. Bb.: backbone, R: ResNet, X: ResNeXt.

5.5 Conclusion

We have presented an end-to-end panoptic segmentation approach that exploits a novel pairwise instance affinity operation. It is lightweight, learnt from data, and capable of modelling a dynamic number of instances. By integrating information across the image in a differentiable manner, the instance affinity operation with the panoptic matching loss enables end-to-end training and heuristics-free inference, leading to improved qualities for panoptic segmentation. Furthermore, our method bestows additional flexibility upon network design, allowing our model to perform well even if it only uses bounding boxes as localisation cues.

Acknowledgements This work was supported by Huawei Technologies Co., Ltd., the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1. We would also like to thank the Royal Academy of Engineering and FiveAI.

Appendices

We first provide further details in terms of architecture and design in Sec. 5.A. Furthermore, we explain the implementation details for training and inference on the Cityscapes and COCO datasets in Sec. 5.B. This is followed by an empirical study in Sec. 5.C on a number of evaluation metrics for "stuff" classes, where we raise questions on the suitability of using the PQ metric for "stuff". Lastly, we provide further quantitative and qualitative results on the Cityscapes and COCO datasets in Sec. 5.D, 5.E, and 5.F.

5.A Architecture and Design

This section presents design details of some components of our proposed network, including the semantic segmentation submodule (Sec. 5.A.1), object detection submodule (Sec. 5.A.2), and dynamic potential head (Sec. 5.A.3). Empirical evidence is also provided in Sec. 5.A.4 that show benefits of training the panoptic segmentation submodule with predicted detections instead of ground truth ones.

5.A.1 Semantic segmentation submodule

Our semantic segmentation submodule is modified from [157], by performing Group Normalisation [155] after each 3×3 convolution. We illustrate the pipeline in Fig. 5.8. Note that the architecture of the *feature decoder* inside this submodule is also adopted by our dense instance affinity head to extract affinity features $\boldsymbol{Q}$. This submodule is supervised by a cross-entropy loss, unless otherwise stated.

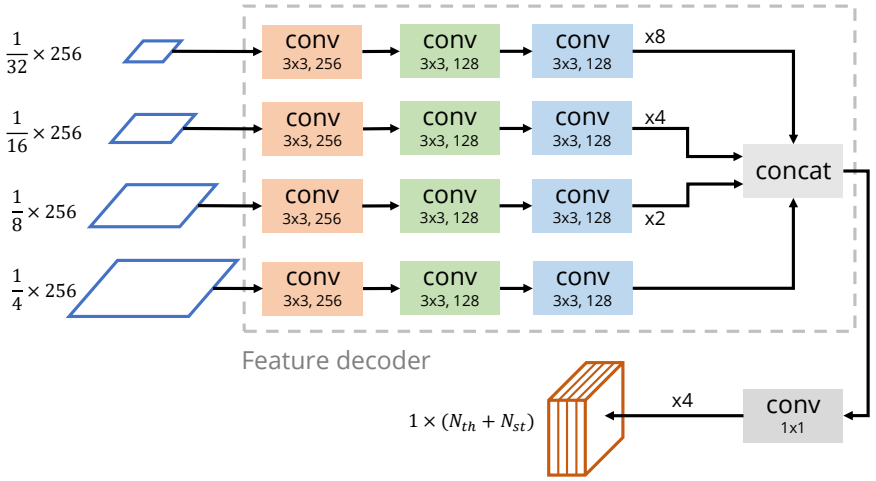


Figure 5.8: Semantic segmentation submodule. Each 3×3 convolution block consists of a deformable convolution (with the indicated number of output channels), a Group Normalisation operation, and a ReLU activation. Weights are **shared** across 3×3 convolution blocks with the same colour code.

	Variant B			Variant C		
Dataset	PQ	SQ	RQ	PQ	SQ	RQ
Cityscapes	**61.4**	**81.8**	**74.7**	60.3	80.8	73.5
COCO	42.7	79.4	52.2	**43.4**	**79.6**	**53.0**

Table 5.6: Ablation study on two design variants for the dynamic potential head. On Cityscapes, variant B outperforms variant C, whereas on COCO, variant C achieves higher accuracies.

5.A.2 Object detection submodule

In our experiments, we use the standard box head from Faster-RCNN [127] and optionally the mask head from Mask-RCNN [59] for this submodule, following [157, 70]. For the mask head, we use the Lovasz Hinge loss to replace the binary cross-entropy loss. Thanks to the modular design of our network, it is easy to substitute it with any other detector architecture.

5.A.3 Dynamic potential head

We refer to the design variant B and C presented in Sec. 5.3.4.1 (Fig. 5.4). At first glance, variant B, which multiplies semantic segmentation probabilities $V_i(c_k)$

		Classified as	
		th.	st.
GT	th.	95.1	4.9
	st.	0.0	100.0

(1) Cityscapes

		Classified as	
		th.	st.
GT	th.	90.1	9.9
	st.	4.8	95.2

(2) COCO

Table 5.7: Confusion matrices between "thing" and "stuff" for semantic segmentation submodule outputs on Cityscapes and COCO validation sets. All numbers are percentages, normalised row-wise.

with mask scores $M_i(k)$, appears to be a more appropriate method than variant C which sums probabilities instead. The output of variant B is high only when both inputs are unanimously high. This can filter out spurious misclassifications from either input, and improve robustness towards false positive predictions. Indeed, on Cityscapes, we observe that variant B achieves a 1.1 PQ lead over the variant C counterpart (first row of Table 5.6).

However, on COCO, we notice a high tendency for the semantic segmentation submodule to mistake "things" for "stuff" (Table. 5.72). The multiplicative action of variant B can systematically and substantially weaken the panoptic logits for "thing" classes, relative to the unattenuated panoptic logits of "stuff" classes. This can be undesirable for models whose semantic segmentation submodule is already prone to misclassifying "things" as "stuff". On the other hand, the opposite is true for variant C, as summation strengthens panoptic logits of "things" in comparison to unmodified "stuff" scores. This led us to use variant C for COCO, and we observe a 0.7 PQ improvement in comparison to B (second row of Table 5.6).

5.A.4 Training with predicted detections

In contrast with the practice in [157], we argue that, during training, the dynamic potential head should use predicted detections instead of ground truth ones to construct its output $\mathbf{\Psi}$. This allows the network to learn from realistic examples, and

Dets. for training	PQ	SQ	RQ	IoU	AP_{box}
Ground truths	58.6	80.0	72.0	77.8	36.8
Predictions	**59.0**	**80.1**	**72.4**	77.8	**38.1**

Table 5.8: Comparison of two different training strategies. In the *top row*, ground truth detections are used to train the panoptic segmentation submodule, whereas in the *bottom row* those predicted by the network *on-the-fly* are used instead. Results are reported on the Cityscapes validation set.

build up its robustness towards imperfections in detection localisation and scoring. To test our hypothesis, we carried out an ablation study on Cityscapes using our mask-free model. When training with ground truth boxes, a uniform score of 1.0 is used for their confidence scores. Results are shown in Table 5.8. As expected, training with predicted detections yields performance improvements across all panoptic metrics, including a 0.4 increase in PQ. A large boost in observed for AP_{box} ($+1.3$), because training with predicted boxes allows gradients from the panoptic segmentation submodule to flow to the object detection submodule, giving it more fine-grained supervision. IoU has not changed, as this ablation setting does not affect the semantic segmentation module.

5.B Implementation Details

Cityscapes training. We run our experiments on four V100-32GB GPUs. This allows us to load each GPU with eight image crops and obtain an effective batch size of 32. The large number of crops per GPU enables us to use a Lovasz Softmax loss [12] instead of a cross-entropy loss for supervising semantic segmentation, which we found to be effective. Following [70], we use a base learning rate of 0.01, a weight decay of 0.0001, and train for a total of 65k iterations. The learning rate is reduced by 10 folds after the first 40k iterations, and once more after 15k more iterations. Additionally, we adopt a "warm-up" period at the start of training –

linearly increasing the learning rate from a third of the base rate to the full rate in 500 iterations, which helps stabilise the training.

We augment input images on-the-fly during training to reduce the network's tendency to overfit. Our augmentation pipeline resizes the input image by a random factor between 0.5 and 2, takes a random 512×1024 crop, and applies a horizontal flip with 50% chance. On top of these techniques, we also apply image relighting, randomly adjusting the brightness, contrast, hue, and saturation of the image by a small amount, as used in [70].

COCO training. On COCO, as the dataset is larger than Cityscapes, less overfitting is observed. Therefore, in terms of data augmentation techniques, we only apply resizing where the shorter size is resized to 800 and the longer size is kept under 1333, and random horizontal flipping with 0.5 probability.

Miscellaneous. We use ImageNet pretrained ResNet-50 to initialise all experiments. The batch normalisation statistics are kept unchanged, though further performance gains are likely if they are finetuned on the target dataset. When a normalisation step is used in either the semantic or panoptic submodules, we use the Group Normalisation operation [155], as it is less sensitive to batch sizes.

Inference. We conduct single-scale inference for all experiments, letting the network process and make predictions on full-resolution images in a single forward pass. Note that only detection predictions whose confidence scores are more than a threshold are fed into the dynamic potential head during inference, to minimise unnecessary computation. This cut-off is 0.5 and 0.75 for Cityscapes and COCO respectively.

5.C Evaluation of "Stuff"

The PQ metrics effectively treats "stuff" classes as image-wide instances – making all "stuff" segments undergo the same matching procedure with ground truth segments as "thing" segments. While this approach has its merits including a unified evaluation logic and a simplified PQ implementation, it should be noted that matching "stuff" predictions to ground truth is not strictly necessary, since at most one "stuff" instance for each "stuff" class is present per image.

Furthermore, this approach towards "stuff" is neither robust nor fair as a measure for "stuff" segmentation quality, and arguably encourages post-processing of panoptic predictions. Under the PQ formulation, misclassifying even a single pixel into a "stuff' class absent in the ground truth will increment false positive detections by one, and such mistakes – exacerbated by the relatively small number of ground truth "stuff" segments in a dataset – attract a large penalty on the "stuff" RQ, even though the practical impact on perceptual quality is minimal. This also contrasts in spirit with the mean IoU metric widely adopted to measure semantic segmentation quality, as the mean IoU accumulates intersection and union counts over the whole dataset and is minimally affected by individual pixels.

On the other hand, CNN-based semantic segmentation models are typically prone to produce spurious misclassifications, as they usually do not explicitly enforce smoothness. As a result, recent panoptic segmentation works [71, 87, 83, 70, 157, 160] collectively resort to setting small "stuff" segments to "void" in the final panoptic segmentation. Therefore, to foster meaningful comparison with other state-of-the-art panoptic segmentation approaches, unless specified otherwise, we also carry out this strategy as part of the evaluation.

Effects of trimming small stuff segments on evaluation metrics. On Cityscapes validation set, we test our full model, our re-implemented Panoptic FPN [70], and

Table 5.9: Comparison of various evaluation metrics for "stuff", before and after small stuff areas are set to "void" on Cityscapes validation set. Note that the IoU^{st} here is computed from the final panoptic segmentation, by combining instances of the same semantic class. This is different from the IoU metrics reported in Table 5.1 and 5.2, which measure the quality of the semantic segmentation input to the heuristic merger / our panoptic segmentation submodule.

Model	Trim stuff	PQ^{st}	SQ^{st}	RQ^{st}	IoU^{st}
Panoptic FPN [70]*		59.9	79.3	72.9	74.7
Panoptic FPN [70]*	✓	62.0	79.6	75.5	74.5
		+2.1	+0.3	+2.6	-0.2
UPSNet [157]†		60.5	79.8	73.6	75.8
UPSNet [157]†	✓	62.8	80.0	76.3	75.7
		+2.3	+0.2	+2.7	-0.1
Ours		64.2	81.4	77.1	78.3
Ours	✓	66.3	81.8	79.4	78.2
		+2.1	+0.4	+2.3	-0.1

* Results obtained from our re-implementation of Panoptic FPN.
† Results obtained by running the public inference script of [157].

the released UPSNet model [157] with and without trimming off small "stuff" regions, to quantitatively assess the impact of this step on state-of-the-art models. The findings are reported in Table 5.9.

The results show that PQ and RQ are very sensitive to such operations, as removing small stuff segments consistently results in an increase of approximately 2 points for "stuff" PQ, and 2.5 points for "stuff" RQ. This can be largely attributed to the reduced number of false positive stuff segments. On the other hand, the "stuff" IoU metric is insensitive to such modifications, as in all three cases, it suffers a slight decrease of 0.1 or 0.2 points. This prompts us to believe that "stuff" IoU is a better metric for capturing "stuff" segmentation quality than the "thing"-centric PQ family.

Dataset	Method	PQ all	PQ th.	PQ st.	SQ all	SQ th.	SQ st.	RQ all	RQ th.	RQ st.	IoU all	IoU th.	IoU st.	AP mask	AP box
Cityscapes	Ours (w/o mask)	59.0	50.2	65.3	80.1	78.4	81.2	72.4	63.9	78.6	77.8	78.7	77.2	–	38.1
Cityscapes	Ours (w/ mask)	61.4	54.7	66.3	81.1	80.0	81.8	74.7	68.2	79.4	79.5	81.0	78.4	33.7	38.8
COCO	Ours (w/ mask)	43.4	48.6	35.5	79.6	80.0	78.9	53.0	59.2	43.8	53.7	60.4	43.6	36.4	40.5

Table 5.10: Full panoptic segmentation results on Cityscapes validation set and COCO validation set. All models are ResNet-50 based, and tested with a *single-scale* inference scheme, without test-time augmentation.

5.D Detailed Validation Set Results

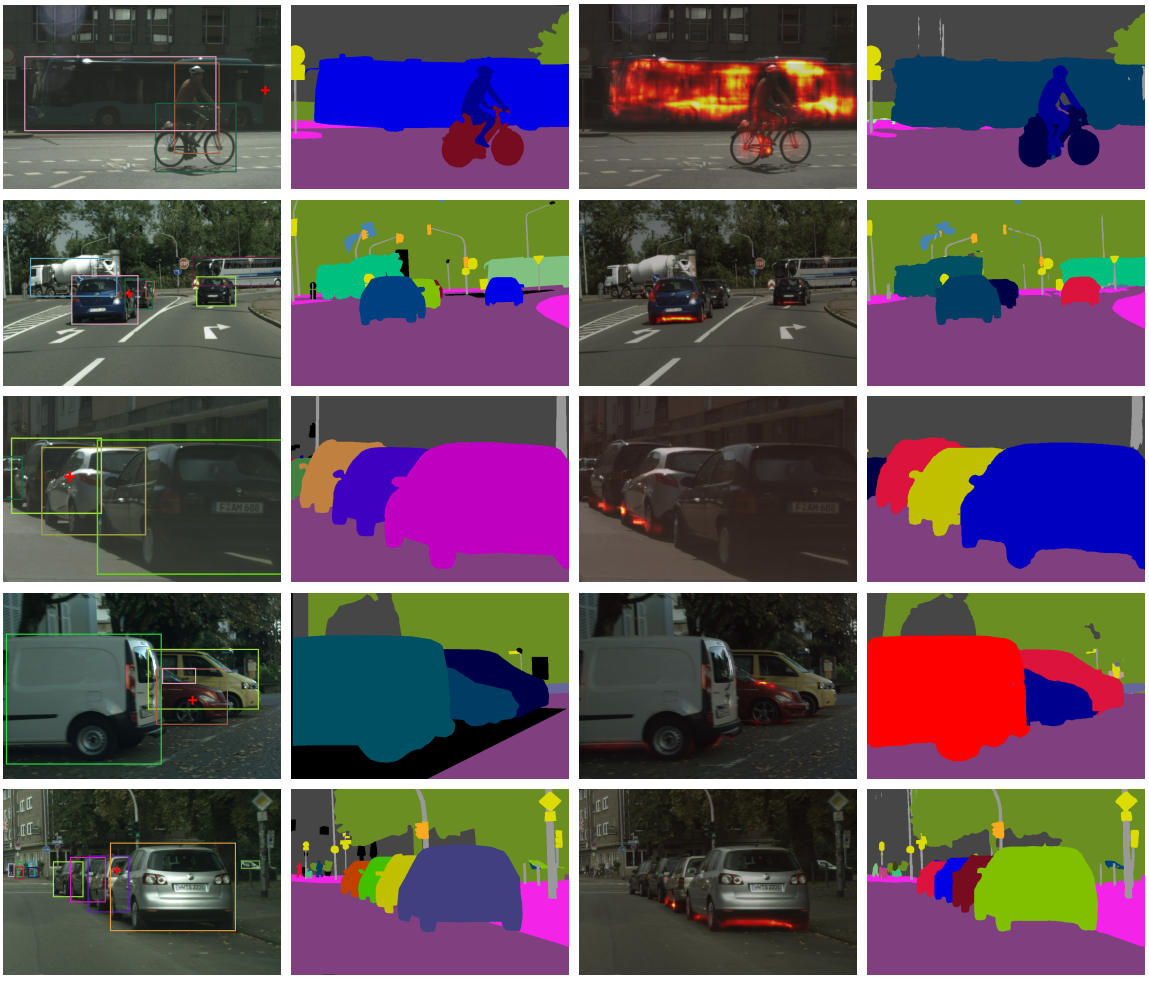
We report the detailed results of our models on the Cityscapes and COCO validation sets in Table 5.10. In addition to the metrics reported in Sec. 5.4, this table also includes breakdowns of SQ and RQ by "stuff" and "thing".

5.E Visualisation of Learnt Instance Affinities

Additional visualisations of some predicted instance affinities are provided in Fig. 5.9. Note that these instance affinities are extracted from our mask-free model. Interestingly, the model has learnt to resolve cars regions covered by multiple car bounding boxes – a problem difficult for methods only using boxes as localisation cues – by creating strong instance affinities to the bottoms and tyres of cars. The model has found that these regions of cars are normally not covered by multiple bounding boxes, and therefore it is most helpful for instance discrimination by associating uncertain pixels with these regions.

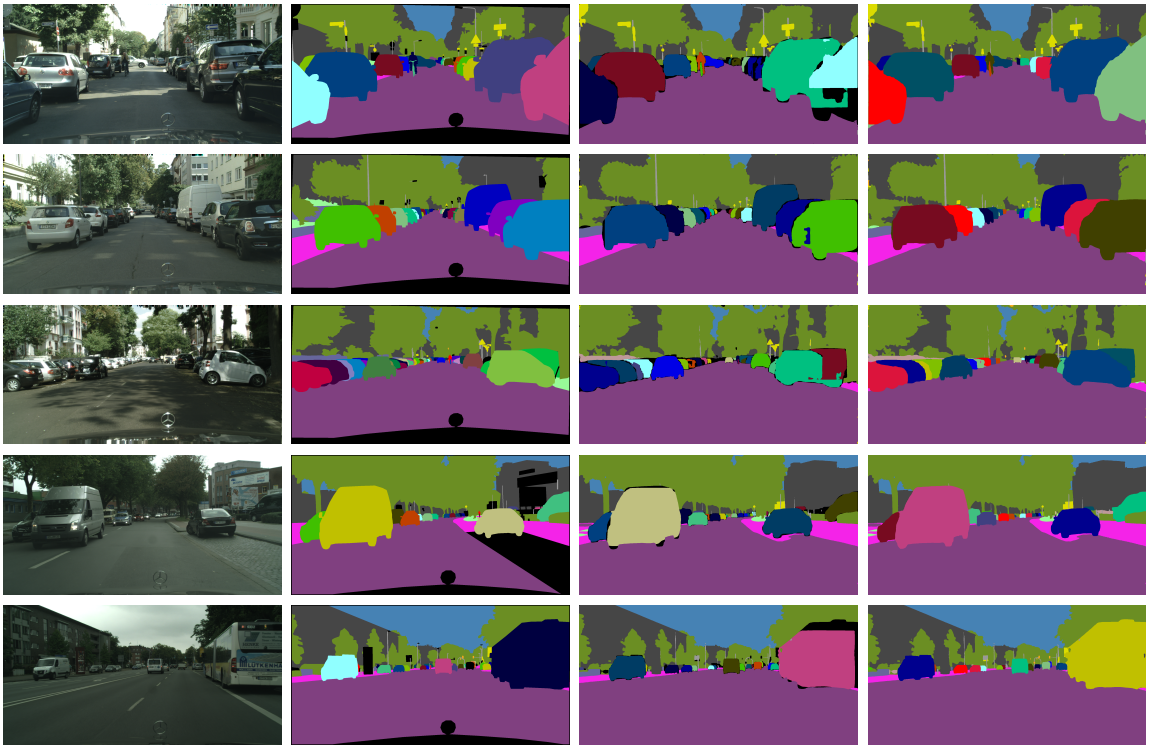
5.F Qualitative Results

We show more qualitative results in Fig. 5.10 and 5.11, and comparisons to previous state-of-the-art methods [70, 157].



(a) Image with boxes (b) Ground truth (c) Instance affinities (d) Panoptic prediction

Figure 5.9: Additional examples of instance affinities. In (c), we show the instance affinities – overlaid on input images to aid visualisation – of the cross-marked pixels in (a). These affinities and predictions are predicted by our mask-free models which use only bounding boxes. They can be seen to help segment full objects when bounding box localisation is poor (Row 1), and attribute pixels within multiple bounding boxes to the correct instances (Row 2 to 5). For Row 4, our proposed method is able to overcome a false positive detection, as the dynamic potential is robust towards false detections. For Row 5, the cross-marked pixel is on the wing mirror of the closest silver car, and our fine-grained instance affinity is able to attribute the mirror to the correct car, while the ground truth has failed to correctly label as such.



(a) Image (b) Ground truth (c) Heuristic Fusion [71] (d) Ours

Figure 5.10: Qualitative results on the Cityscapes dataset. Column (c) and (d) are produced by the same model under different inference strategies – either by heuristic merger [71] or with our proposed panoptic segmentation submodule. The first three rows show that our model is able to revise erroneous cues and resolve conflicts between overlapping object masks. The final two rows demonstrate our network's ability to segment outside boxes, when boxes do not cover the full extent of an object.

(a) Image (b) Ground truth (c) UPSNet [157] (d) Ours

Figure 5.11: Qualitative results on the COCO dataset. Column (c) is produced by running the publicly available inference script of [157]. With our parametrised panoptic segmentation submodule, we are able to produce more coherent, accurate, and visually appealing predictions than the parameter-free approach of [157].

Chapter 6

Conclusion

In this integrated thesis, we have focused on advancing our capability to perform instance-level scene parsing in three ways. First, in a step towards fine-grained hierarchical understanding, we extend the task of human parsing – traditionally formulated as a category-level segmentation problem – to the instance level, and solve it by proposing a method which produces a rich hierarchy of information in a single forward pass. To our best knowledge, it is the first work to tackle instance-level human parsing. While there has been steady progress in the field of instance-level segmentation, we have noticed that most methods have heavily relied upon costly pixel-level ground truth masks, the scarcity of which have hampered the development and application of such algorithms. To reduce this reliance, we introduce a weakly- and semi-supervised training strategy using only image-level tags and bounding boxes. Furthermore, realising that the CRF-based instance segmentation approach used in Chapter 3 and 4 suffers from a number of important shortcomings, We develop the next generation instance-level segmentation pipeline based on a densely connected instance affinity, which is parametrised, fully learnt from data, and takes advantage of a lightweight implementation. The overall model is efficient, trained end-to-end, and produces state-of-the-art results on several

public panoptic segmentation leaderboards at the time of publication.

In the rest of this Chapter, we will summarise the contributions from each of the previous three chapters, and finish with a brief discussion of remaining challenges, open questions, and future directions.

6.1 Discussion of Contributions

Chapter 3: Holistic, Instance-level Human Parsing

The task of human parsing has attracted a considerable level of research interest due to its important role in enabling robot-human interactions, virtual reality, action recognition, person re-identification, *etc*. Before the publication of this work, human parsing has been approached in the literature as a semantic segmentation problem, where the expectation is to find all human pixels in an image and assign each of them to one of the body parts. However, for many of the aforementioned applications, precise instance-level differentiation between human subjects becomes necessary once there are multiple people in a scene.

To bridge this gap, we propose what to our best knowledge is the first instance-level human parsing network. Given detections, our deep neural network is trained end-to-end, and produces category-, instance-, and part-level segmentation of humans in an image with just a single forward pass. At the time of this work, the prevalent approach for instance-level segmentation is detection-based. Compared to detection-based methods, our approach considers all segments in an image jointly and predicts non-overlapping segmentation by design. We present the first instance-level results on the Pascal Person Parts dataset and outperform the state-of-art detection-based method [31] which we re-implement for the task of instance-level human parsing.

Our work has demonstrated the potential of deep neural networks to perform

accurate instance-level human parsing. Since the publication of our original conference paper, this topic has seen a rising level of interest from the research community, in terms of both the amount of research carried out, and the number of new datasets for this task [51, 175, 91, 114].

This pipeline can be trivially adapted to parse any object. However, a major limitation of the architecture is that it can only handle one object category per model. Using a number of individually trained models, where each model targets one object category, can technically achieve multi-object instance-level parsing, but suffers from inefficiency, and a lack of global reasoning for all objects in an image. It is thus desirable that parsing and understanding multiple types of objects can be achieved with a single model. To the best of our knowledge, as of the time of writing, multi-object instance-level parsing is still largely unstudied.

Chapter 4: Weakly- and Semi-Supervised Panoptic Segmentation

This work has been motivated by the fact that most instance-level segmentation methods are trained with full supervision in the form of densely labelled segmentation ground truth masks. As every pixel has to be manually labelled, it is extremely time-consuming and financially costly to annotate such a dataset. Often, budget-constrained annotators are faced with a difficult choice between quantity and quality.

We propose to train instance-level segmentation models with weaker labels, consisting of image-level tags – binary indicators for presence or absence of objects – for "stuff" classes, and bounding boxes for "thing" instances. We demonstrate that based on the principles of self-training and Expectation Maximisation (EM), the instance-level ground truth can be iteratively refined, and in turn network performance improved. With this weak form of supervision, we are able to achieve

up to a 30 times reduction in annotation time on the Cityscapes dataset, while still obtaining up to 95% of fully supervised performance.

To validate our training strategy on a recently popularised task – panoptic segmentation, we modify the instance-level segmentation used in the previous chapter to perform panoptic segmentation. Specifically, its semantic segmentation submodule is extended to predict for both "thing" and "stuff" classes. Furthermore, in addition to the detection predictions for "thing" objects, we create dummy image-wide boxes for each of the "stuff" classes, thus enabling the box term to process both instances and non-instances with the same architecture. We then train this network with the proposed weak labels and report satisfactory results in comparison to our fully-supervised baseline. As of the time of writing, it is still the only weakly-supervised method for panoptic segmentation.

Nevertheless, this weakly-supervised training approach has room for improvement. The multi-label classification network, which extracts coarse segmentation masks from image-level tags, assumes that the distribution of class labels in a dataset is reasonably balanced and independent. This assumption is no longer valid if there are classes present in all training images (*e.g.*, the "floor" in an indoor scene dataset), or groups of classes that always appear together (*e.g.*, the "traffic pole" and "traffic light" in a driving scene dataset). Furthermore, using bounding boxes for "things" might be suboptimal as their instance-level information is greatly diminished when boxes of the same object class overlap (*e.g.*, pedestrians in a crowded street), and its semantic information can also be corrupted by interfering objects that always appear together with the object of interest (*e.g.*, the shadow under cars, road around pedestrians' legs). As a potential future direction, one can look into using scribbles [97] as the unified supervision for both "thing" and "stuff" classes. They are fast (and hence economical) to annotate. Compared to image-level tags, they contain some location cues, whereas compared to bounding boxes,

they offer additional shape and occlusion information, which can help handle the challenging scenarios described above.

There are some additional shortcomings related to the architecture of the panoptic segmentation network used in this work. It requires an external detector that is trained separately from the main network, hurting efficiency and potentially performance. Furthermore, its Instance CRF layer suffers from a paucity of learnable parameters, and a high sensitivity – in terms of network performance – to Gaussian kernel hyperparameters which have to be grid-searched. Finally, its iterative nature, apart from translating to much computational load, may also lead to the phenomenon of vanishing gradients, thus inhibiting the network's rate of convergence [173]. The next chapter takes notice of these problems and proposes a new panoptic segmentation architecture.

Chapter 5: Unifying Training and Inference for Panoptic Segmentation

In addition to the aforementioned concerns concerning the Instance CRF, this work has been partially motivated by the observation that most state-of-the-art methods for panoptic segmentation have adopted a multi-task architecture which only predicts instance and semantic segmentation, and then relies on an offline heuristic merger to combine the two outputs to the desired panoptic format. As a result, they are not trained end-to-end, and cannot directly and globally optimise the panoptic predictions.

In this chapter, we solve the task of panoptic segmentation with a novel network design that is efficient, expressive, and trained end-to-end. Moving away from a CRF-driven design used in previous chapters, the central piece of this network is a densely connected instance affinity operation, which represents the probability that any pair of pixels belong in the same "thing" instance or "stuff" class. In the

past, affinity-based instance segmentation has required complex post-processing steps to extract instances from predicted affinity values [108]. We eliminate this requirement by designing a lightweight mechanism which propagates panoptic logits according to the predicted affinity strengths. This design allows us to predict panoptic segmentation differentiably, and directly optimise the desired output format, end-to-end. At the time of publication, we achieve state-of-the-art results on the Cityscapes and COCO datasets.

We also demonstrate that, thanks to the dense instance affinity operation, our model can achieve competitive performance even without an internal instance mask predictor, which has been widely adopted by and plays an essential role in state-of-the-art methods [70, 87, 83, 157]. For applications with a limited computation budget, one might consider using our mask-free approach.

A shortcoming of this method stems from its large number of loss terms, with seven losses in total. As demonstrated in [70], it is a delicate matter to re-weigh the losses to achieve the best performance. Besides, having loss weights as additional hyperparameters makes the method more difficult to use for new datasets, since an expensive search for best loss weights has to be carried out. A recently proposed method, Panoptic Deeplab [26], seeks to address this issue by designing a bottom-up approach, with only three losses. However, they rely on the aforementioned heuristic merger, and their performance does not match that of the state-of-the-art methods when compared at the same backbone setting.

6.2 Remaining Challenges and Future Directions

Data-efficient training. Despite the vast research efforts on data-efficient learning, fully-supervised neural networks still lead their weakly-supervised counter-

parts by a comfortable margin across scene parsing leaderboards [39, 29, 101]. Compared to other tasks, the amount of labour and budget needed for annotating pixel-level ground truths is very high [29], leading to smaller and potentially fewer datasets available for scene parsing tasks. Some have resorted to using synthetic datasets, which can be churned out by a computer at a much cheaper cost and larger quantity. However, experiments show that neural networks tend to overfit onto the training domain (in this case, the synthetic domain), and perform poorly when there is a shift between the training and deployment domains [164]. This prompts extensive research on domain adaptation techniques, but so far has not managed to match the level of performance that could be achieved with training on manually labelled real-world data [146, 89, 112, 117]. While there have been promising results on the front of self-supervised pretext learning, the model still needs to finetune on a fully supervised dataset to achieve good performance [49, 16, 58]. Overall, it is still an open research question as to how deep neural networks' performance can be maximised with the most efficient data usage.

Explainable and robust algorithms. Throughout much of the history of modern deep neural networks, they have been a black box, which converts images into predictions in a "mysterious" way. While the remarkable modelling capacity of state-of-the-art neural networks has been a boon in terms of performance, it has also challenged our ability to systematically and analytically understand their inner workings and thought process. As a result, it has been difficult to provide theoretical guarantees for neural network's performance in safety-critical applications, such as autonomous navigation and medical imaging analysis. It has also been noticed that deep neural networks are susceptible to adversarial attacks, where a minimally altered input leads to catastrophic failures in prediction [143], calling into question whether we can trust and act upon such a system. There have been

an increasing number of works that seek to advance our understanding and gain insight into the interpretability of deep neural networks [43, 47, 128, 111], but it is still a relatively young field, with many challenges lying ahead.

Incremental learning. The world is constantly evolving, with new inventions and discoveries being made daily. A computer vision model which is trained to perfection on a static set of data can quickly find itself "outgrown" by the rapidly changing world. Whether it is to learn to recognise a new electronic gadget or adapt to the ever-changing trends of fashion, a computer vision model has to continuously learn from new data and update its capabilities in order to remain relevant over time. This ability to incrementally add to an existing knowledge and skills base comes naturally to humans, but is challenging for deep neural networks, and prone to "catastrophic forgetting", a phenomenon where new pieces of knowledge are learnt at the cost of rapid and significant loss of old ones [54].

Synergy with non-visual signals. While vision is an important source of knowledge, it can be advantageous to exploit other sources of information simultaneously, as evident in the biological evolution of non-visual senses in animals. These other senses, including the feeling of sound, touch, temperature, smell, air pressure, *etc.*, provide vital complementary pieces of information regarding the surroundings and can help validate or invalidate a visual detection (or any sensory firing) by cross-examination, leading to more accurate perception and evaluation of the external conditions. For some applications, an ability to process multiple sources of sensory inputs and take actions accordingly is crucial, *e.g.*, nursing robots which need to physically interact with patients and household robots that verbally communicate with their owners. Some progress has been achieved by training machines to answer questions based on visual inputs, especially with questions that have binary answers [113, 144, 140], but much of the field remains uncharted waters.

6.3 Concluding remarks

With the advent of deep learning, there have been major strides in computers' ability to parse and "understand" a scene. Over the course of this thesis, many challenging scene parsing benchmarks [39, 29] have seen huge performance leaps brought about by deep neural networks. However, many challenges have remained untackled, and important questions unanswered. It will take steady effort and rigorous research to tread this long journey, from the algorithms we have now, to a machine vision system which is truly safe, and reliable, intelligent.

Bibliography

[1] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Computer Vision and Pattern Recognition*, 2018. 16

[2] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, 2010. 19

[3] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human Vision and Electronic Imaging VI*. International Society for Optics and Photonics, 2001. 55

[4] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition*, 2014. 60, 73

[5] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision*, 2016. 27, 68

[6] A. Arnab and P. H. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Computer Vision and Pattern Recognition*, 2017. 9, 17, 21, 22, 38, 55, 57, 63, 64, 65, 67, 68, 69, 72, 94, 98, 110

[7] A. Arnab and P. H. S. Torr. Bottom-up instance segmentation with deep

higher order crfs. In *British Machine Vision Conference*, 2016. 28, 32, 36, 37, 38, 55, 68, 69

[8] A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H. Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 2018. 65

[9] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *Computer Vision and Pattern Recognition*, 2017. 16, 55, 57, 67, 72

[10] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. In *arXiv preprint arXiv:1702.06506*, 2017. 59

[11] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*, 2016. 55, 57, 59

[12] M. Berman, A. Rannen Triki, and M. B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Computer Vision and Pattern Recognition*, 2018. 14, 115

[13] G. Bertasius, L. Torresani, S. X. Yu, and J. Shi. Convolutional random walk networks for semantic image segmentation. In *Computer Vision and Pattern Recognition*, 2017. 95

[14] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, 2006. 27

[15] S. R. Bulo, G. Neuhold, and P. Kontschieder. Loss max-pooling for semantic image segmentation. In *Computer Vision and Pattern Recognition*, 2017. 14

[16] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Unsupervised pre-training of image features on non-curated data. In *International Conference on Computer Vision*, 2019. 129

[17] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *Computer Vision and Pattern Recognition*, 2017. 16

[18] A. Chaudhry, P. K. Dokania, and P. H. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *British Machine Vision Conference*, 2017. 56

[19] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, et al. Hybrid task cascade for instance segmentation. In *Computer Vision and Pattern Recognition*, 2019. 15

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015. 27, 32, 38, 63

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 13, 27, 29, 30, 38, 40, 41, 42, 52, 66, 71

[22] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv:1706.05587*, 2017. 66

[23] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Computer Vision and Pattern Recognition*, 2016. 27, 29, 30, 38

[24] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Computer Vision and Pattern Recognition*, 2014. 26, 49, 50, 51

[25] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *Computer Vision and Pattern Recognition*, 2015. 38, 69

[26] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Computer Vision and Pattern Recognition*, 2020. 109, 110, 128

[27] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 56, 62

[28] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Computer Vision and Pattern Recognition*, 2017. 107

[29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition*, 2016. 15, 16, 54, 65, 85, 129, 131

[30] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *International Conference on Computer Vision*, 2015. 55, 57, 58, 68

[31] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Computer Vision and Pattern Recognition*, 2016. 15, 25, 26, 27, 29, 32, 36, 37, 40, 44, 49, 50, 51, 55, 56, 57, 124

[32] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Conference on Neural Information Processing Systems*, 2016. 35, 45, 47

[33] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005. 34

[34] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. In *Computer Vision and Pattern Recognition Workshop*, 2017. 16, 55, 57

[35] D. de Geus, P. Meletis, and G. Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 18, 109, 110

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009. 11

[37] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In *European Conference on Computer Vision*, 2014. 24, 35

[38] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015. 70

[39] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010. 11, 33, 35, 37, 55, 65, 67, 70, 76, 84, 129, 131

[40] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 34

[41] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005. 27

[42] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 1973. 27

[43] R. Fong, M. Patrick, and A. Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *International Conference on Computer Vision*, pages 2950–2958, 2019. 130

[44] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. *Finding pictures of objects in large collections of images*. Springer, 1996. 55

[45] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *Computer Vision and Pattern Recognition*, 2019. 13, 95

[46] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *International Conference on Computer Vision*, 2009. 19

[47] Y. Gal. Uncertainty in deep learning. *University of Cambridge*, 2016. 130

[48] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *International Conference on Computer Vision*, 2019. 109, 110

[49] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 129

[50] R. Girshick. Fast r-cnn. In *International Conference on Computer Vision*, 2015. 14

[51] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin. Instance-level human parsing via part grouping network. In *European Conference on Computer Vision*, 2018. 125

[52] K. Gong, X. Liang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Computer Vision and Pattern Recognition*, 2017. 25

[53] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 2008. 19

[54] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 2020. 130

[55] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision*, 2011. 36, 42, 44, 50, 66

[56] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection

and segmentation. In *European Conference on Computer Vision*, 2014. 26, 27, 29, 35, 38, 57, 67, 69

[57] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Computer Vision and Pattern Recognition*, 2015. 26, 27, 29, 32, 37

[58] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition*, 2020. 129

[59] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *International Conference on Computer Vision*, 2017. 11, 15, 18, 55, 56, 57, 58, 93, 113

[60] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016. 11, 29, 30, 54, 67, 83, 107, 109

[61] X. He and R. S. Zemel. Learning hybrid models for image annotation with partially labeled data. In *Conference on Neural Information Processing Systems*, 2009. 57

[62] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Computer Vision and Pattern Recognition*, 2004. 19

[63] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Computer Vision and Pattern Recognition*, 2018. 96

[64] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *Computer Vision and Pattern Recognition*, 2018. 58

[65] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Computer Vision and Pattern Recognition*, 2017. 66

[66] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *International Conference on Computer Vision*, 2019. 13, 95

[67] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Computer Vision and Pattern Recognition*, 2018. 5

[68] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2017. 57, 58, 59, 60, 63, 68, 69

[69] M. Kiefel and P. V. Gehler. Human pose estimation with fields of parts. In *European Conference on Computer Vision*, 2014. 27

[70] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. In *Computer Vision and Pattern Recognition*, 2019. vi, 18, 90, 91, 92, 93, 96, 97, 104, 105, 106, 107, 108, 109, 113, 115, 116, 117, 118, 119, 128

[71] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Computer Vision and Pattern Recognition*, 2019. 17, 18, 55, 57, 67, 90, 93, 104, 105, 108, 109, 117, 121

[72] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *Computer Vision and Pattern Recognition*, 2017. 16, 55, 57, 72

[73] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *Computer Vision and Pattern Recognition*, 2020. 15

[74] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 2009. 19

[75] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, 2016. 55, 57, 59, 62, 63

[76] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Conference on Neural Information Processing Systems*, 2011. 19, 21, 32, 43, 63, 64, 91, 94

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems*, 2012. 11

[78] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *International Conference on Computer Vision*, 2005. 19

[79] L. Ladickỳ, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *International Conference on Computer Vision*, 2009. 19

[80] L. Ladicky, P. H. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *Computer Vision and Pattern Recognition*, 2013. 27

[81] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989. 11

[82] H. Li, C. Tao, X. Zhu, X. Wang, G. Huang, and J. Dai. Auto seg-loss: Searching metric surrogates for semantic segmentation. *arXiv preprint arXiv:2010.07930*, 2020. 14

[83] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 18, 90, 91, 92, 93, 109, 110, 117, 128

[84] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *Computer Vision and Pattern Recognition*, 2016. 26, 27, 29, 32

[85] Q. Li, A. Arnab, and P. H. Torr. Holistic, instance-level human parsing. In *British Machine Vision Conference*, 2017. 60

[86] Q. Li, A. Arnab, and P. H. Torr. Weakly-and semi-supervised panoptic segmentation. In *European Conference on Computer Vision*, 2018. 94, 100, 107, 109, 110

[87] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang. Attention-guided unified network for panoptic segmentation. In *Computer Vision and Pattern Recognition*, 2019. 18, 90, 91, 92, 93, 109, 110, 117, 128

[88] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *Computer Vision and Pattern Recognition*, 2017. 55, 56, 57

[89] Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Computer Vision and Pattern Recognition*, 2019. 129

[90] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun. Poly-

transform: Deep polygon transformer for instance segmentation. In *Computer Vision and Pattern Recognition*, 2020. 16

[91] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *Computer Vision and Pattern Recognition*, 2018. 125

[92] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan. Proposal-free network for instance-level object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 38, 68, 69

[93] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan. Deep human parsing with active template regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 26

[94] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, 2016. 25, 26, 27, 38

[95] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *Computer Vision and Pattern Recognition*, 2016. 27, 38

[96] X. Liang, Y. Wei, X. Shen, Z. Jie, J. Feng, L. Lin, and S. Yan. Reversible recursive instance-level object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 25, 38

[97] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition*, 2016. 55, 57, 58, 126

[98] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement

networks with identity mappings for high-resolution semantic segmentation. In *Computer Vision and Pattern Recognition*, 2017. 38

[99] G. Lin, C. Shen, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Computer Vision and Pattern Recognition*, 2016. 27

[100] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Computer Vision and Pattern Recognition*, 2017. 15, 96

[101] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 15, 55, 66, 70, 76, 84, 129

[102] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang. An end-to-end network for panoptic segmentation. In *Computer Vision and Pattern Recognition*, 2019. 94, 103

[103] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *International Conference on Computer Vision*, 2017. 16, 55, 69, 72

[104] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Computer Vision and Pattern Recognition*, 2018. 15, 55, 57

[105] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *Computer Vision and Pattern Recognition*, 2016. 25, 26

[106] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia. Multi-scale patch aggregation (mpa)

for simultaneous detection and segmentation. In *Computer Vision and Pattern Recognition*, 2016. 55, 57, 69

[107] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 13

[108] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu. Affinity derivation and graph merge for instance segmentation. In *European Conference on Computer Vision*, 2018. 16, 128

[109] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Computer Vision and Pattern Recognition*, 2016. 26

[110] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition*, 2015. 11, 12, 27, 29

[111] J. Lu and M. P. Kumar. Neural network branching for neural network verification. In *International Conference on Learning Representations*, 2019. 130

[112] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Computer Vision and Pattern Recognition*, 2019. 129

[113] D. Massiceti, P. K. Dokania, N. Siddharth, and P. H. Torr. Visual dialogue without vision or dialogue. *arXiv preprint arXiv:1812.06417*, 2018. 130

[114] P. Meletis, X. Wen, C. Lu, D. de Geus, and G. Dubbelman. Cityscapes-panoptic-parts and pascal-panoptic-parts datasets for scene understanding. *arXiv preprint arXiv:2004.07944*, 2020. 125

[115] D. Modolo and V. Ferrari. Learning semantic part-based models from google images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 24

[116] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Computer Vision and Pattern Recognition*, 2004. 26

[117] L. Musto and A. Zinelli. Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation. *arXiv preprint arXiv:2009.01166*, 2020. 129

[118] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele. Exploiting saliency for object segmentation from image level labels. In *Computer Vision and Pattern Recognition*, 2017. 56, 62

[119] D. P. Papadopoulos, A. D. Clarke, F. Keller, and V. Ferrari. Training object class detectors from eye tracking data. In *European Conference on Computer Vision*, 2014. 54, 84

[120] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. Extreme clicking for efficient object annotation. In *International Conference on Computer Vision*, 2017. 54, 84

[121] G. Papandreou, L. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. In *International Conference on Computer Vision*, 2015. 55, 57, 58, 68

[122] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *International Conference on Computer Vision*, 2015. 57, 58, 59

[123] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Computer Vision and Pattern Recognition*, 2015. 57

[124] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Computer Vision and Pattern Recognition*, 2017. 60

[125] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder. Seamless scene segmentation. In *Computer Vision and Pattern Recognition*, 2019. 109

[126] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *Computer Vision and Pattern Recognition*, 2017. 27, 72

[127] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Conference on Neural Information Processing Systems*, 2015. 11, 14, 54, 67, 82, 113

[128] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining*, 2016. 130

[129] B. Romera-Paredes and P. H. Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, 2016. 27, 33

[130] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics*, 2004. 60, 73

[131] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 1986. 11

[132] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang,

A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. 54, 62

[133] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965. 57

[134] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, 2017. 62

[135] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li. Efficient attention: Attention with linear complexities. *arXiv preprint arXiv:1812.01243*, 2018. 13, 95, 101

[136] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 56, 62

[137] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 2009. 19

[138] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Computer Vision and Pattern Recognition*, 2016. 34

[139] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 11, 54, 68

[140] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training

of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019. 130

[141] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision*, 2017. 54

[142] K.-K. Sung. Learning and example selection for object and pattern detection. In *MIT A.I. Memo No. 1521*, 1996. 34

[143] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 129

[144] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 2019. 130

[145] B. Triggs and J. J. Verbeek. Scene segmentation with crfs learned from partially labeled images. In *Conference on Neural Information Processing Systems*, 2008. 19

[146] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *Computer Vision and Pattern Recognition*, 2018. 129

[147] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, 2016. 72

[148] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems*, 2017. 13

[149] J. J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. In *Conference on Neural Information Processing Systems*, 2008. 57

[150] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Joint object and part segmentation using deep learned potentials. In *International Conference on Computer Vision*, 2015. 26, 27

[151] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Computer Vision and Pattern Recognition*, 2018. 95

[152] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Computer Vision and Pattern Recognition*, 2017. 55, 56, 57, 62

[153] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 56, 62

[154] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Computer Vision and Pattern Recognition*, 2006. 27

[155] Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision*, 2018. 97, 112, 116

[156] F. Xia, P. Wang, L.-C. Chen, and A. L. Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *European Conference on Computer Vision*, 2016. 25, 26, 27, 38

[157] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. Upsnet: A unified panoptic segmentation network. In *Computer Vision and Pattern Recognition*, 2019. 18, 90, 92, 94, 96, 97, 98, 100, 103, 107, 109, 110, 112, 113, 114, 117, 118, 119, 122, 128

[158] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *Computer Vision and Pattern Recognition*, 2012. 26

[159] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition*, 2013. 56, 62

[160] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen. Deeperlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 18, 90, 91, 107, 109, 117

[161] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 67

[162] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 12

[163] Y. Yuan and J. Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 13, 95

[164] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. Fernan-

dez Dominguez. Wilddash-creating hazard-aware benchmarks. In *European Conference on Computer Vision*, 2018. 129

[165] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, 2016. 62

[166] L. Zhang, D. Xu, A. Arnab, and P. H. Torr. Dynamic graph message passing networks. In *Computer Vision and Pattern Recognition*, pages 3726–3735, 2020. 13

[167] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, 2014. 24

[168] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Computer Vision and Pattern Recognition*, 2016. 27, 29

[169] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *International Conference on Computer Vision*, 2015. 27, 29, 55

[170] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Computer Vision and Pattern Recognition*, 2017. 13, 18, 30, 66, 71, 72, 82, 83

[171] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision*, 2018. 13

[172] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, 2015. 27, 31, 32, 65

[173] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, 2015. 19, 21, 22, 94, 127

[174] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition*, 2016. 61, 62

[175] Q. Zhou, X. Liang, K. Gong, and L. Lin. Adaptive temporal encoding network for video instance-level human parsing. In *International Conference on Multimedia*, 2018. 125