UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Universitat Politècnica de Catalunya
Departament d'Arquitectura de Computadors

This thesis is submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy (PhD)

Image and Video Object Segmentation in Low Supervision Scenarios

---

by Míriam Bellver Bueno

Advisors: Jordi Torres, Xavier Giró-i-Nieto
Barcelona, December 2020

Abstract

Image and video segmentation are central tasks within the computer vision field. Nevertheless, deep learning solutions for segmentation typically rely on pixel-level annotations, which are very costly to collect. Likewise, some segmentation systems require human interaction at inference time, which involves effort for the end-user. In this thesis, we look into diverse supervision scenarios for image and video object segmentation. We discern between supervision when learning the model, i.e., which type of annotations are used during training, and supervision at inference, namely which kind of human input is required when running the system. Our target are models that require low forms of supervision.

In the first part of the thesis we present a novel recurrent architecture for video object segmentation that is end-to-end trainable in a fully-supervised setup, and that does not require any post-processing step, i.e., the output of the model directly solves the addressed task. The second part of the thesis aims at lowering the annotation cost, in terms of labeling time, needed to train image segmentation models. We explore semi-supervised pipelines and show results when a very limited budget is available. The third part of the dissertation attempts to alleviate the supervision required by semi-automatic systems at inference time. Particularly, we focus on semi-supervised video object segmentation, which typically requires generating a binary mask for each instance to be tracked. In contrast, we present a model for language-guided video object segmentation, which identifies the object to segment with a natural language expression. We study current benchmarks, propose a novel categorization of referring expressions for video, and identify the main challenges posed by the video task.

Acknowledgments

This thesis has been completed thanks to the help and support of many, to whom I would like to dedicate some acknowledgement words.

First of all, this dissertation was carried out thanks to the *Severo-Ochoa La Caixa* grant of the Barcelona Supercomputing Center (BSC) that financed me during these years. Thanks to BSC for helping me to develop my research. I would also like to thank the Image Processing Group from UPC for their collaboration and assistance.

I want to thank my advisor Xavi, as he was the one that introduced me into research and that encouraged me into doing a PhD. Your optimism and excitement when doing research are contagious. Thanks for really caring about your students and our future, and also for creating a nice atmosphere in our research group. I would also like to thank my advisor Jordi, who was crucial to find resources to pursue my PhD. You have been very helpful when defining the different milestones to achieve during these years. I really appreciate your very good humour and attitude, and how you cared, as I said for Xavi, for us in all situations. You were both very supportive and motivating, thanks!

This journey would have not been the same without my PhD colleagues and also very good friends. Firstly I want to thank Víctor Campos. We started down this road together and I could not have asked for a better company. I dare to say that you are the one I have learnt the most during my PhD, I am really thankful for your unconditional help. We had a lot of fun inside and outside the office, and we were always there for each other when we needed it. Thanks my friend, I am very happy to have completed this cycle by your side, and I wish the best of the futures for you. Another elemental piece in these years has been my very good friend Amaia. When I collaborated with you, I started to really enjoy research at the same time I was discovering one of the most awesome and inspiring people I know. Thanks for everything I learned by your side, not only technical, but also personally. I can proudly say that throughout these years we have built a beautiful friendship, and this is one of the greatest gifts that the PhD has brought me. I also want to thank my friend Amanda for being so supportive and for all the moments we have lived together in conferences, in the office, and also outside of it. Thanks! Finally, a huge thanks to all those who have been in my office in C6 during these years and have made my days much funnier: Juanlu, Eloy, Pol, Adri, Paola, Joan, Xisco, Alberto, Javi, Laia, Cesare, Fabrizio, Maurici, thanks to all!!

I want to also thank my collaborators and advisors during these years. First of all, I want to especially thank Carles Ventura, who has also become a friend after all these years. From my BSc thesis until the very last publication of the thesis, we have been in constant collaboration. I really appreciate working by your side, thanks for everything we have learned and achieved together. I would also like to thank Ferran Marques, for

his essential feedback. Thanks to Carina Silberer, for introducing me to the linguistics world. I wanted to thank everyone who participated in the X-Theses group meetings and that collaborated with me at some point, as Andreu and Giannis, thanks for the good moments and work together! I would also like to especially thank the chances I had to learn from other groups and institutions. First, my stay in ETH Zürich, with Jordi and Kevis, who taught me very valuable knowledge about image segmentation that has surely helped this thesis. Next, my internship in Amazon Berlin, where I had a wonderful time and I learned a lot with all the team, thanks to all! Finally from my internship in Amazon Barcelona, I want to especially thank Javier Romero, who has taught me many valuable lessons about research in general and about 3D in particular.

Following I want to thank my friends, who have been fundamental throughout these years. To Carolina, Mire, Ivo, Marina, Alejo, Xavi, Carles and Guillem, for always being there since we were very little being one of my greatest supports. Also to my friends from university, being some of them in the same deep learning journey as I am. Especially I wanted to thank Ferran, for being the most loyal friend I know. Thanks to Xavi, because you were the first one to talk to me about machine learning, and because all the moments we have shared talking about our own PhDs, thanks my friend! Particularly I wanted to thank Víctor, for being a great support in my PhD, you have helped me both technically and personally, thanks a lot. Last but not least I wanted to thank my friends Sergi, Aida and Jordi, who have been a great support in the last year.

I wanted to especially thank Guillem, for being the greatest discovery in the last year of my PhD, and for giving me support in the last stages of this cycle, thanks from the bottom of my heart.

This thesis is dedicated to my family, who have always believed in me, and have encouraged me to pursue my goals. This was possible thanks to you. I want to first thank my grandparents María de la Paz and Aurelio, who have always been role models to me. I would really love to celebrate this moment by your side. I especially dedicate this thesis to my parents, Montserrat and Josep Antoni, and to my sister, Carla, and her partner, Carlos, for all their support and for always believing in me. Moltes gràcies, aquesta tesis també és vostra, per tot el suport incondicional que m'heu donat durant tots aquests anys, tota l'estima que he rebut, i pels valors que m'heu ensenyat des de que sóc petita. Sou els meus majors referents i sé que no podria tenir-ne de millors. Moltes gràcies de tot cor.

Contents

Acronyms

VOS: Video Object Segmentation

RSIS: Recurrent Semantic Instance Segmentation

RVOS: Recurrent Video Object Segmentation

BASIS: Budget-aware Semi-Supervised Instance and Semantic Segmentation

RE: Referring Expression

LVOS: Language-guided Video Object Segmentation

Glossary

Annotation Budget/Cost: The time required for data labeling.

Training Time: Referred to the training of a deep learning model.

Inference Time: Referred to the stage in which a model has already been trained, and is used to infer predictions for new data. Also known as test time.

Supervision Setup: We distinguish between supervision setup at *training* or *inference* time. It refers to the type of data provided to the algorithm to be trained or to be used at inference, respectively.

Strongly-labeled Data: Data is annotated with complete labels, i.e., the optimal labels for the task addressed.

Weakly-labeled Data: Data is annotated with some partial or inexact label.

Unsupervised Setup: Only data is available without any kind of annotation.

Fully-supervised Setup: Data and the corresponding strong labels are available.

Weakly-Supervised Setup: Given some data, only weak labels available, or only a subset of the data contains some kind of annotation.

Semi-Supervised Setup: It is a type of weakly-supervised setup, that typically consists of having some strongly-labeled data and another set of data which is unlabeled or weakly-labeled.

One-shot Video Object Segmentation: Also known as semi-supervised at inference VOS. It is the task that, given a video sequence and pixel-wise masks for the objects to be tracked in the first frame, the algorithm has to produce masks of the target objects for the rest of the frames.

Zero-shot Video Object Segmentation: Also known as unsupervised at inference VOS. It is the task that, given only a video sequence with no other initialization cue, the algorithm has to discover objects along the video and produce masks for them.

1 Introduction

Computer vision plays a key role in Artificial Intelligence because of the rich semantic information contained in pixels and the ubiquity of cameras nowadays. Multimedia content is on a rise since social networks have such a strong impact in our society and access to the internet becomes more widespread. This context allows the gathering of large datasets which have fostered great advancements in the computer vision field thanks to deep neural networks. These models can effectively exploit large amounts of data to reach a high expressive power. Since the breakout of Imagenet [152], a large dataset for image classification, most computer vision tasks have benefited from deep neural networks. Among the different tasks in the computer vision field, locating objects in images and videos is a central one, as it has many applications in autonomous driving, surveillance, image and video edition, medical diagnosis and biometrics along with others. Localization of objects can be obtained with bounding boxes around the target objects, or with accurate pixel-level masks that delineate the instances. The latter is a more challenging task, but fundamental for certain applications where edges of objects need to be determined. The main task addressed in this thesis is *instance segmentation*, that consists in, given an image or video, providing pixel-level masks for each instance of certain semantic object classes.

In order to train a segmentation model, current solutions rely on large amounts of pixel-wise annotations, which demand significant human effort to collect. Furthermore, expert knowledge is needed to gather certain annotations, such as labels for medical images. In consequence, there is a huge interest for systems that work with less-demanding forms of supervision, such as weakly or semi-supervised pipelines.

Besides, in some segmentation tasks, human effort is not only needed for training the models, but also at inference. In semi-automatic systems, user input may be required as guidance to start the system. One example is the task of *one-shot Video Object Segmentation* (osVOS) [136], which expects that the end-user provides a pixel-level mask for each object to be tracked in the first frame of the video. Following, the model must predict the segmentation mask of the tracked objects for the remaining frames. These initialization cues are crucial for high accuracy, but they are arduous to obtain. An alternative are models that depend on weaker input signals that are user-friendlier.

This thesis explores different supervision scenarios for the instance segmentation task, distinguishing between supervision during *training* and at *inference*, and focusing on low-supervision setups. We start with fully-supervised models that rely on large amounts of annotated data, to later reduce the annotation burden during training by using semi-supervised setups. Lastly, we focus on the inference mode of the system, and we leverage language as a weak guidance for the osVOS task.

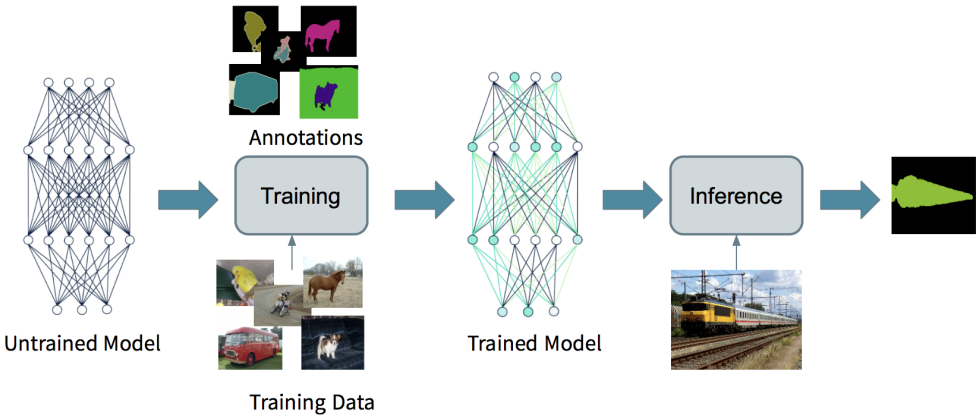

Figure 1.1: Two different stages when deploying a deep learning model: training and inference. In this example, the model would be trained to address semantic segmentation.

1.1 Objectives

In this Section we define which are the objectives of this dissertation. The main goal of the thesis is **to lower the human effort required for running segmentation models**. We work with different supervision scenarios on both *training* and *inference*. To make this idea clearer, we show in Figure 1.1 a typical deployment of a deep learning system. It has two different stages: firstly, the model is trained with some collected data and the corresponding annotations. Secondly, once the model is trained and it has capabilities to address the target task, it can be used in inference mode in order to infer predictions on unseen data.

This thesis consists of three different Parts. Following, we enumerate our main objectives for each of them:

Part I: Supervised Learning for Image and Video Segmentation

- We aim at solving video object segmentation (VOS) in and end-to-end manner.
- We aim at addressing VOS in an unsupervised setup at inference (also named zero-shot), i.e., that the model learns to discover objects within the sequence without any initialization cue of what objects to follow.

Part II: Semi-Supervised Learning for Image Segmentation

- The goal is to lower the annotation required at training time for segmentation systems, by exploiting semi-supervised models.

Part III: Language-guided Video Object Segmentation

- Our target is to lower the human effort required at inference time for video object segmentation systems compared to one-shot systems, by exploiting language as a

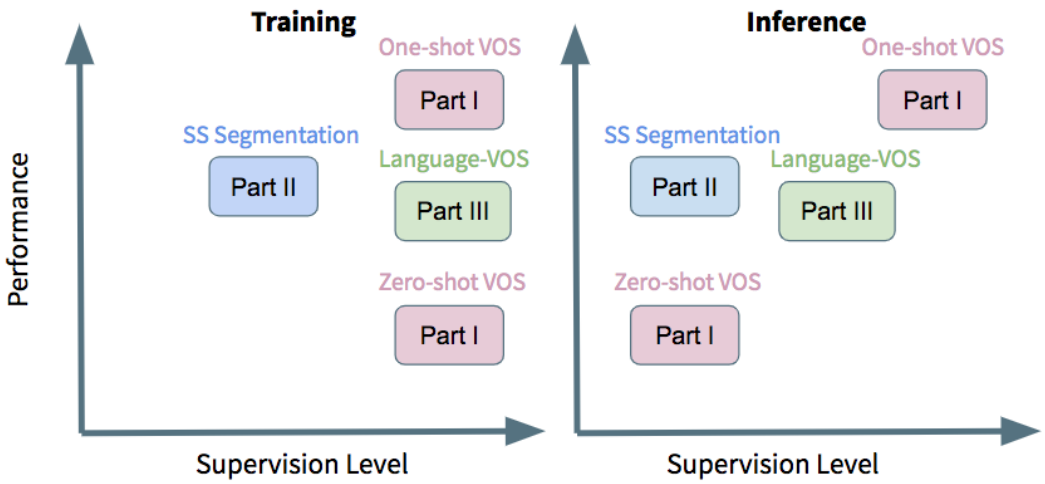

Figure 1.2: The expected performance for each different supervision setup that we target at each Part of the thesis. *SS* stands for semi-supervised, and *VOS* for Video Object Segmentation. Language-VOS refers to Language-guided VOS. We distinguish between training and inference mode.

weak supervision to indicate which objects to segment in the video. We refer to this task with the term *language-guided VOS*.

- We aim at analyzing current benchmarks for language-guided VOS, in order to identify the main challenges of the task.

The different supervision scenarios that we explore are summarized in Figure 1.2, where the expected performance depending on the supervision level is illustrated, for both training and inference. The Figure illustrates in which supervision scenarios we want to contribute with this dissertation.

1.2 Research Questions

Most best-performing models for instance segmentation are composed of two stages [143]. Given an image or video, a first stage proposes object candidate regions, and a second stage semantically classifies these candidates between a predefined set of semantic categories. Two-stage models typically require a post-processing step in order to filter overlapping predictions, so that a single bounding box is assigned to each object instance. The current trend in deep learning moves towards systems that are fully end-to-end trainable [21], meaning that the loss optimized corresponds to the actual task addressed, and that no additional post-processing is required. The first research question addressed in **Part I**, is: **Is it possible to train fully end-to-end architectures for video object segmentation with Recurrent Neural Networks?** In this Part of the thesis we propose a novel architecture for video object segmentation that is composed of a single stage and that does not require any post-processing step. The architecture is based on Recurrent Neural Networks in order to discover and track objects in a video sequence, and it is recurrent in both the spatial (within the image) and the temporal (across video frames) domains. Moreover, the most common setup for video object segmentation is

the one-shot case, where a pixel-level mask for each object to be tracked is expected at inference time. In this Part, we also formulate the following question: **Can we train a video model that discovers objects along the video sequence without any initialization cue?** Ours was the first end-to-end trainable solution that tackled this task for videos.

Whereas the first Part of the thesis focuses on fully-supervised systems, in Part II we explore semi-supervision to reduce the annotation time required to train segmentation models. Instance segmentation solutions based on deep learning are traditionally trained with strong labels, that is, all objects in the training dataset are manually labeled at a pixel-level. This annotation procedure is time-expensive and may require expert knowledge, for example when annotating medical images. Some alternatives to alleviate the annotation burden exploit weaker forms of supervision, by using weak annotations such as image-level labels or bounding boxes, or by relying on fewer strong annotations (semi-supervision). The latter leverages a limited amount of strongly-annotated data with a large amount of unlabeled or weakly-labeled samples, and it aims at better exploiting the scarce expert knowledge that we may have available. The research question addressed in **Part II** of the dissertation is the following: **Can we train semi-supervised systems for segmentation on very low annotation budgets?** Part II of this thesis focuses on training segmentation models with semi-supervised pipelines, showing the first results with very low annotation budgets for instance segmentation, and proving that semi-supervised setups for image segmentation reach better results compared to relying on weak annotations (image-level labels or bounding boxes), with matching annotation costs.

Part I and II cover fully- and semi-supervised setups for image segmentation, focusing on the supervision when *training* the models. In Part III we explore weakly-supervised segmentation systems at *inference*. Specifically, we leverage natural language for the video object segmentation task, first addressed in Part I. In the commonly-named *one-shot video object segmentation* task [18], also referred as semi-supervised object segmentation, the objects of the first frame of a video must be pixel-wise labeled at inference time, so that the model can track the instances throughout the video sequence. In **Part III**, the research question addressed is the following: **Can we use language to reduce the human effort required at inference time in semi-supervised VOS systems?** Aiming at a user-friendlier human-computer interaction, in Part III we explore natural language as a weak form of supervision at test time. Particularly, we exploit referring expressions to indicate which object is to be segmented. This setup reduces the complexity of generating a mask for the target object, to producing a simple linguistic expression that uniquely refers to the instance. We build a neural network that processes natural language as input and that discovers the referred objects in the video sequence. Our model is competitive for images and outperforms all previous works for video. Furthermore, another research question that we address in this Part is the following: **Are current benchmarks suitable for the video task?** We argue that existing datasets [85, 55] are unsuitable for this task as they mostly contain trivial examples, so we carefully analyze, filter and augment referring expressions of current benchmarks to identify the main challenges of language-guided video object segmentation.

Chapter	Image	Video	Training	Inference
3		✓	Supervised	Semi- & Unsupervised
4	✓		Semi-Supervised	Unsupervised
5	✓		Active Learning	Unsupervised
6	✓	✓	Supervised	Weakly-Supervised

Table 1.1: Classification of the thesis Chapters based on the supervision level.

1.3 Contributions and Thesis Outline

The main contributions of each Part of the thesis are highlighted in this Section.

Part I: Supervised Learning for Image and Video Segmentation

- **Chapter 3**: We introduce RVOS, the first end-to-end architecture for video object segmentation. We evaluate our model on DAVIS 2017 [136] and Youtube-VOS [182] benchmarks. We are the first work to address zero-shot video object segmentation, that is, to discover objects in video sequences without any initialization cue.

Part II: Semi-Supervised Learning for Image Segmentation

- **Chapter 4**: We propose BASIS, a semi-supervised pipeline for image segmentation based on self-learning. BASIS is state-of-the-art for semantic and instance segmentation with very low annotation budgets on the Pascal VOC dataset [47]. We identify how semi-supervised pipelines can surpass the performance achieved by weakly-supervised setups, at matching annotation cost.
- **Chapter 5**: We study a novel active learning mechanism to better choose which images to strongly-annotate for our semi-supervised pipeline BASIS.

Part III: Language-guided Video Object Segmentation

- **Chapter 6**: We present RefVOS, a model for language-guided video object segmentation that is state-of-the-art on DAVIS 2017 [136] and A2D datasets [179].

  We introduce a novel semantic categorization of referring expressions tailored for the video task. We augment the referring expressions of A2D to analyze the impact of the different categories, and identify the main challenges for the video task.

Table 1.1 summarizes the content presented in this thesis, by classifying the different Chapters based on the modality tackled (image or video), and the type of supervision applied either while training or at inference. Prior to the three main Parts of the thesis, a brief technical background review is presented in Chapter 2.

1.4 List of Peer-Reviewed Publications

This Section contains a list of the peer-reviewed publications for each Part of the thesis, together with publications that are not covered in this dissertation.

Part I: Supervised Learning for Image and Video Segmentation

- **Chapter 3**: Amaia Salvador, *Miriam Bellver*, Victor Campos, Manel Baradad, Ferran Marqués, Jordi Torres, Xavier Giro-i-Nieto. Recurrent Neural Networks for Semantic Instance Segmentation. In DeepVision Workshop in CVPR 2018 [154]. This project is open-source (https://github.com/imatge-upc/rsis).

- **Chapter 3**: Carles Ventura* [1], *Miriam Bellver**, Andreu Girbau, Amaia Salvador, Ferran Marques and Xavier Giro-i-Nieto. RVOS: End-to-End Recurrent Net for Video Object Segmentation. In CVPR 2019 Proceedings [169]. This project is open-source (https://github.com/imatge-upc/rvos).

Part II: Semi-Supervised Learning for Image Segmentation

- **Chapter 4**: *Miriam Bellver*, Amaia Salvador, Jordi Torres and Xavier Giro-i-Nieto. Budget-aware Semi-Supervised Semantic and Instance Segmentation. In DeepVision Workshop Proceedings in CVPR 2019 [14] (Best paper award).

- **Chapter 5**: *Miriam Bellver*, Amaia Salvador, Jordi Torres and Xavier Giro-i-Nieto. Mask-guided sample selection for Semi-Supervised Instance Segmentation. Multimedia Tools and Applications 2020 [12]. DOI: 10.1007/s11042-020-09235-4.

Part III: Language-guided Video Object Segmentation

- **Chapter 6**: *Miriam Bellver*, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres and Xavier Giro-i-Nieto. RefVOS: A Closer Look at Referring Expressions for Video Object Segmentation. Submitted [13]. This project is open-source (https://github.com/miriambellver/refvos).

Publications not covered in the thesis

- **Hierarchical object detection with deep reinforcement learning**: In this work we presented a method for performing hierarchical object detection in images guided by a deep reinforcement learning agent. The key idea is to focus on those parts of the image that contain richer information and zoom on them. We train an intelligent agent that, given an image window, is capable of deciding where to focus the attention among five different predefined region candidates (smaller windows). This procedure is iterated providing a hierarchical image analysis. We compare two different candidate proposal strategies to guide the object search: with and without overlap. Moreover, our work compares two different strategies to extract features from a convolutional neural network for each region proposal: a first one that computes new feature maps for each region proposal, and a second one that computes the feature maps for the whole image to later generate crops for each region proposal. Experiments indicate better results for the overlapping candidate proposal strategy and a loss of performance for the cropped image features due to the loss of spatial resolution. We argue that, while this loss seems unavoidable when working with large amounts of object candidates, the much more reduced amount of region proposals generated by our reinforcement learning agent allows considering

[1] (*) Joint work with Dr. Carles Ventura from Universitat Oberta de Catalunya

to extract features for each location without sharing convolutional computation among regions.

This work resulted into two publications:

- *Miriam Bellver*, Xavier Giró-i-Nieto, Ferran Marqués and Jordi Torres. Hierarchical object detection with deep reinforcement learning. In Deep Reinforcement Learning Workshop in Neurips 2016.
- *Miriam Bellver*, Xavier Giró-i-Nieto, Ferran Marqués and Jordi Torres. Hierarchical object detection with deep reinforcement learning. In Deep Learning for Image Processing Applications, 2017 [17]. DOI: 10.3233/978-1-61499-822-8-164.

This project is open-source (https://github.com/imatge-upc/detection-2016-nipsws).

- **Detection-aided liver lesion segmentation using deep learning**: A fully automatic technique for segmenting the liver and localizing its unhealthy tissues is a convenient tool in order to diagnose hepatic diseases and assess the response to the according treatments. In this work we propose a method to segment the liver and its lesions from Computed Tomography (CT) scans using Convolutional Neural Networks (CNNs), that have proven good results in a variety of computer vision tasks, including medical imaging. The network that segments the lesions consists of a cascaded architecture, which first focuses on the region of the liver in order to segment the lesions on it. Moreover, we train a detector to localize the lesions, and mask the results of the segmentation network with the positive detections. The segmentation architecture is based on DRIU [111], a Fully Convolutional Network (FCN) with side outputs that work on feature maps of different resolutions, to finally benefit from the multi-scale information learned by different stages of the network. The main contribution of this work is the use of a detector to localize the lesions, which we show to be beneficial to remove false positives triggered by the segmentation network.

  This work derived into the following publication:

  - *Miriam Bellver*, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Xavier Giró-i-Nieto, Jordi Torres, Luc Van Gool. Detection-aided liver lesion segmentation using deep learning. In Machine Learning 4 Health Workshop in Neurips 2017 [11].

  This project is open-source (https://github.com/imatge-upc/liverseg-2017-nipsws).

  With our method, we ranked 10th on the Liver Tumor Segmentation Benchmark (LITS), and we participated in the journal of the challenge [15].

- **Distributed training strategies for a computer vision deep learning algorithm on a distributed GPU cluster**: Deep learning algorithms base their success on building high learning capacity models with millions of parameters that are tuned in a data-driven fashion. These models are trained by processing millions of examples, so that the development of more accurate algorithms is usually limited by the throughput of the computing devices on which they are trained. In this work, we explore how the training of a state-of-the-art neural network for computer

vision can be parallelized on a distributed GPU cluster. The effect of distributing the training process is addressed from two different points of view. First, the scalability of the task and its performance in the distributed setting are analyzed. Second, the impact of distributed training methods on the final accuracy of the models is studied.

This work derived into the following publication:

– Victor Campos, Francesc Sastre, Maurici Yagües, *Miriam Bellver*, Xavier Giró-i-Nieto and Jordi Torres. Distributed training strategies for a computer vision deep learning algorithm on a distributed GPU cluster. In Procedia Computer Science, 2017 [20].

1.5 List of Research Stages and Internships

This Section presents a list of the research stages and internships pursued in the course of this dissertation:

- February 2017 - August 2017: Research internship at the Eidgenössische Technische Hochschule Zürich (ETHZ) in the Computer Vision Lab (CVL) team, under the supervision of Dr. Jordi Pont Tuset and Dr. Kevis-Kokitsi Maninis.
- April 2019 - August 2019: Internship at Amazon Berlin, in the Computer Vision Team, under the supervision of Dr. Matthieu Guillaumin.
- April 2020 - July 2020: Internship at Amazon Barcelona, in the Rhapsody Team, under the supervision of Dr. Javier Romero.

2 Technical Background

This Chapter aims at introducing some technical concepts to ease the reading of the manuscript.

Firstly, Section 2.1 introduces the main core computer vision tasks related to this dissertation. Secondly, in Section 2.2 we introduce neural-based architectures which are the foundations of the different models presented in this thesis. Following, as the main thread of discussion of this thesis is the supervision level applied at either *training* or *inference*, in Section 2.3 different supervision scenarios are described. Lastly, key concepts on Natural Language Processing which are relevant for Part III of the thesis are introduced in Section 2.4.

2.1 Computer Vision Tasks

Computer Vision is the field that studies how computers perceive and understand digital images and videos [161]. Some of the most popular tasks are image classification, object detection, image segmentation or image captioning in images or videos. This thesis focuses on methods related to image segmentation. Particularly, in Part I we work with a novel end-to-end architecture for instance segmentation, and in Parts II and III of the thesis, we focus on segmentation models in low supervision scenarios. Thus, in this Section we introduce relevant related work about object detection (as instance segmentation is a natural extension of this task) and image segmentation.

2.1.1 Object Detection

Object detection is the task of specifying a bounding box for each object instance in an image or video. Typically a semantic class category for each instance is also required. Deep learning approaches have addressed the task following two basic strategies: proposal-based object detection and single-shot object detection, explained in Sections 2.1.1.1 and 2.1.1.2 respectively.

2.1.1.1 Proposal-based object detection

Proposal-based methods are composed of two stages. The first one proposes candidate regions that may contain objects in the image. The second one classifies these object candidates among a set of pre-defined class categories. These algorithms rely on a final post-processing step that filters out the object proposals in order to have a single bounding box around each object in the image.

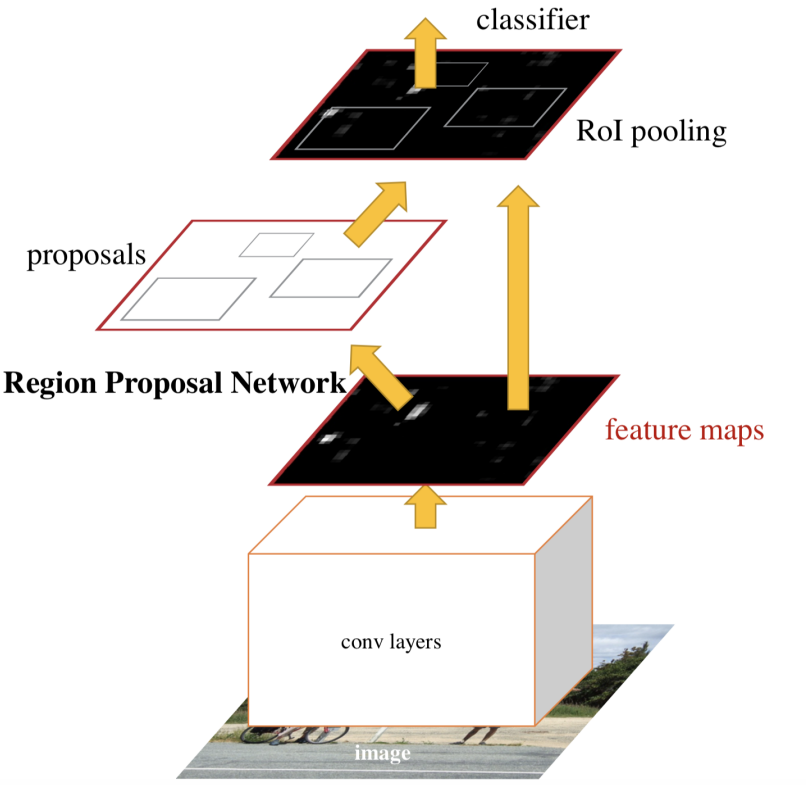

Figure 2.1: Faster R-CNN. Given an image, features are extracted with a CNN and fed to the Region Proposal Network, that provides a number of proposal regions that may contain objects. Next, each region proposal is described using the Roi-Pooling operation. The last step consists in classifying semantically the proposal. Figure from [143].

Region-based Convolutional Neural Networks (R-CNNs) [57] was the first work that applied deep learning for object detection, and it is a proposal-based method. In its first stage, object proposals are obtained. These are produced with algorithms based on hand-crafted features, such as Selective Search [167] of MCG [134]. In the second stage, each of these regions is represented with a feature vector obtained from a CNN, and later classified with a class category. The initial proposals are usually regressed and improved during this second stage. The same authors of the aforementioned work proposed an improvement with Fast R-CNN [56]. Instead of forwarding each object proposal to the CNN, which is computationally very expensive, they proposed to extract features only once for the whole image, and introduced the Region Of Interest Pooling operation (ROI-pooling) to crop from these feature maps the regions of interest belonging to each original proposal. Thus, the computational resources required are decreased significantly. Following, Faster R-CNN [143] (Figure 2.1) removed the step of having an external algorithm to obtain object proposals, and added into the architecture a Region Proposal Network (RPN) that directly leveraged convolutional features to predict regions of interest within the image. These same convolutional features are later used to describe each region. Faster R-CNN is much faster than previous works, and is still a standard reference as object detector. Subsequent modifications have added a branch to obtain a binary mask for each bounding box predicted, solving the task of image instance segmentation [64].

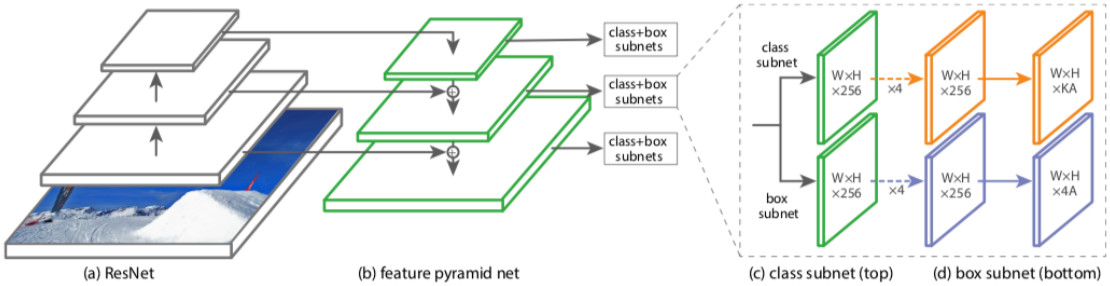

Figure 2.2: RetinaNet architecture. It is a one-stage network composed of a feature pyramid to extract features from the input image, and two subnetworks: one to classify the anchor boxes and another to regress them. Figure from [99].

Many works have built on top of Faster-RCNN for object detection or instance segmentation [41, 39, 97]. All of them share the following steps: (1) Objects proposal generation, (2) Objects proposals classification and regression, and (3) Post-processing step to filter object proposals. Although these methods can be very fast at inference, they still need a final filtering step to have a single bounding box per object in the image. Furthermore, the fact that they have two different stages prevents from having a single loss that reflects the task of object detection. Other works, discussed in Section 2.1.1.2, develop strategies for single-shot architectures, with a single loss to optimize the model.

2.1.1.2 Single-shot object detection

This class of object detectors are called *single-shot* because it takes only one shot to detect multiple objects present in an image. Therefore, they are typically faster than proposal-based methods, although the performance is also typically lower. These models apply a single neural network to the full image, by dividing the input image into regions and predicting bounding boxes and probabilities for each region [138, 105, 99]. The main difference compared to proposal-based methods, is that in this case regions are not pre-selected. From these models, we want to highlight RetinaNet [99] (Figure 2.2). The authors discovered that extreme foreground-background class imbalance encountered during training of single-shot detectors is the main reason for low performance compared to proposal-based methods. Therefore, they propose a novel focal loss that focuses on training on hard examples by removing all those regions that belong to background and are easy negative examples. With this loss, they achieve state-of-the-art performance for object detection at faster rates compared to proposal-based strategies.

2.1.2 Image Segmentation

The broad definition of image segmentation is the process of partitioning an image into multiple segments, i.e., sets of pixels, in order to simplify and/or change the representation from an image into something more meaningful. In the context of deep learning, image segmentation is a dense prediction task, as each input pixel requires an output. In this thesis we focus on semantic and instance segmentation, which are described in more details in Sections 2.1.2.1 and 2.1.2.2 respectively.

Figure 2.3: Semantic Segmentation examples. The second and fourth images are the results obtained by a semantic segmentation method for first and third image respectively.

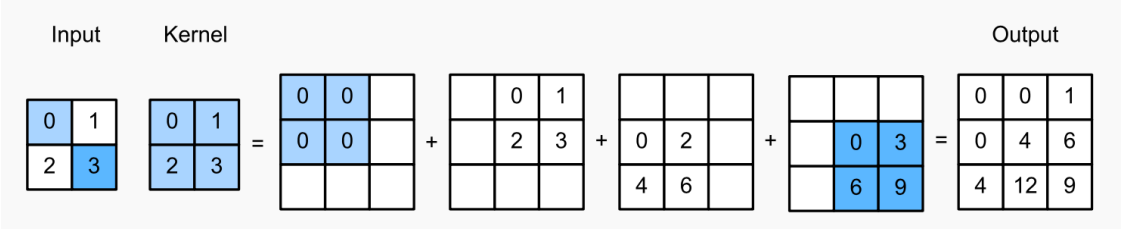

Figure 2.4: Transposed convolution layer with a 2×2 kernel. Each parameter value in the kernel is multiplied to the input tensor. Following, the resulting matrices are added. Figure from [191].

2.1.2.1 Semantic Segmentation

In semantic segmentation, given an image, a classification between a predefined set of semantic categories for each input pixel is required. Some examples of the task are depicted in Figure 2.3. Typical architectures for semantic segmentation are Fully Convolutional Networks (FCNs) [106]. The main characteristic of FCNs is that they are composed of convolutional and pooling layers only, without any fully connected layer. This allows that FCNs can handle varying input sizes. Moreover, they can work faster as matrix multiplications from fully connected layers are computationally very expensive.

In segmentation tasks it is required to preserve the low-level details, as edges and borders are crucial to obtain a precise segmentation. Deep learning architectures are typically tailored to overcome the loss of spatial information induced by pooling operations from CNNs, such as *upsampling methods*, *dilated convolutions* and *skip connections*.

One manner to overcome the loss of spatial information produced by down-sampling operations is to compensate them with upsampling ones, and recover the input resolution at the end of the network. This can be performed through bilinear interpolation or with transposed convolutions. *Transposed convolutions* (Figure 2.6), also known as deconvolutions or strided convolutions, are just convoluting the input back to a larger size in order to increase resolution [106]. Transposed convolutions apply a regular convolution but reverting the spatial information.

Another strategy to keep low-level details is to avoid down-sampling operations, and keep a high resolution throughout all the network. However, pooling layers are not only meant to down-sample the resolution, but also to increase the field of view to learn more abstract concepts. Increasing the field of view can also be achieved by increasing the kernel size, but at the expense of more parameters. An alternative is to exploit *Dilated convolutions*, also known as atrous convolutions [27, 28, 187, 29]. The dilation rate defines the spacing

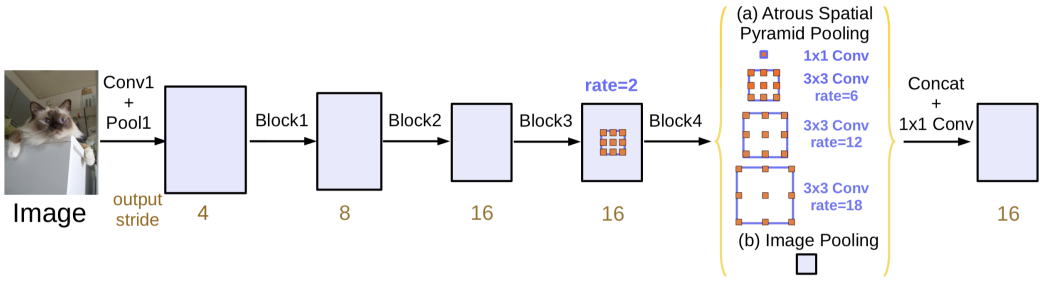

Figure 2.5: DeepLabv3 architecture for semantic segmentation. The model has an Atrous Spatial Pyramid Pooling module with parallel branches working at different dilation rates in order to capture multi-scale context. Figure from [29].

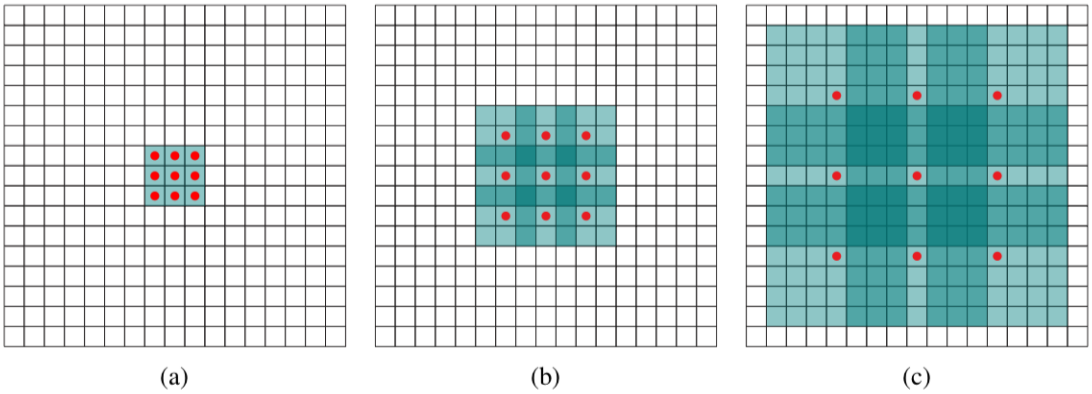

Figure 2.6: Dilated Convolution. a) shows a 3x3 kernel with dilation rate=1, with a field of view of 3×3. Figure b) shows the same kernel but with a dilation rate of 2, so the field of view is increased to 7×7. In Figure c) there is depicted the same kernel but with dilation ratio of 4, being now the field of view of 15×15. Figure from [187].

between the values in a kernel, i.e., a 3×3 kernel with dilation rate of 2 has the same field of view as a kernel of size 5×5, but it only requires 9 parameters (Figure 2.6). A model which employs atrous convolutions for semantic segmentation is DeepLabv3 [29] and is depicted in Figure 2.5. DeepLabv3 has parallel modules that apply atrous convolutions at different rates. In their work they show how this helps to segment objects at different scales.

Another strategy to preserve the information lost in the contracting path of a convolutional neural network, is to leverage skip connections. *Skip connections* are connections from early to latter layers, and they can appear in a CNN in many ways. One option is to connect features from the contracting path of the network to the features of the expanding path (i.e., the path with upsampling operations to recover the spatial resolution) [146]. These connections can be in form of additions, concatenations or multiplications, among other options. An alternative is to produce predictions at several stages of the network (from shallow and also from deep layers) and combine the results [107].

Figure 2.7: Instance Segmentation examples. Each image has an overlay of segmentation masks predicted by an instance segmentation method.

2.1.2.2 Semantic Instance Segmentation

Semantic instance segmentation is defined as the task of assigning a binary mask and a categorical label to each object in an image. This task is very similar to object detection, but instead of bounding boxes, binary masks must be predicted for each object instance. In Figure 2.7 some examples with images and the corresponding obtained masks by an instance segmentation method are shown.

Pipelines for this task are typically an extension of proposal-based object detection methods, by just adding some mechanism to segment the object within the proposal region [62, 63, 32]. Some works build on top of Faster R-CNN, and add cascade of predictors [62, 63, 97] and refinement of the binary masks [97]. A very popular work is [64], which adds a parallel branch to Faster R-CNN that predicts the binary mask for the proposal.

In contrast to the aforementioned architectures, some works consider the image holistically in order to obtain the objects segments. Thus, these approaches do not rely on object proposals. These include works that use Conditional Random Fields [6] or watershed transform on top of a semantic segmentation map to identify the different object instances [8]. An alternative is metric learning in order to cluster object pixels to obtain the instance segments [43].

In Part I of the thesis, we cast instance segmentation as a sequential problem, leveraging Recurrent Neural Networks (RNNs). Particularly, we adapt RSIS [154], an architecture for recurrent semantic instance segmentation, in order to perform video object segmentation. Previous to RSIS, other works had treated instance segmentation as a sequential process. Ren & Zemel [141] exploit attention to focus on different regions of the image in a sequential manner, and produce the instance segmentation results. Additionally, their model has a recurrent module to improve the segmentation mask provided for each region of interest in an iterative way. More similar to RSIS, Romera-Paredes & Torr [144] leverage Recurrent Neural Networks to predict binary masks for the different objects within an image, but their model is class-agnostic and predicts instances of a single object category. Furthermore, they rely on features pre-trained on semantic segmentation.

A detailed overview of the architectures for instance segmentation for videos is presented in Section 3.2 of Part I of this thesis.

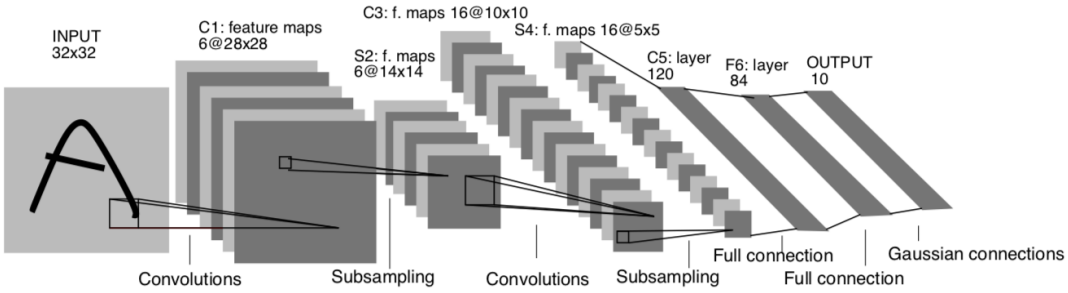

Figure 2.8: LeNet-5 architecture for digits recognition [91]. Each convolutional layer produces a set of feature maps, that are sub-sampled to reduce the resolution. After 4 convolutional layers, the network has a classifier in the form of a Multi-Layer Perceptron.

2.2 Deep Learning Architectures

This Section describes two neural networks that are fundamental for the different models presented within the thesis. Firstly, in Section 2.2.1 Convolutional Neural Networks are described. Secondly, Section 2.2.2 explains Recurrent Neural Networks.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the most popular type of deep neural networks in computer vision [88, 159, 66]. CNNs are networks specialized in processing data that has a known grid-like topology [58], like images or video frames, that are a 2-D grid of pixels. The main property of CNNs is that at least one of the layers of the network is a *convolutional layer*, meaning that it employs convolutions instead of general matrix multiplication applied in a typical fully-connected layer. CNN architectures often combine *convolutional layers* with *pooling layers*, which sub-sample the resulting feature maps from the convolutional layers. An example of the overall architecture of a CNN is illustrated in Figure 2.8.

The motivation to employ convolutions is grounded on three reasons: sparse interactions, parameter sharing and equivariant representations. Neurons in fully-connected layers are connected, and hence interact, to all neurons of the next layer. Convolutional layers, on the other hand, have sparse interactions, as convolutional kernels are typically smaller than the input. Sparse interactions are suitable for data such as images, where the information is local. An example to understand that information in images is local, is that objects, or object parts, only occupy certain region of the image, that can be detected by a convolutional kernel. Convolutional filters are specialized in detecting certain patterns, and will be convolved throughout all the input data. The filters share parameters throughout all locations, reducing the overall number of parameters of the convolutional layer compared to fully connected layers. Parameter sharing causes that convolutional layers are equivariant to translation, i.e., if the input changes, the output changes in the same way, which allows the network to generalize edge, texture and shape detection in different locations.

The result of convolving a filter produces what is called a *feature map*. Each layer produces as many feature maps as filters convolved to the given input. Subsequently,

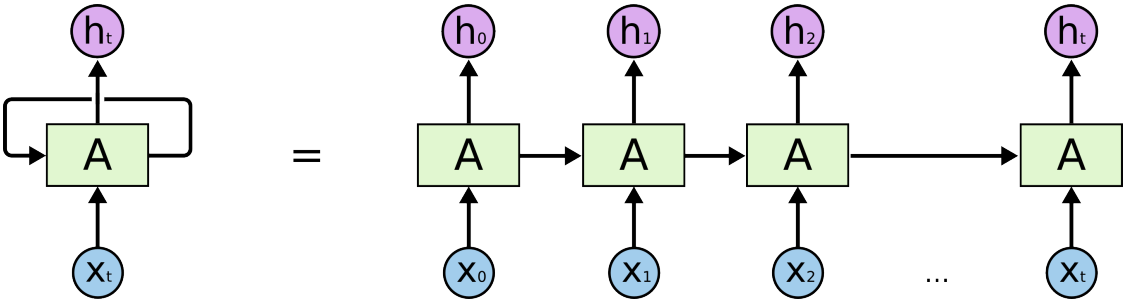

Figure 2.9: Unrolled RNN. It can be thought as multiples copies of the same network that pass information from one step to the next one. Figure from [120].

these feature maps are the input of the following layer. The role of a *pooling layer* is to sub-sample feature maps, usually performing a max pooling operation over grid regions from the input. Pooling is desirable for several reasons. First, from a computational point of view, if the feature maps are sub-sampled, less computations are required. Another reason is to gain spatial invariance to small changes of the input, which helps to learn more generalized representations. Finally, pooling also enables to increase the receptive field as we go deeper into the network by making the feature maps smaller. Consequently, filters see features corresponding to wider regions from the input data, which allows to learn more abstract representations. This helps into learning a hierarchy of features, starting from low-level features in early layers, and moving to more abstract and global representations in the deepest levels of the network.

Another key for the success of CNNs, is that they can be trained very efficiently thanks to modern accelerators such as graphics processing units (GPUs) and optimized deep learning frameworks [1, 125].

2.2.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a class of neural network designed to recognize patterns in sequences of data. These networks take as their input the current input and also information from the previous time step of the sequence. RNNs have a feedback loop that connects each time step to past decisions, so it is commonly said that RNNs have memory as they allow information to persist. The sequential information is preserved in the network's hidden state. Figure 2.9 depicts an unrolled RNN model.

The mathematical formulation to define the hidden state of a vanilla RNN is shown in Equation 2.1, where $\mathbf{h}_t$ is the hidden state of a RNN at step t. The hidden state is a function of the input at this same time step $\mathbf{x}_t$ modified by a weight matrix W, and then added to the hidden state from the previous time step $\mathbf{h}_{t-1}$ modified by another weight matrix U. The weights matrices W and U determine how much information to keep from the current input and from the previous hidden states.

$$\mathbf{h}_t = \phi(W\mathbf{x}_t + U\mathbf{h}_{t-1}) \tag{2.1}$$

A popular type of RNNs are Long Short-Term Memory (LSTMs) [69] networks. An LSTM unit is composed of a *cell*, an *input gate*, an *output gate* and a *forget gate*. The

cell is capable of remembering values over time intervals and the different gates regulate the flow of information into and out of the cell. In contrast to vanilla RNNs, LSTMs are popular for remembering longer-term dependencies. The gates of a LSTM are implemented with fully connected layers. If, instead, they are convolutional layers, then it is a Convolutional LSTM (ConvLSTM) [178] unit.

RNNs are typically used in the fields of natural language processing and speech recognition, but they can also be leveraged in computer vision as we will explore in Part I of this thesis, where we employ ConvLSTMs in our proposed architectures for instance segmentation and video object segmentation.

2.3 Supervision Paradigms

This Section aims at introducing different supervision setups at both *training* and *inference* of a deep learning pipeline. First, in Section 2.3.1 the three main learning paradigms are introduced. Next, Section 2.3.2 describes weak supervision. Lastly, Section 2.3.3 focuses on the supervision scenarios at inference time.

2.3.1 Training Supervision Scenarios

We distinguish different machine learning paradigms depending on the level of supervision during training: supervised, unsupervised and semi-supervised learning.

In *supervised learning*, ground truth labels are available for the training samples, i.e., a prior knowledge of what the output values for the given data samples should be in our problem. Thus, the goal is to learn a function that maps your input data to the desired output or target. Taking the notation from [24], formally the setup consists of a training set made of pairs (x_i, y_i), where $y_i \in \mathcal{Y}$ are the labels or targets of the samples $x_i \in \mathcal{X}$. The pairs (x_i, y_i) are samples i.i.d. (independently and identically distributed) from some distribution. The goal is to estimate a function $\mathcal{Y} = f(\mathcal{X})$.

In *unsupervised learning*, only input data is available without corresponding output targets. Thus, the goal is to model the underlying structure or distribution present within a set of data points. Formally, let $X = (x_1, ..., x_n)$ be a set of n examples, where $x_i \in \mathcal{X}$ for all $i \in [n] := \{1, ..., n\}$. It is typically assumed that points are drawn i.i.d. from a common distribution on $\mathcal{X}$. Unsupervised learning consists in estimating a density which is likely to have generated X. A type of unsupervised learning that has gained relevance in the recent years, is *self-supervised learning*, which are algorithms where the input data itself provides the supervision. In self-supervised learning a proxy task is defined so that the algorithm learns valuable representations for the actual downstream task.

In *semi-supervised learning* (SSL), a large amount of input data is typically available, but only some of it is labeled. Taking again the notation from [24], in SSL the data set $X = \big((x_i)_{i\in[n]}\big)$ can be divided into two parts: the data $X_l := (x_1, ..., x_l)$ with the corresponding labels $Y_l := (y_1, ..., y_l)$, and the data $X_u := (x_{l+1}, ..., x_{l+u})$ without labels, or with some weak constraint. In order to address semi-supervised learning, supervised and unsupervised techniques can be combined.

2.3.2 Weak Supervision Learning

Full-supervision refers to the setup where all labels are available during training. In recent years machine learning and deep learning have seen an impressive growth, but these models are dependent on large hand-labeled datasets in order to be trained on a fully-supervised way. These annotations are expensive and time-consuming to collect, especially when domain expertise is required. Therefore, the trend is to advance in leveraging coarser labels that are easier to obtain, or to directly work with unlabeled data by training the models in a semi-supervised or unsupervised learning setup (introduced in the previous Section 2.3.1.

In this Section we introduce the concept of **weak supervision**, which is central is this dissertation. Weak supervision [199] aims at reducing the annotation cost required when training a model. Weakly-supervised methods can either rely on low-quality labels that can be acquired efficiently, or directly leverage unlabeled data. Following the classification from [199], there are three different types of weak supervision paradigms:

- **Incomplete supervision:** only a subset of training data is labeled. There are two main techniques for this purpose, i.e., *active learning* [158] and *semi-supervised learning* [200, 24]. *Active Learning* [158] assumes that there is a human expert, i.e. oracle, than can be queried to get ground-truth labels for selected unlabeled instances. The idea is to minimize human intervention (and thus the labeling cost) by minimizing the number of queries. Active Learning selects the most valuable unlabeled instances to be strongly-annotated by the oracle. On the other hand, *Semi-Supervised Learning* (SSL) automatically exploits a large amount of unlabeled/weakly-labeled data, and a limited amount of strongly-labeled samples, without any human intervention.

- **Inexact supervision:** in this case only coarse-grained labels are given. This scenario can be formally defined by *Multiple Instance Learning (MIL)* [45]. In this formulation, training instances are arranged in sets that are called *bags*. The learner receives a label for each entire bag, but not for the individual instances. Formally, the task is to learn the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ from a training data set $D = \{(X_1, y_1), ..., (X_m, y_m)\}$ where $X_i = \{x_{i1}, ...x_{i,m_i}\} \subseteq \mathcal{X}$ is called a *bag*, $x_{ij} \in \mathcal{X} (j \in \{1, ..., m_i\})$ is an instance, m_i is the number of instances in X_i, and y_i are the labels for the bags. X_i is a positive bag (and then has a positive y_i label) if there exists x_{ip} that is positive, while $p \in \{1, ..., m_i\}$ is unknown. The final goal is to predict labels for unseen bags. For the sake of understanding, in an object detection problem, inexact supervision could mean that only image-level labels are provided, i.e., information of which class categories of objects appear in the image but not the precise location. This can be expressed with MIL formulation, as we know that objects are located in regions of the image, but do not know exactly where. In this case a *bag* would be the whole image, and a positive *bag* means that at least one example in the bag is positive, i.e., at least one region in the image contains an object of a certain class category [192]. This type of supervision is also referred as *indirect supervision*.

- **Inaccurate supervision:** the training data has labels, but these are noisy or may suffer from mistakes, so they can not be always considered ground truth.

One manner to address it is to learn with label noise [51], for instance identifying mistakes and attempting to correct them.

2.3.3 Inference Supervision Scenarios

We introduced the different paradigms when training models in Section 2.3.2, and focused on weak supervision in Section 2.3.1. This Section describes different supervision scenarios at inference time. We distinguish between the following cases:

- **Unsupervised:** Unsupervised at inference refers to those tasks that do not require any effort from the human side at test time. For instance, when addressing image segmentation or instance segmentation, only an image is the input of the system, and no other cue is required.

  In this thesis, we present the first results for unsupervised at inference video object segmentation. We call this task **zero-shot video object segmentation**. A model addressing this task discovers objects along a video sequence without any initialization cue. In Zero-shot VOS the model must segment all those objects that appear and move in the scene. This task is interesting as it does not require any effort from the end-user at inference mode. An example application is video-surveillance, in which any object that appears and moves in the scene must be detected, without any kind of supervision at test time. Another application is video editing; zero-shot VOS could be useful to select all elements that appear in a video without any extra supervision.

- **Semi-Supervised:** Semi-Supervised at inference refers that an example of the task is given at test time.

  In this thesis, we address semi-supervised inference for video object segmentation, which is a very popular task in video object segmentation. This task is also called **one-shot video object segmentation**. Given a video, the user has to provide pixel-level masks of the objects to be tracked for the first frame of the video, so that the model can produce segmentation masks for the rest of the sequence.

- **Weakly-Supervised:** Weakly-supervised at inference refers that a weak signal is given at test time.

  For instance, when addressing video object segmentation, instead of providing pixel-level masks for the first frame, scribbles or bounding boxes could be used to indicate which objects to segment. In this thesis, we address weakly-supervised at inference video object segmentation by using language to indicate the target objects.

2.4 Natural Language Processing Fundamentals

This section introduces concepts on Natural Language Processing which are important for Part III of the dissertation.

Natural Language Processing, also known as NLP, is the field of artificial intelligence that addresses the ability of machines to read, understand and derive meaning from human language. NLP is the core of many popular tasks: machine translation, speech recognition, sentiment analysis, question answering, text classification and dialogue systems. In

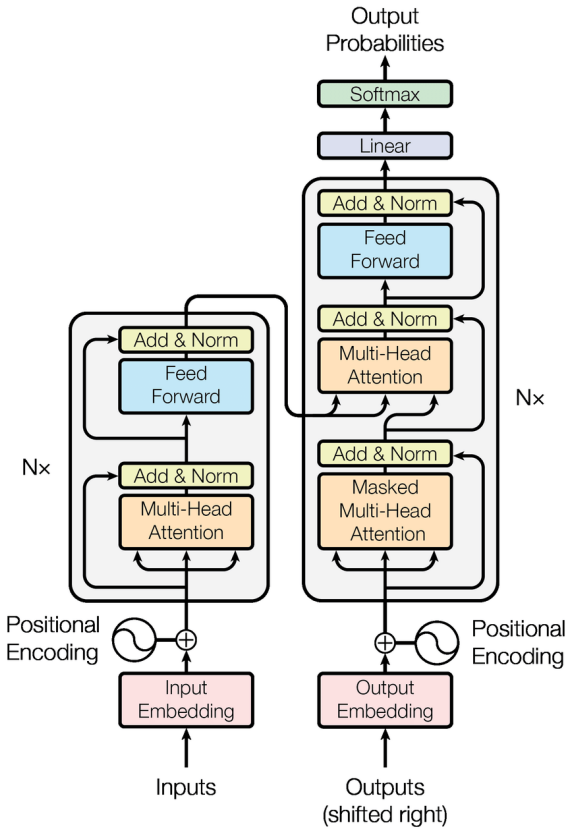

Figure 2.10: The Transformer model architecture. It consists of an encoder (left) and a decoder (right) composed of stacked multi-head attention modules and feed forward layers. A positional encoding is also used to gain notion of the input sequence order. This Figure and more details can be found in [168].

the next subsections, first the concept of word embeddings, crucial to represent natural language, is introduced (Sec. 2.4.1). Following, in Section 2.4.2 we briefly describe the Transformer model, which plays a central role in the advancement of NLP tasks. In Section 2.4.3 we introduce models for language modeling based on Transformers.

2.4.1 Word Embeddings

One of the tools most used in NLP are word embeddings, which are low-dimensional representations of words that can be used for many NLP tasks. The first word embedding that employed neural networks was word2vec [115], which was trained to reconstruct linguistic context of words. A posterior work, GloVe [128], built on top of the same idea. The resulting word representations from word2vec or GloVe allow for latent semantic analysis, and they are still very popular nowadays. The disadvantage of the two aforementioned methods, is that each word is represented by the same vector regardless of its context. As a matter of a fact, in NLP context is very relevant, as a word can have different meanings depending on it. This initiated research on contextualized word embeddings, such as ELMo [132], which instead of using fixed embeddings for each word, it processes the entire sentence before assigning an embedding to each of its words by using a bi-directional LSTM [69]. ELMo was trained to address Language Modeling, i.e., to predict the next word in a sequence of words.

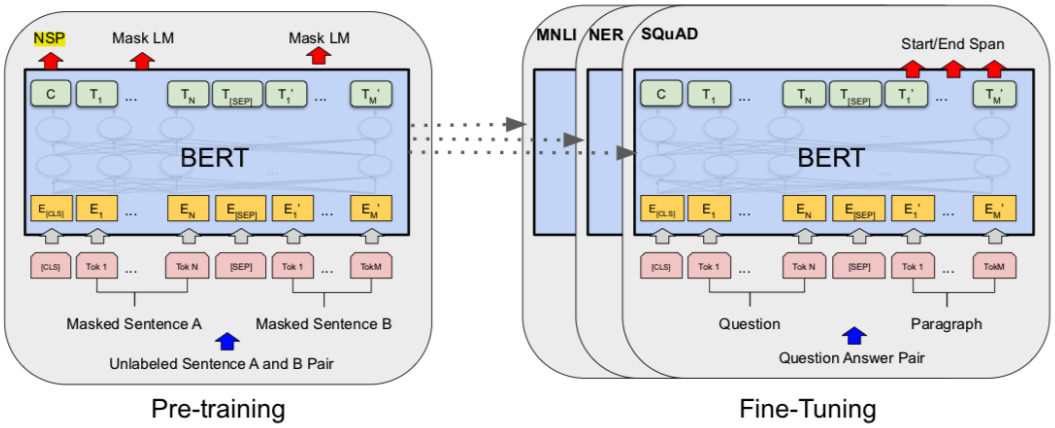

Figure 2.11: BERT pipeline for pre-training the language model in the left, and for fine-tuning for other tasks in the right. *NSP* refers to Next Sentence Prediction task, whereas *Mask ML* refers to Mask Language Model. When pre-training the language model, the different tokens of the sentence are encoded and then the decoder is optimized for the two aforementioned objectives. When fine-tuning, the architecture is the same, but in this case the loss optimized depends on the given task (SQuaD, NER and MNLI are some of the NLP tasks from the GLUE benchmark [172]). Something worth noting, is that [CLS] is a special token added in front of every input example, and that [SEP] is a special separator token (e.g. separating questions/answers). This Figure and more details can be found in [44].

2.4.2 The Transformer

The Transformer [168] (Figure 2.10) represents an alternative to recurrent models, as they can deal with long-term dependencies better than LSTMs [69]. The Transformer exploits attention mechanisms that learn contextual relations between words (or sub-words) in a text. The architecture is composed of an encoder and a decoder, in which the encoder processes the text input, and the decoder produces a prediction for the task. The Transformer became popular as it outperformed previous RNN-based models for machine translation.

2.4.3 Language Modeling with Transformers

The OpenAI GPT Transformer [137] was the first to leverage the decoder of the Transformer model for language modeling, which could later be used for other downstream tasks, such as question answering or translation. It is a left-to-right architecture, so that every token can only attend to previous tokens in the self-attention layers of the Transformer.

Following, Bidirectional Encoder Representations from Transformers (BERT) [44] (Figure 2.11) overcame the unidirectionality of GPT. BERT adopts the *Mask Language Model* (MLM) task to train their model. MLM objective loss consists in masking a percentage of words from the input sentences, and train the model to predict the original value. BERT is also pre-trained to address the *Next Prediction task*, that is to predict if, given two sentences, how likely it is that one follows the other.

Part I

Supervised Learning for Video Segmentation

3 Recurrent Video Object Segmentation

3.1 Introduction

Part I of this dissertation focuses on fully-supervised methods for instance segmentation that are end-to-end trainable. Specifically, we present a novel architecture for video object segmentation, i.e., the task of discovering objects throughout a video sequence, that is based on Recurrent Neural Networks (RNNs). This work was a joint collaboration with Dr. Carles Ventura from the Universitat Oberta de Catalunya (UOC).

Video object segmentation (VOS) aims at separating object pixels from the background in a video sequence. This task has raised a lot of interest in the computer vision community since the appearance of benchmarks [130] that have given access to annotated datasets and standardized metrics. Recently, new benchmarks [136, 182] that address multi-object segmentation and provide larger datasets have become available, leading to more challenging tasks.

Most works addressing VOS treat frames independently [18, 171, 110, 34], and do not consider the temporal dimension to gain coherence between consecutive frames. Some works have leveraged the temporal information using optical flow estimations [35, 77, 165, 9] or propagating the predicted masks through the video sequence [129, 184].

In contrast to these works, some methods propose to train models on spatio-temporal features, e.g., [165] used RNNs to encode the spatio-temporal evolution of objects in the video sequence. However, their pipeline relies on an optical flow stream that introduces extra computation and prevents a fully end-to-end trainable model. A posterior work [181] proposed an encoder-decoder architecture based on RNNs that is similar to our proposed pipeline. The main difference is that they process only a single object in an end-to-end manner. Thus, a separate forward pass of the model is required for each object that is present in the video. None of these models consider multi-object segmentation in a unified manner.

We present an architecture (see Figure 3.1) that serves for several video object segmentation scenarios (single-object vs. multi-object). In our model for VOS we adapt RSIS [154], a model for recurrent semantic instance segmentation, by adding recurrence in the temporal domain to predict instances for each frame of the sequence. RSIS was developed at the initial stage of this PhD in collaboration with Dr. Amaia Salvador, and although it is not a contribution of this thesis, it is a fundamental tool used in Part I and Part II of this dissertation.

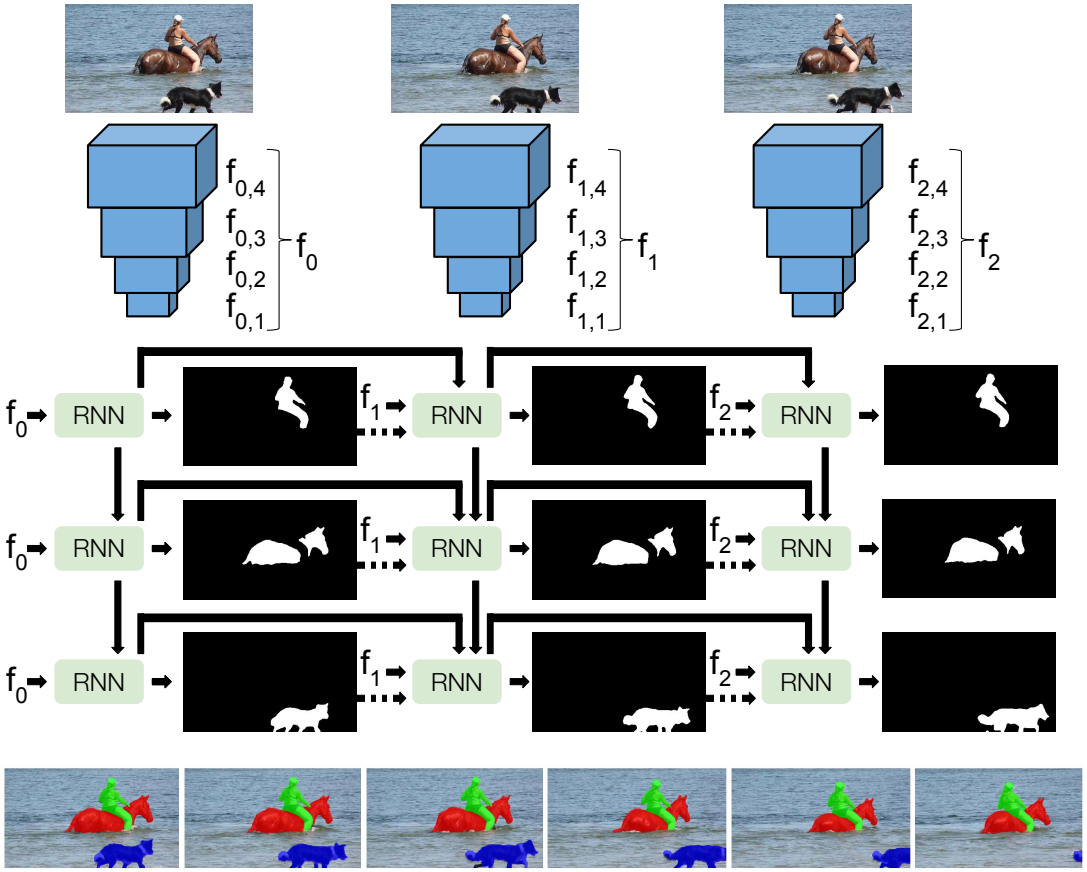

Figure 3.1: Our proposed architecture where RNN is considered in both spatial and temporal domains. We also show qualitative results where each predicted instance mask is displayed with a different color. At the top of the Figure, there are the different frames being fed to the encoder. Next, the recurrent decoder processes the features extracted from the encoder. In this Figure we can see the flow of information between the different time steps, and the different instances within a single frame.

The fact that our proposed method is recurrent in the spatial (the different instances of a single frame) and the temporal (different frames) domains allows that the tracking of instances at different frames can be handled naturally by the network. For the spatial recurrence, we force that the ordering in which multiple instances are predicted is the same across temporal time steps. Thus, our model is a fully end-to-end solution, as we obtain multi-object segmentation for video sequences without any post-processing.

The main task addressed in VOS in the recent years has been one-shot segmentation, also known as semi-supervised VOS. In this setup, the system receives a pixel-level mask of each object to be segmented for the first frame of the video sequence at inference time, and the goal is to predict the mask for the following frames of the video. Our architecture can also handle the more challenging task of zero-shot learning for VOS (also known as unsupervised VOS in a posterior challenge to our work, DAVIS-2019[1], although we prefer to name it unsupervised at *inference*). In this case, no initial masks are given, and the model must discover segments along the sequences. We present quantitative results

[1] https://davischallenge.org/challenge2019/unsupervised.html

Figure 3.2: Example of *one-shot* and *zero-shot* VOS. In *one-shot* VOS, at inference time the first frame comes with the annotations of the different objects. The task is to segment those objects throughout the following frames of the sequences. In *zero-shot* VOS no annotations are provided, and the algorithm has to segment the different objects without any initial reference. This video sequence belongs to DAVIS 2017 benchmark [136]. These segmentation masks correspond to the ground truth provided by the benchmark.

for zero-shot learning for two benchmarks: DAVIS-2017 [136] and YouTube-VOS [182]. Furthermore, we can easily adapt our architecture for one-shot VOS by feeding the objects masks from previous time steps to the input of the recurrent network.

For clarity, we want to highlight that in this context both one-shot and zero-shot VOS are trained fully-supervised, i.e., all masks are available during training time. The one-shot and zero-shot terms refer to the number of frames annotated during test time, being the first case semi-supervised and the latter unsupervised at inference. The two tasks are illustrated in Figure 3.2. Refer for more details about the different supervision scenarios to Section 2.3 of the Technical Background Section.

Our contributions can be summarized as follows: (a) We present the first end-to-end architecture for video object segmentation that tackles multi-object segmentation and does not need any post-processing, (b) our model can easily be adapted to one-shot and zero-shot scenarios, and we present the first quantitative results for zero-shot video object segmentation for the DAVIS-2017 and Youtube-VOS benchmarks [136, 182], (c) we outperform previous VOS methods which do not use online learning. Our model achieves a remarkable performance without needing finetuning for each test sequence, becoming the fastest method.

3.2 Related Work

Deep learning techniques for the video object segmentation task have gained attention in the research community during the recent years [18, 171, 129, 184, 35, 78, 165, 73,

160, 87, 164, 77, 89, 74, 178, 154]. In great measure, this is due to the emergence of new challenges and segmentation datasets, from Berkeley Video Segmentation Dataset (2011) [5], SegTrack (2013) [93], Freiburg-Berkeley Motion Segmentation Dataset (2014) [118], to more accurate and dense labeled ones as DAVIS (2016-2017) [130, 136], to the latest segmentation dataset YouTube-VOS (2018) [181], which provides the largest amount of annotated videos up to date.

Video object segmentation: When considering the temporal dimension of video sequences, we differentiate between algorithms that aim to model the temporal dimension of an object segmentation through a video sequence, and those without temporal modeling that predict object segmentations at each frame independently.

For segmentation without temporal modeling, one-shot VOS has been handled with online learning, where the first annotated frame of the video sequence is used to fine-tune a pretrained network and segment the objects in other frames [18]. Some approaches have worked on top of this idea, by either updating the network online with additional high confident predictions [171], or by using the instance segments of the different objects in the scene as prior knowledge and blend them with the segmentation output [110]. Others have explored data augmentation strategies for video by applying transformations to images and object segments [84], tracking of object parts to obtain region-of-interest segmentation masks [34], or meta-learning approaches to quickly adapt the network to the object mask given in the first frame [184].

To leverage the temporal information [35, 77, 165, 117] depend on pretrained models on other tasks (e.g. optical flow or motion segmentation). Subsequent works [9] use optical flow for temporal consistency after using Markov random fields based on features extracted from a Convolutional Neural Network.

An alternative to gain temporal coherence is to use the predicted masks in the previous frames as guidance for next frames [129, 184, 73, 79]. In the same direction, [78] propagate information forward by using spatio-temporal features. Whereas these works cannot be trained end-to-end, we propose a model that relies on the temporal information and can be fully trained end-to-end for VOS. Finally, S2S [181] makes use of an encoder-decoder recurrent neural network structure, that uses Convolutional LSTMs for sequence learning. One difference between our work and S2S, is that our model is able to handle multiple objects in a single forward pass by including spatial recurrence, which allows the object being segmented to consider previously segmented objects in the same frame.

One and zero-shot video object segmentation: In video object segmentation, one-shot learning is understood as making use of a single annotated frame (often the first frame of the sequence) to estimate the segmentation of the remaining frames in the sequence. On the other hand, zero-shot or unsupervised learning is understood as building models that do not need an initialization to generate segmentation masks of objects in the video sequence.

There are several works in the literature that rely on the first mask as input to propagate it through the sequence [18, 171, 129, 184, 78, 165, 73]. In general, one-shot methods reach better performance than zero-shot ones, as the initial segmentation is already given, thus not having to estimate the initial segmentation mask from scratch. Most of these models rely on online learning, i.e. adapting their weights given an initial frame and its corresponding masks. Typically online learning methods reach better results, although

they require more computational resources for the fine-tuning. In our case, we do not rely on any form of online learning or post-processing to generate the prediction masks.

In zero-shot learning, in order to estimate the segmentation of the objects in an image, several works have exploited object saliency [160, 77, 74], leveraged the outputs of object proposal techniques [87] or used a two-stream network to jointly train with optical flow [35]. Exploiting motion patterns in videos was studied in [164], while [89] formulates the inference of a 3D flattened object representation and its motion segmentation. A foreground-background segmentation based on instance embeddings was proposed in [95]. Our model is able to handle both zero and one-shot cases. In Section 3.4 we show results for both configurations, tested on the Youtube-VOS [182] and DAVIS-2017 [136] datasets. For one-shot VOS our model has not been fine-tuned with the mask given at the first frame. Furthermore, on the zero-shot case, we do not use any pretraining on detection tasks or rely on object proposals. This way, our model can be fully trained end-to-end for VOS, without depending on models that have been trained for other tasks.

End-to-end training: Regarding video object segmentation we distinguish between two types of end-to-end training. A first type of approach is frame-based and allows end-to-end training for multiple-objects [171, 110]. A second group of models allow training in the temporal dimension in an end-to-end manner, but deal with a single object at a time [181], requiring a forward pass for each object and a post-processing step to merge the predicted instances.

To the best of our knowledge, at the time of publication RVOS was the first work that allowed a full end-to-end training given a video sequence and its masks, without requiring any kind of post-processing.

3.3 Model

We propose a model based on an encoder-decoder architecture to solve two different tasks for the video object segmentation problem: one-shot and zero-shot VOS. For one-shot VOS, the input consists of the set of RGB video frames, as well as the masks of the first appearance of each object. For the zero-shot VOS, the input only consists of the set of RGB video frames. In both cases, the output consists of a sequence of masks for each object in the video, with the difference that the objects to segment are unknown in the zero-shot VOS task.

The architecture is based on RSIS [154], a recurrent model for semantic instance segmentation, that produces sequences of binary masks that cover the different objects within the image. In order to design RVOS, we extended RSIS by adding recurrence to the temporal dimension. RSIS architecture is explained in details in Section 3.3.1, followed by a detailed description of the encoder and decoder for video object segmentation in Sections 3.3.2 and 3.3.3 respectively.

3.3.1 Recurrent Semantic Instance Segmentation

This Section presents the architecture of RSIS [154], a recurrent neural network to address Recurrent Semantic Instance Segmentation. RSIS is the predecessor of RVOS, presented in this Part of the thesis. Furthermore, RSIS is used in Part II to explore image segmentation in low supervision scenarios. Instance segmentation is the task that,

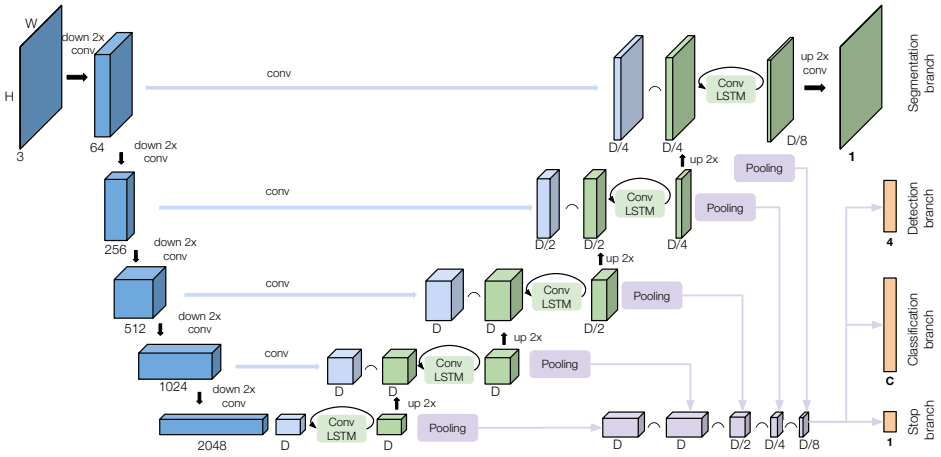

Figure 3.3: RSIS, Recurrent architecture for semantic instance segmentation. Figure from [154]. Our proposed architecture for VOS, namely RVOS, builds upon RSIS.

given an image, a set of binary masks with a corresponding class label must be provided. RSIS addresses this task by generating a sequence of binary masks that cover the objects within the image, similarly to previous works on instance segmentation [145, 142].

RSIS is composed of an encoder-decoder architecture (Figure 3.3), similarly to the one used in semantic segmentation works [106, 147]. The pipeline has skip connections from the encoder to the decoder to preserve the low-level details, which are central for segmentation tasks. The encoder receives as input the RGB image, and it is a Resnet-101 [67] architecture pretrained on Imagenet [152] for object classification. The last pooling layer of the Resnet-101 is removed to keep a higher resolution at the bottleneck of RSIS. The decoder receives as inputs features at different stages from the encoder part, and is composed of several Convolutional LSTMs [178] layers. The architecture is hierarchical, as it gradually increases the resolution until matching the input image size, in order to produce a binary mask for an object contained in the image at each time step. The segmentation is trained with a soft intersection over union loss (sIoU) between the predicted and the ground truth mask. During training, the order for the objects to be predicted is not imposed. In contrast, the model can decide whichever order it prefers. The Hungarian algorithm is exploited to determine which is the permutation of the ground truth masks that better matches the sequence predicted by the network. Once each predicted mask is matched with a ground truth one, the loss can be computed for the whole sequence.

Additionally, an aggregation of multi-resolution features taken from different stages of the decoder are concatenated and connected to three parallel fully-connected layers with their corresponding losses. The first one is a classification branch to obtain the semantic category of the object being segmented at the current time step, that is trained with categorical cross entropy. The second, is a detection branch, and it must predict a bounding box for the segment. Adding the detection loss improved the performance of the segmentation task. The loss employed to train the detection branch is the mean squared error between the predicted and the ground truth bounding boxes. Lastly, a third branch predicts whether there are more objects in the image or not. This branch is named stop branch, and must predict a '1' when there are objects within the image, and a '0' when all objects have been covered. Hence, the model is able to predict when there are no more objects to segment, and for this reason there is no need of a post-processing

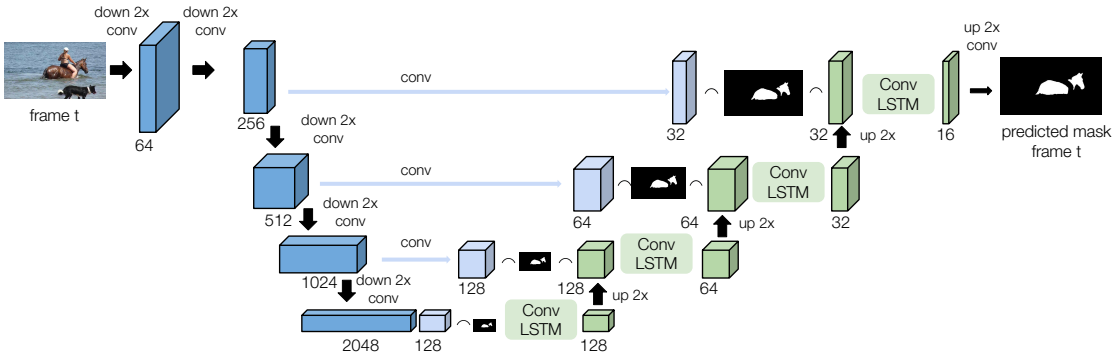

Figure 3.4: Our proposed recurrent architecture for video object segmentation for a a single frame at time step t for the one-shot case. The figure illustrates a single forward of the decoder, predicting only the first mask of the image. Notice that for the zero-shot case, the channel with the mask from the previous frame is not added.

step. The stop loss is trained with a binary cross entropy objective.

3.3.2 Encoder for VOS

The encoder for RVOS is based on the encoder from RSIS, explained in the previous Section 3.3.1. The input x_t of the encoder is an RGB image, which corresponds to frame t in the video sequence, and the output $f_t = \{f_{t,1}, f_{t,2}, ..., f_{t,k}\}$ is a set of features at different resolutions. The architecture of the encoder is illustrated as the blue part (on the left) in Figure 3.4. We propose two different configurations: (i) an architecture that includes the mask of the instances from the previous frame as one additional channel of the output features (as showed in the figure), and (ii) which preserves the original architecture from RSIS, i.e. without the additional channel. The inclusion of the mask from the previous frame is especially designed for the one-shot VOS task, where the first frame masks are given.

3.3.3 Decoder for VOS

The green blocks on the right of Figure 3.4 depict the decoder architecture for a single frame and a single step of the spatial recurrence. The RVOS decoder is designed as a hierarchical recurrent architecture of ConvLSTMs [178] which can leverage the different resolutions of the input features $f_t = \{f_{t,1}, f_{t,2}, ..., f_{t,k}\}$, where $f_{t,k}$ are the features extracted at the level k of the encoder for the frame t of the video sequence, just as in RSIS (described in Section 3.3.1), but removing the skip connections from the earliest layer of the encoder to reduce the memory requirements. The output of the decoder is a set of object segmentation predictions $\{S_{t,1}, , ..., S_{t,i}, ..., S_{t,N}\}$, where $S_{t,i}$ is the segmentation of object i at frame t. The recurrence in the temporal domain has been designed so that the mask predicted for the same object at different frames has the same index in the spatial recurrence. For this reason, the number of object segmentation predictions given by the decoder is constant (N) along the sequence. This way, if an object i disappears in a sequence at frame t, the expected segmentation mask for object i, i.e. $S_{t,i}$, will be empty at frame t and the following frames. We do not force any specific order in the spatial recurrence for the first frame. Instead, we find the optimal assignment

between predicted and ground truth masks with the Hungarian algorithm using the soft Intersection over Union score as cost function, as in RSIS.

Figure 3.5 depicts the difference between having only spatial recurrence, over having spatial and temporal recurrence. The output $h_{t,i,k}$ of the k-th ConvLSTM layer for object i at frame t depends on the following variables: (a) the features f_t obtained from the encoder from frame t, (b) the preceding $k-1$-th ConvLSTM layer, (c) the hidden state representation from the previous object $i-1$ at the same frame t, i.e. $h_{t,i-1,k}$, which will be referred to as the *spatial hidden state*, (d) the hidden state representation representation from the same object i at the previous frame $t-1$, i.e. $h_{t-1,i,k}$, which will be referred to as the *temporal hidden state*, and (e) the object segmentation prediction mask $S_{t-1,i}$ of the object i at the previous frame $t-1$:

$$h_{input} = [\, B_2(h_{t,i,k-1}) \mid f'_{t,k} \mid S_{t-1,i} \,] \tag{3.1}$$

$$h_{state} = [\, h_{t,i-1,k} \mid h_{t-1,i,k} \,] \tag{3.2}$$

$$h_{t,i,k} = \text{ConvLSTM}_\text{k}(\, h_{input} \,,\, h_{state} \,) \tag{3.3}$$

where B_2 is the bilinear upsampling operator by a factor of 2 and $f'_{t,k}$ is the result of projecting $f_{t,k}$ to have lower dimensionality via a convolutional layer.

Equation 3.3 is applied in chain for $k \in \{1, ..., n_b\}$, being n_b the number of convolutional blocks in the encoder. $h_{t,i,0}$ is obtained by considering

$$h_{input} = [\, f'_{t,0} \mid S_{t-1,i} \,]$$

and for the first object, h_{state} is obtained as follows:

$$h_{state} = [\, Z \mid h_{t-1,i,k} \,]$$

where Z is a zero matrix that represents that there is no previous spatial hidden state for this object.

In Section 3.4, an ablation study will be performed in order to analyze the importance of spatial and temporal recurrence in the decoder for the VOS task.

3.4 Experiments

The experiments are carried out for two different tasks of the VOS: the one-shot and the zero-shot. In both cases, we analyze how important the spatial and the temporal hidden states are. Thus, we consider three different options: (i) spatial model (temporal recurrence is not used), (ii) temporal model (spatial recurrence is not used), and (iii) spatio-temporal model (both spatial and temporal recurrence are used). In the one-shot VOS, since the masks for the objects at the first frame are given, the decoder always considers the mask $S_{t-1,i}$ from the previous frame when computing h_{input} (see Eq. 3.1). On the other hand, in the zero-shot VOS, $S_{t-1,i}$ is not used since no ground truth masks are given.

The experiments are performed in two VOS benchmarks: YouTube-VOS [182] and DAVIS-2017 [136]. YouTube-VOS consists of 3,471 videos in the training set and 474

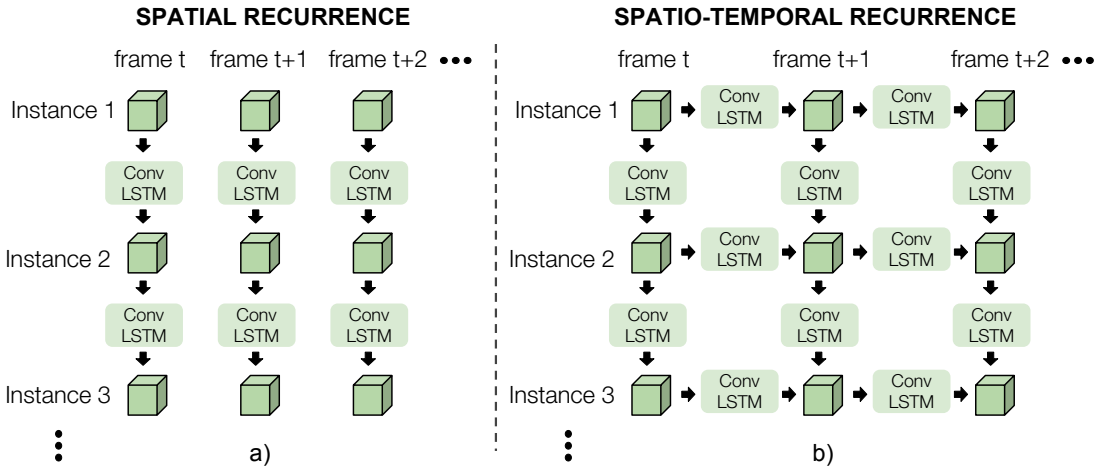

Figure 3.5: Comparison between original spatial RSIS [154] (left) and proposed spatio-temporal recurrent networks (right).

videos in the validation set, being the largest video object segmentation benchmark. The training set includes 65 unique object categories which are regarded as seen categories. In the validation set, there are 91 unique object categories, which include all the seen categories and 26 unseen categories. On the other hand, DAVIS-2017 consists of 60 videos in the training set, 30 videos in the validation set and 30 videos in the test-dev set. Evaluation is performed on the YouTube-VOS validation set and on the DAVIS-2017 test-dev set. Both YouTube-VOS and DAVIS-2017 videos include multiple objects and have a similar duration in time (3-6 seconds).

The experiments are evaluated using the standard evaluation measures for VOS used in the aforementioned benchmarks: (i) the region similarity J, and (ii) the contour accuracy F. In YouTube-VOS, each of these measures is split into two different measures, depending on whether the categories have already been seen by the model (J_{seen} and F_{seen}), i.e. these categories are included in the training set, or the model has never seen these categories (J_{unseen} and F_{unseen}). Note that this distinction between seen and unseen categories applies for both one-shot and zero-shot segmentation.

3.4.1 One-shot video object segmentation

One-shot VOS consists in segmenting the objects from a video given the objects masks from the first frame. Since the initial masks are given, the experiments have been performed including the mask of the previous frame as one additional input channel in the ConvLSTMs from our decoder.

YouTube-VOS benchmark: Table 3.1 shows the results obtained in YouTube-VOS validation set for different configurations: spatial (RVOS-Mask-S), temporal (RVOS-Mask-T) and spatio-temporal (RVOS-Mask-ST). All models from this ablation study have been trained using a 80%-20% split of the training set. We can see that the spatio-temporal model improves both the region similarity J and contour accuracy F for seen and unseen categories over the spatial and temporal models. Figure 3.6 shows qualitative results comparing the spatial and the spatio-temporal models, where we can see that RVOS-Mask-ST preserves better the segmentation of the objects along the time.

	YouTube-VOS one-shot			
	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
RVOS-Mask-S	54.7	37.3	57.4	42.4
RVOS-Mask-T	59.9	39.2	63.1	45.6
RVOS-Mask-ST	60.8	**44.6**	63.7	50.3
RVOS-Mask-ST+	**63.1**	44.5	**67.1**	**50.4**

Table 3.1: Ablation study about spatial and temporal recurrence in the decoder for one-shot VOS in YouTube-VOS dataset. Models have been trained using 80%-20% partition of the training set and evaluated on the validation set. + means that the model has been trained with a curriculum of first using the ground truth masks and then the inferred masks.

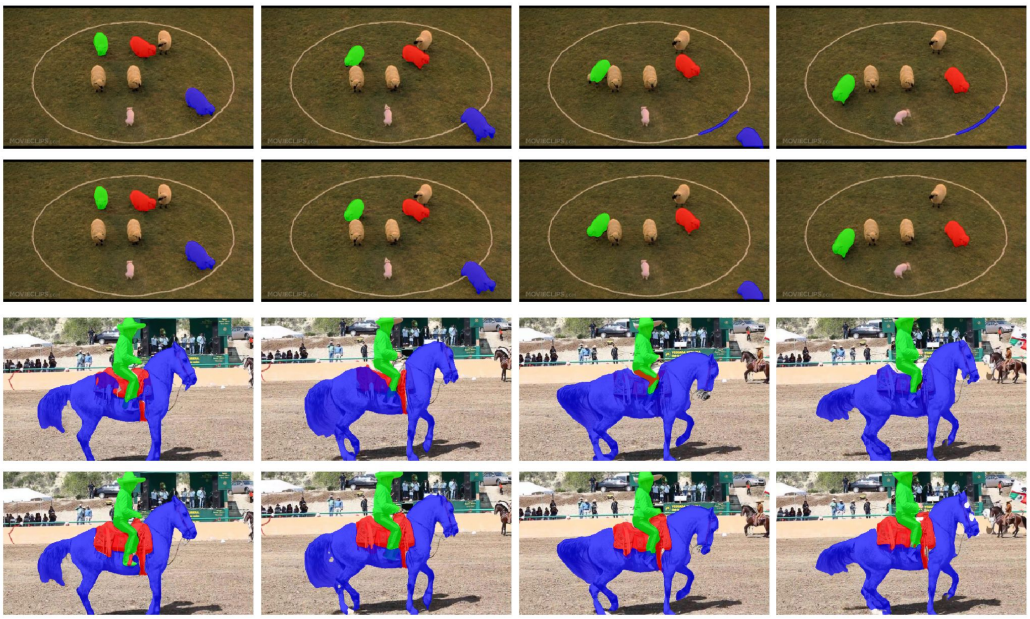

Figure 3.6: Qualitative results comparing spatial (rows 1,3) and spatio-temporal (rows 2,4) models.

Furthermore, we have also considered doing a curriculum of training the models some additional epochs using the inferred mask from the previous frame $\hat{S}_{t-1,i}$, instead of using the ground truth mask $S_{t-1,i}$. This way, the model can learn how to fix some errors that may occur in inference. In Table 3.1, we can see that this model (RVOS-Mask-ST+) is more robust and outperforms the model trained only with the ground truth masks. Figure 3.7 shows some qualitative results comparing the model trained with the ground truth mask and the model trained with the inferred mask.

Once stated that the spatio-temporal model is the one that gives the best performance, we have trained it using the whole YouTube-VOS training set to compare it with other state-of-the-art techniques (see Table 3.2). Our proposed spatio-temporal model (RVOS-Mask-ST+) has comparable results with respect to S2S w/o OL [182], with a slightly worse performance in region similarity J but with a slightly better performance in contour accuracy F. At the time of publication our model outperformed all previous works

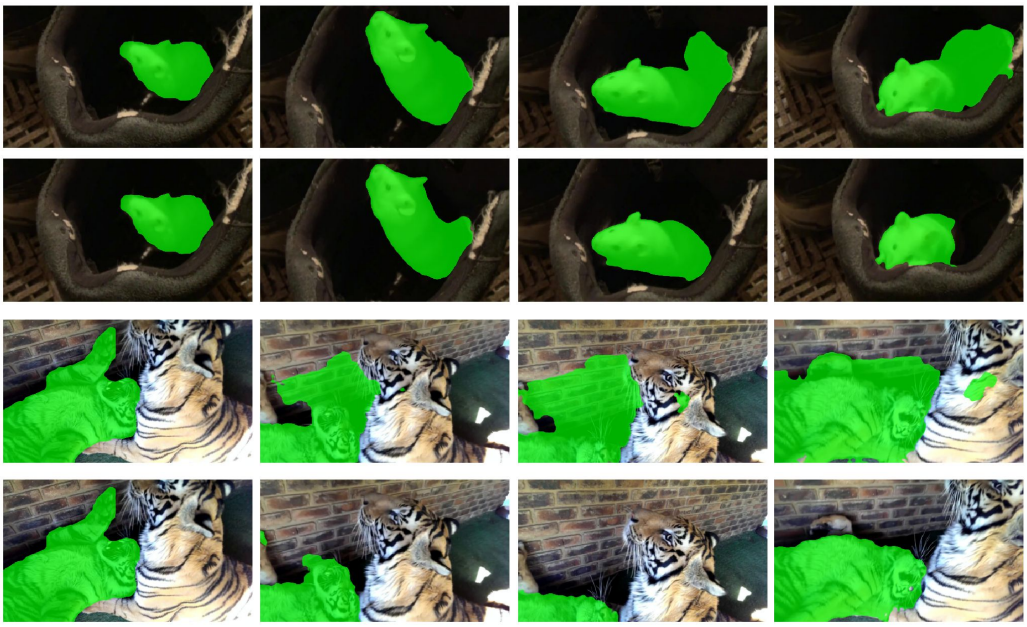

Figure 3.7: Qualitative results comparing training with ground truth masks (rows 1,3) and training with inferred masks (rows 2,4).

		YouTube-VOS one-shot			
	OL	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
OSVOS [18]	✓	59.8	**54.2**	60.5	**60.7**
MaskTrack [129]	✓	59.9	45.0	59.5	47.9
OnAVOS [171]	✓	**60.1**	46.6	**62.7**	51.4
OSMN [184]	✗	60.0	40.6	60.1	44.0
S2S w/o OL [182]	✗	**66.7**	**48.2**	65.5	50.3
RVOS-Mask-ST+	✗	63.6	45.5	**67.2**	**51.0**

Table 3.2: Comparison against state of the art VOS techniques for one-shot VOS on YouTube-VOS validation set. OL refers to online learning. The table is split in two parts, depending on whether the techniques use online learning or not.

[18, 129, 184, 171] for the *seen* categories, while OSVOS [18] reached the best performance for the *unseen* categories. However, note that the comparison of S2S without online learning [182] and our proposed model with respect to OSVOS [18], OnAVOS [171] and MaskTrack [129] is not fair for J_{unseen} and F_{unseen} because OSVOS, OnAVOS and MaskTrack models are fine-tuned using the annotations of the first frames from the validation set, i.e. they use online learning. Therefore, *unseen* categories should not be considered as such since online models have actually seen them.

Table 3.3 shows the results on the region similarity J and the contour accuracy F depending on the number of instances in the videos clips. We can see that the fewer the objects to segment, the easier the task, obtaining the best results for sequences where only one or two objects are annotated.

Figure 3.8 shows some qualitative results of our spatio-temporal model for different se-

	Number of instances (YouTube-VOS)				
	1	2	3	4	5
J mean	78.2	62.8	50.7	50.2	56.3
F mean	75.5	67.6	56.1	62.3	66.4

Table 3.3: Analysis of our proposed model RVOS-Mask-ST+ depending on the number of instances per video in one-shot VOS.

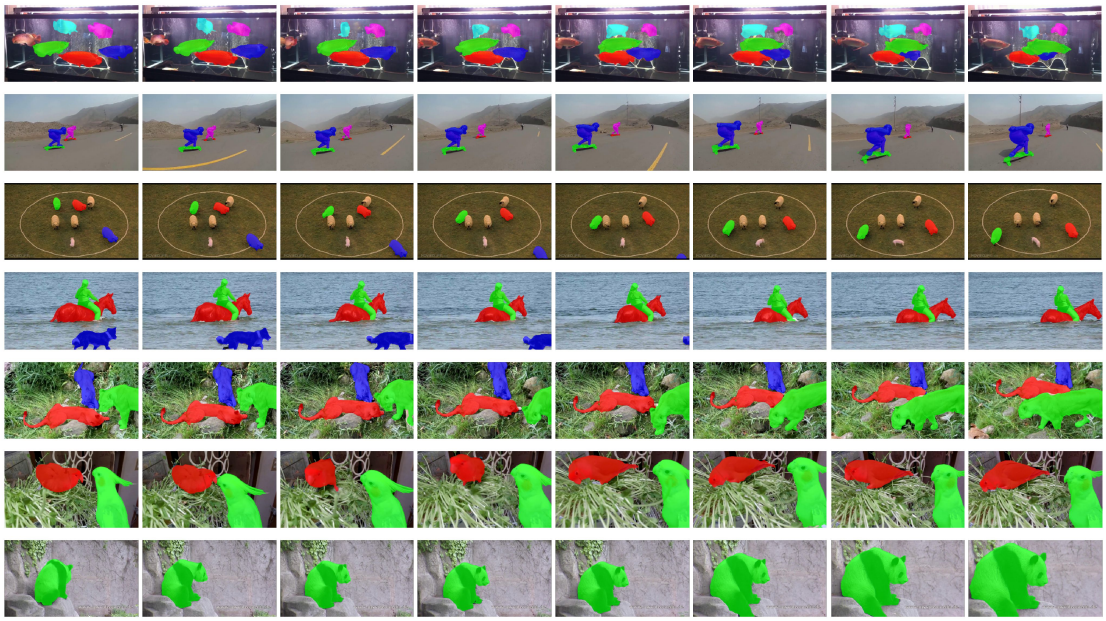

Figure 3.8: Qualitative results for one-shot video object segmentation on YouTube-VOS with multiple instances.

quences from YouTube-VOS validation set. It includes examples with different number of instances per clip. Note that the instances have been properly segmented although there are different instances of the same category in the sequence (fishes, sheeps, people, leopards or birds) or there are some instances that disappear from the sequence (one sheep in third row or the dog in fourth row).

DAVIS-2017 benchmark: Our pretrained model RVOS-Mask-ST+ in YouTube-VOS was also tested on a different benchmark: DAVIS-2017. As it can be seen in Table 3.4, when the pretrained model is directly applied to DAVIS-2017, RVOS-Mask-ST+ (pre) outperforms the rest of state-of-the-art techniques that do not make use of online learning, i.e. OSMN [184] and FAVOS [34]. Furthermore, when the model is further finetuned for the DAVIS-2017 training set, RVOS-Mask-ST+ (ft) outperforms some techniques as OSVOS [18], which is among the techniques that make use of online learning. Note that online learning requires finetuning the model at test time.

Figure 3.9 shows some qualitative results obtained for DAVIS-2017 one-shot VOS. As depicted in some qualitative results for YouTube-VOS, RVOS-Mask-ST+ (ft) is also able to deal with objects that disappear from the sequence.

		DAVIS-2017 one-shot	
	OL	J	F
OSVOS [18]	✓	47.0	54.8
OnAVOS [171]	✓	49.9	55.7
OSVOS-S [110]	✓	52.9	62.1
CINM [9]	✓	**64.5**	**70.5**
OSMN [184]	✗	37.7	44.9
FAVOS [34]	✗	42.9	44.2
RVOS-Mask-ST+ (pre)	✗	46.4	50.6
RVOS-Mask-ST+ (ft)	✗	**48.0**	**52.6**

Table 3.4: Comparison against state of the art VOS techniques for one-shot VOS on DAVIS-2017 test-dev set. OL refers to online learning. The model RVOS-Mask-ST+(pre) is the one trained on Youtube-VOS, and the model RVOS-Mask-ST+ (ft) is after fine-tuning the model for DAVIS-2017. The table is split in two parts, depending on whether the techniques use online learning or not.

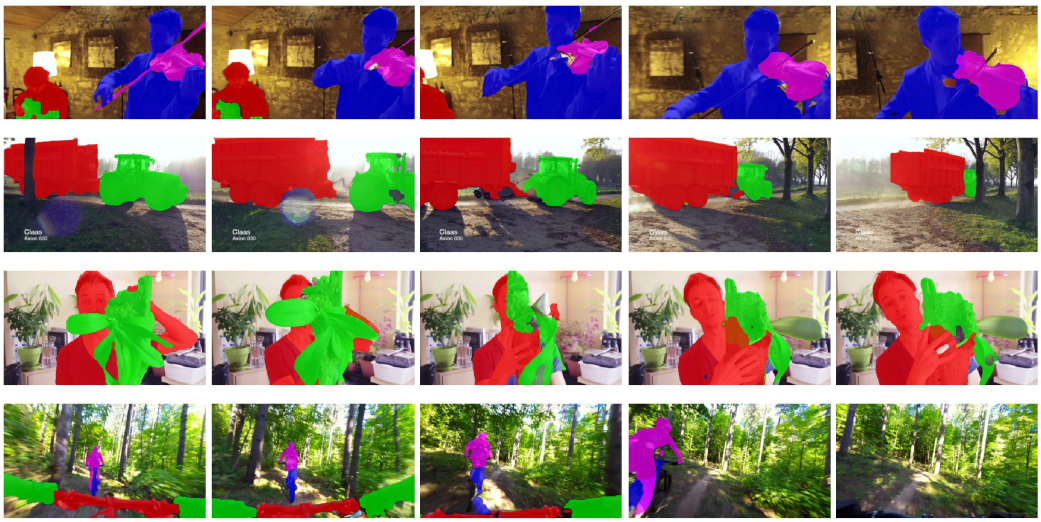

Figure 3.9: Qualitative results for one-shot on DAVIS-2017 test-dev.

3.4.2 Zero-shot video object segmentation

Zero-shot VOS consists in segmenting the objects from a video without having any prior knowledge about which objects should be segmented, i.e. no object masks are provided at inference time. This task is more complex that the one-shot VOS since the model has to detect and segment the objects appearing in the video.

Previous to the publication of this work, there was no benchmark specially designed for zero-shot VOS. Although YouTube-VOS and DAVIS benchmarks can be used for training and evaluating the models without using the annotations given at the first frame, both benchmarks had the limitation that not all objects appearing in the video were annotated. Specifically, in YouTube-VOS, there are up to 5 object instances annotated per video. This makes sense when the objects to segment are given (as done in one-shot VOS), but it may be a problem for zero-shot VOS since the model could be segmenting correctly

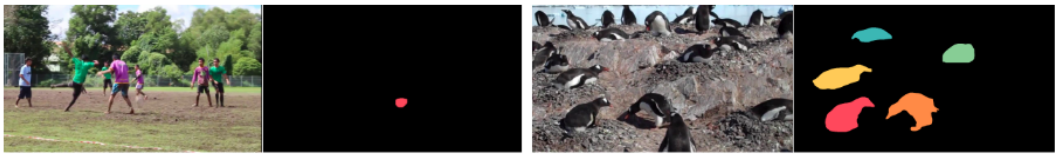

Figure 3.10: Missing object annotations may suppose a problem for zero-shot video object segmentation.

	YouTube-VOS zero-shot			
	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
RVOS-S	40.8	19.9	43.9	23.2
RVOS-T	37.1	20.2	38.7	21.6
RVOS-ST	**44.7**	**21.2**	**45.0**	**23.9**

Table 3.5: Ablation study about spatial and temporal recurrence in the decoder for zero-shot VOS in YouTube-VOS dataset. Our models have been trained using 80%-20% partition of the training set and evaluated on the validation set.

objects that have not been annotated in the dataset. Figure 3.10 shows a couple of examples where there are some missing object annotations.

Despite the problem stated before about missing object annotations, we trained our model for the zero-shot VOS problem using the object annotations available in these datasets. To minimize the effect of segmenting objects that are not annotated and missing the ones that are annotated, we allow our system to discover up to 10 object instances along the sequence, expecting that the up to 5 annotated objects are among the predicted ones. During training, each annotated object is uniquely assigned to one predicted object in order to to compute the loss. Therefore, those predicted objects that have not been assigned do not result in any loss penalization. However, an erroneous prediction of any annotated object is considered by the loss. Analogously, in inference, in order to evaluate our results for zero-shot video object segmentation, the masks provided for the first frame in one-shot VOS are used to select which predicted instances are selected for evaluation. Note that the assignment is only performed at the first frame and the predicted segmentation masks considered for the rest of the frames are the corresponding ones.

YouTube-VOS benchmark: Table 3.5 shows the results obtained on YouTube-VOS validation set for the zero-shot VOS problem. As stated for the one-shot VOS problem, the spatio-temporal model (RVOS-ST) also outperforms both spatial (RVOS-S) and temporal (RVOS-T) models.

Figure 3.11 shows some qualitative results for zero-shot VOS in YouTube-VOS validation set. Note that the masks are not provided and the model has to discover the objects to be segmented. We can see that in many cases our spatio-temporal model is temporal consistent although the sequence contains different instances of the same category.

DAVIS-2017 benchmark: Previous to the publication of our work in CVPR 2019, there were no published results for this task in DAVIS-2017 to be compared. The zero-shot VOS had only been considered for DAVIS-2016, where some unsupervised techniques

Figure 3.11: Qualitative results for zero-shot video object segmentation on YouTube-VOS with multiple instances.

had been applied. However, in DAVIS-2016, there is only a single object annotated for sequence, which could be considered as a foreground-background video segmentation problem and not as a multi-object video object segmentation. Our pretrained model RVOS-ST on Youtube-VOS for zero-shot, when it is directly applied to DAVIS-2017, obtains a mean region similarity $J = 21.7$ and a mean contour accuracy $F = 27.3$. When the pretrained model is fine-tuned for the DAVIS-2017 trainval set achieves a slightly better performance, with $J = 23.0$ and $F = 29.9$.

Although the model has been trained on a large video dataset as Youtube-VOS, there are some sequences where the object instances have not been segmented from the beginning. The low performance for zero-shot VOS in DAVIS-2017 ($J = 23.0$) can be explained due to the bad performance also in YouTube-VOS for the *unseen* categories ($J_{unseen} = 21.2$). Therefore, while the model is able to segment properly categories which are included among the YouTube-VOS training set categories, e.g. persons or animals, the model fails when trying to segment an object class that has not been seen before. Note that it is specially for these cases when online learning becomes relevant, since it allows to finetune

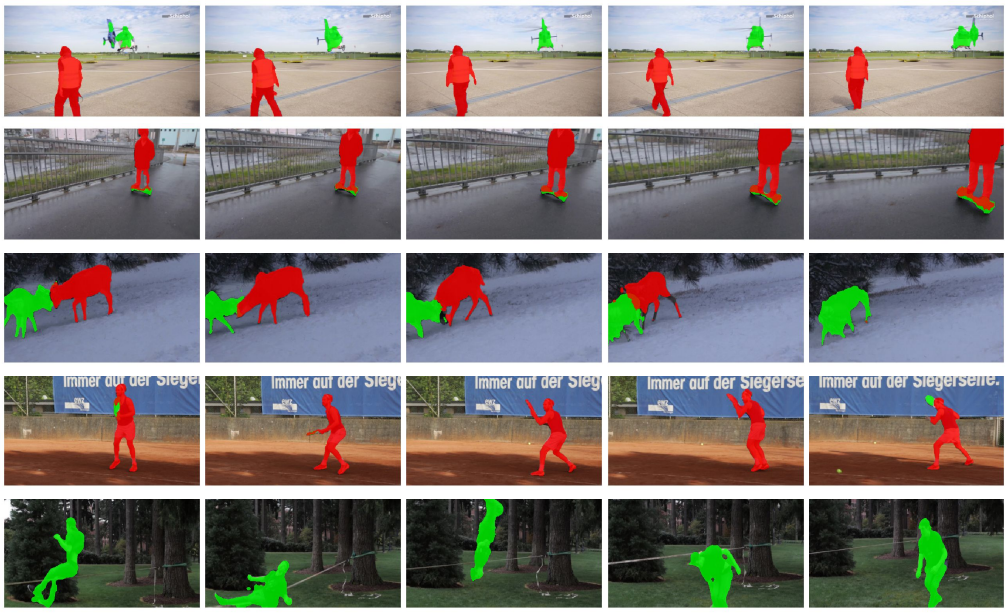

Figure 3.12: Qualitative results for zero-shot video object segmentation on DAVIS-2017 with multiple instances.

the model by leveraging the object mask given at the first frame for the one-shot VOS problem. Figure 3.12 shows some qualitative results for the DAVIS-2017 test-dev set where no object mask is provided but the RVOS-ST model has been able to segment the multiple object instances appearing in the sequences.

3.4.3 Runtime analysis

At the time of publication, RVOS was the fastest method amongst all while achieving comparable segmentation quality with respect to previous state-of-the-art works as seen previously in Tables 3.2 and 3.4. The inference time for RVOS is 44ms per frame with a GPU P100 and 67ms per frame with a GPU K80. Methods not using online learning (including ours) are two orders of magnitude faster than techniques using online learning. Inference times for OSMN [184] (140ms) and S2S [182] (160ms) have been obtained from their respective papers. For a fair comparison, we also compute runtimes for OSMN [184] in our machines (K80 and P100) using their public implementation (no publicly available code was found for [182]). We measured better runtimes for OSMN than those reported in [184], but RVOS was still faster in all cases (e.g. 65ms vs. 44ms on a P100, respectively). To the best of our knowledge, our method was the first to share the encoder forward pass for all the objects in a frame, which explains its fast overall runtime.

3.5 Training Details

The original RGB frames and annotations were resized to 256×448 in order to have a fair comparison with S2S [181] in terms of image resolution. In training, due to memory restrictions, each training mini-batch was composed with 4 clips of 5 consecutive frames. However, in inference, the hidden state is propagated along the whole video. Adam optimizer is used to train our network and the initial learning rate is set to 10^{-6}. Our

model was trained for 20 epochs using the previous ground truth mask and 20 epochs using the previous inferred mask in a single GPU with 12GB RAM, taking about 2 days.

3.6 Conclusions

In this Part we have presented RVOS, a fully end-to-end trainable model for multiple objects in video object segmentation (VOS) with a recurrence module based on spatial and temporal domains. The model has been designed for both one-shot and zero-shot VOS, and tested on YouTube-VOS and DAVIS-2017 benchmarks.

The experiments show that RVOS trained with spatio-temporal recurrence improves over considering the spatial or the temporal domain only. In our work we presented the first results for zero-shot VOS on both benchmarks and we also outperformed previous works that do not make use of online learning for one-shot VOS on them. Posterior to our work, the DAVIS-2019 benchmark [19] introduced the unsupervised VOS Challenge (in our terminology, zero-shot VOS), and used our architecture RVOS as baseline for the new challenge.

Subsequent works to RVOS have improved performance for VOS by exploiting space-time memory networks in order to read relevant information from the past frames [119, 157], or by training spatio-temporal embeddings to cluster pixels belonging to a specific object instance over the entire video sequence [7]. However, we want to highlight that one of the main advantages of our model, is that it is very fast in inference, being ours the fastest of all these aforementioned methods as pointed out in [7]. The code is available in our project website https://imatge-upc.github.io/rvos/.

Posterior to our work, the interest in end-to-end architectures for object localization and segmentation has caused great developments in the field, such as the recent work DETR [21], which leverages Transformers [168] to build a flexible pipeline for end-to-end object detection. As in our work, they cast the object detection task as an image-to-set problem, arguing that then the network can reason about the image as a whole. In our case, our network is built with Recurrent Neural Networks. We believe that an interesting topic for future research with end-to-end architectures for object localization, is to focus on videos, as we did with RVOS, but exploiting the advancements with Transformer architectures.

To conclude, the neural networks from Part I have been trained in a fully-supervised setup, i.e., all training data is annotated at a pixel level. Furthermore, we explored two different setups at inference for video object segmentation: the semi-supervised one (one-shot case) and the unsupervised one (zero-shot). In Part II, we will explore how to train segmentation models with low annotation costs for training.

Part II

Semi-supervised Learning for Image Segmentation

Introduction

Image segmentation solutions have traditionally been trained in a fully-supervised setup, i.e., dense annotations at a pixel level are required for all the training set. These annotations represent a bottleneck as they demand significant human labeling effort. Previous works have leveraged weak forms of supervision, such as bounding boxes or scribbles, in order to train segmentation models [180, 98, 163, 10, 124, 40, 83]. However, these methods do not meet the accuracy of their fully-supervised counterparts. A trade-off between performance and annotation budget can be achieved with semi-supervised pipelines, which take advantage of a limited amount of strongly-labeled samples, and a large amount of unlabeled/weakly-labeled data. In this Part of the thesis we explore semi-supervised pipelines for semantic and instance segmentation, and show how semi-supervised methods are capable of reaching better performance than other weakly-supervised methods given a fixed annotation cost.

In this Part of the dissertation we present our contributions on algorithms for weak supervision in an incomplete supervised setup, i.e., when only a subset of the data is annotated. First, Chapter 4 presents a semi-supervised pipeline to leverage weakly-annotated/unlabeled data together with a limited amount of strongly-annotated samples. Our method surpasses previous works at very low annotation budgets. We report results for both semantic and instance segmentation. Secondly, Chapter 5 introduces a novel manner to perform active learning to query which samples to strongly-annotate for the pipeline presented in Chapter 4, proving that our strategy improves random selection. In this second Chapter we focus exclusively on instance segmentation on still images.

4 Budget-aware Semi-Supervised Segmentation

4.1 Introduction

In computer vision, current state-of-the-art models based on Convolutional Neural Networks are data-hungry, and their performance is related to the amount of annotated data available for training. In particular, segmentation annotations are very costly, as they require a label for each pixel of the image. Therefore, there is a growing interest in training segmentation models that do not rely on a high annotation budget but still achieve a competitive performance.

For semantic and instance segmentation, the use of weak labels as a cheaper supervision signal to train segmentation models has been extensively explored in the literature. Some of the most popular weak supervision signals are image-level labels [176, 194, 3, 198] or bounding boxes [40, 124, 83, 94, 195]. Although the results are promising, they are still far from the performance of methods that rely on stronger supervision.

Another option to lower the annotation cost are semi-supervised scenarios, where a small subset of the data is strongly annotated, and the remaining samples are unlabeled/weakly-labeled. The most successful semi-supervised methods handle heterogeneous annotations (few strong and a huge amount of weak labels) and, although they reach higher performance [124, 70, 176], their annotation cost is much higher than the one related to only using weak labels, such as image-level labels or bounding boxes.

The goal of weakly and semi-supervised methods is to obtain segmentation results that are competitive with their fully-supervised counterparts, while requiring a much lower annotation cost. However, previous works do not typically compare to each other in terms of the annotation budget. In this Chapter, we argue that when the goal is to minimize human effort, methods should be compared considering the annotation cost, in terms of time required to annotate the training data, regardless of the type of annotation they use ('costly' annotations such as pixel-level masks, or 'cheaper' ones such as image-level labels). In this direction, [10] proposed a comparison between weakly- and fully-supervised semantic segmentation methods that contemplates the total annotation time required for the training set. We extend this analysis including semi-supervised methods that rely on unlabeled data and also, for the first time, for the instance segmentation task. This will allow a unified analysis across different supervision setups and different supervision signals, comparing the total annotation time when fixing a certain budget.

In this Chapter, we present a semi-supervised scheme trained with low annotation budgets

that reaches significantly better performance than methods trained with weak labels while having the same annotation cost. Our proposed pipeline consists of two networks: a first annotation model that generates pseudo-annotations for the unlabeled or weakly-labeled data, and a second segmentation model that is trained with both the strong and pseudo-annotations (Figure 4.1). In order to lower the annotation budget, first we work with strong and unlabeled data, so that only strong annotations have an associated annotation cost. With only a few strong annotations, we reach higher performance than previous weakly and semi-supervised approaches for both semantic and instance segmentation, at much reduced annotation budgets. We name our semi-supervised pipeline BASIS (from **B**udget **A**ware **S**emi-supervised semantic and **I**nstance **S**egmentation).

We also analyze heterogeneous annotations for instance segmentation, which combine both strong and weak labels. The weak label that we choose consists in counting the number of objects there are for each of the class categories of the dataset [54]. To the authors knowledge, this is the first time this weak supervision is used for instance segmentation. We propose to exploit weak labels by feeding them into the annotation network. As weak labels involve a cost, we adjust the number of samples to analyze different supervision scenarios. We find that, when the number of strongly-labeled samples is extremely reduced, this solution outperforms the standard semi-supervised pipeline.

Our contributions can be summarized as follows: (a) We unify the segmentation benchmarks regardless of the training setting and the supervision signals by comparing them in terms of the total annotation cost they require, (b) we outperform previous semi-supervised semantic segmentation methods at low annotation budgets for the Pascal VOC benchmark [48], and present the first quantitative results for semi-supervised instance segmentation for this dataset when no extra images are available, (c) we show that when fixing a low annotation budget, it is more convenient having fewer but stronger-labeled data over having larger weakly-annotated sets.

4.2 Related Work

In this Section we focus on the related work when aiming at lowering the annotation budget for image segmentation, by reducing the annotation time itself, or by leveraging weakly-supervised methods.

4.2.1 Reducing the annotation time

Many works have focused on how to obtain labels in a more rapid way. The time to annotate a bounding box has been narrowed from 35 seconds [162] to 7 per object using extreme clicking techniques [123]. Regarding pixel-wise annotations, several large-scale segmentation datasets have been collected using polygon annotations [48, 100, 38, 197] to reduce the annotation cost, but still require a significant amount of time per object (79 s/object for Microsoft COCO [100]). Other works automatically provide a segmentation from a bounding box [149, 26], but the final prediction is noisy and cannot be considered as strong supervision. Another option is leveraging interactive segmentation tools with a human in the loop, thus providing valid strong annotations at a reduced annotation time [114, 23, 2, 112]. Nevertheless, these tools typically rely on some already annotated set in order to train the automatic engine, which involves an extra cost that should be taken into account. In contrast to these works, our focus resides in the type of annotations

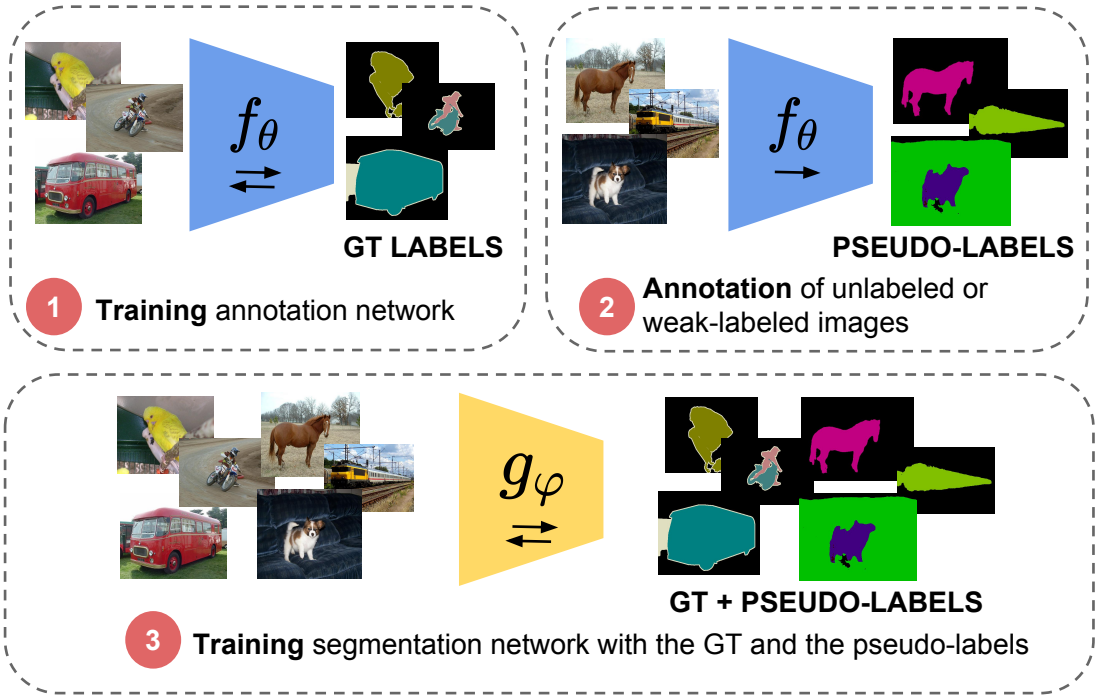

Figure 4.1: Our semi-supervised training pipeline consists of two networks, an annotation network trained with strong-labeled data, and a segmentation network trained with the union of pseudo-annotations and strong-labeled samples. Note that both networks are trained in a fully-supervised way.

being used, instead of the way to obtain them.

4.2.2 Image Segmentation with Synthetic Images

One option to obtain images and annotations for free, is to generate synthetic images with the corresponding per-pixel labels from virtual 3D environments [153, 33, 148], and train segmentation models with those. However, these methods are limited to domains with 3D environments available, such as urban scenes, and the generated synthetic images present a domain gap with real ones.

4.2.3 Image Segmentation with Weak Supervision

Several works in the literature have proposed to use weak supervision to reduce the annotation cost for image segmentation. In this review we will distinguish between those that fall into the category of *inexact supervision*, and those that use *semi-supervised learning*, a method when there is *incomplete supervision*.

4.2.3.1 Image Segmentation with Inexact Supervision

For semantic segmentation, one of the most popular forms to reduce the annotation cost is to leverage coarser labels, such as image-level labels, as they can be obtained with minimum human intervention. There are approaches that treat image-level labels with Multiple Instance Learning (MIL) techniques [133, 127, 126], but these works

achieve an accuracy far from their fully-supervised counterparts. Other works develop Expectation-Maximization (EM) methods to learn from weakly-annotated data [124]. Another pool of works have focused on localizing class-specific cues with Class Activation Maps (CAMs) [196] in order to mine regions [175, 76, 3, 176], while others obtain regions with attention mechanisms [194]. CAMs are a frequently exploited when only image-level labels are available, as they permit to localize class-specific regions within an image with a model trained only with image-level labels.

For semantic segmentation, other weak signals have been exploited, such as scribbles [180, 98, 163], points [10] or bounding boxes [124, 40, 83].

Few works have addressed weakly-supervised instance segmentation in computer vision. Bounding box labels have been exploited to recursively generate and refine pseudo-labels from a weak-labeled sets [83, 195, 94]. These methods typically rely on bottom-up segment proposals [134, 149]. In contrast with this approach, Remez et al. [140] propose an adversarial scheme that learns to segment without using any object proposal technique. Although these works tackle weakly-supervised instance segmentation, their weak supervision consists in using bounding boxes, thus their main challenge resides in how to separate the foreground from the background within a bounding box. The first work [198] that uses image-level supervision for weakly-supervised instance segmentation detects peaks of Class Activation Maps [196], producing what they identify as Peak Response Maps (PRMs). With them they generate a query to retrieve the best candidate among a set of pre-computed object proposals (MCG) [134]. Following works [90] build on PRMs by using the pseudo-masks to train Mask R-CNN [65] in a fully-supervised way, reaching better performance.

All these aforementioned works fall into the category of *inexact* weak supervision, as they exploit coarse-grained labels, without relying on any strongly-labeled image.

4.2.3.2 Image Segmentation with Semi-Supervised Learning

Semi-supervised learning allows to reduce the annotation burden while keeping a competitive performance. Some works that address weakly-supervised semantic segmentation with coarse labels present results for the semi-supervised case by combining their generated pseudo-annotations with a few strong labels [124, 40, 83, 176, 94]. Some other works exclusively tackle the semi-supervised scenario, as it is our case. Image-level labels are leveraged for semi-supervised semantic segmentation by [70]. Their pipeline consists of two separate networks, a classification and a segmentation network with bridged layers. They obtain remarkable results training with only a few strong annotations. A posterior work [71] proposes a new partially supervised training paradigm to combine bounding box annotations and pixel-level masks. To the authors knowledge, only [71, 94] have tackled semi-supervised instance segmentation at the time of publication of this work. However, these approaches assume a huge amount of weakly-labeled samples. In our work, we focus on low-budget scenarios, presenting the first results for semi-supervised instance segmentation for the Pascal VOC benchmark [48] with no extra images from other datasets.

Based on referent surveys [200, 24], the semi-supervised approach that we present in this Chapter falls into the category of a *self-learning* [155] method. It is also known as *self-training*, *self-teaching* or *bootstrapping*. This approach relies on supervised learning

methods. These methods first train a model with few strongly-labeled data, and use it to predict pseudo-labels for the unlabeled pool of samples. Following, the supervised method is retrained using its own predictions as additional labeled data. This process can be done repeatedly.

Concretely, our pipeline consists of two networks. The first one, named *annotation* network, is trained with a few strongly-annotated samples. Next, this network is used to obtain pseudo-labels for the unlabeled pool of samples. Following, a second network named *segmentation* network is trained with both the original strongly-annotated data, and the obtained pseudo-labels. From the literature reviewed in this Section, our model resembles to the work from [176]. Their pipeline consists of two networks as well, a deep neural network that produces pseudo-labels from CAMs, and a network that is trained with the obtained annotations. As our setup is semi-supervised, our first network is trained with strong supervision only, while the second network is trained with both pseudo- and strong annotations. The main difference is that in our case, we do not work with CAMs as pseudo-labels, but with segmentation predictions from the first network.

4.3 Benchmark for Budget-Aware Segmentation

The main focus of our work is to offer a unified analysis across different supervision setups and supervision signals for semantic and instance segmentation. Our motivation raises from the ultimate goal of weakly and semi-supervised techniques: the reduction of the annotation burden. We adopt the analysis framework from [10] and extend it to any supervision setup, applied to two different tasks: semantic and instance segmentation.

We estimate the annotation cost of an image from a well-known dataset for semantic and instance segmentation: the Pascal VOC dataset [48]. Our study considers four level of supervision: image-level, image-level labels + object counts, bounding boxes, and full supervision (i.e. pixel-wise masks). The estimated costs are inferred from three statistical figures about the Pascal VOC dataset drawn from [10]: a) on average 1.5 class categories are present in each image, b) on average there are 2.8 objects per image, and c) there is a total of 20 class categories. Hence, the budgets needed for each level of supervision are:

Image-Level (IL): According to [10], the time to verify the presence of a class in an image is of 1 second. The annotation cost per image is determined by the total number of possible class categories (20 in Pascal VOC). Then, the cost is of $t_{IL} = 20\,\text{classes/image} \times 1\text{s/class} = 20\,\text{s/image}$.

Image-Level + Counts (IL+C): IL annotations can be enriched by the amount of instances of each object class. This scheme was proposed in for weakly-supervised object localization [54], in which they estimate that the counting increases the annotation time to 1.48s per class. Hence, the time to annotate an image with image labels and counts is $t_{IL+C} = t_{IL} + 1.5\,\text{classes/image} \times 1.48\,\text{s/class} = 22.22\,\text{s/image}$.

Full supervision (Full): We consider the annotation time reported in [10] for instance segmentation: $t_{Full} = 18.5\,\text{classes/image} \times 1\text{s/class} + 2.8\,\text{mask/image} \times 79\,\text{s/mask} = 239.7\,\text{s/image}$. As we could not find any reference to the semantic segmentation task, we will assume that semantic segmentation labels require as much time as the instance segmentation ones.

	IL	**IL+C**	**Full**	**BB**
Cost (s/image)	20	22.22	239.7	38.1

Table 4.1: Average annotation cost per image when using different types of supervision. *IL* stands for image-level labels, *IL+C*, stands for image-level labels plus counts, *Full* refers to pixel-wise annotations, and *BB* stands for bounding box labels.

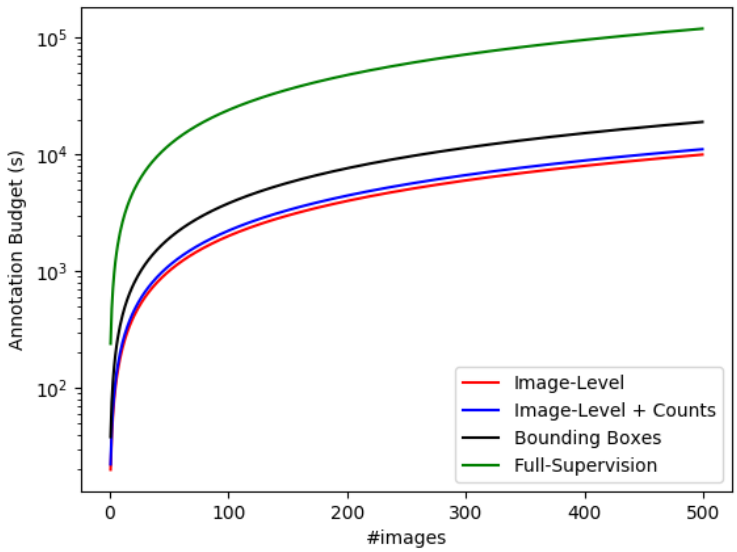

Figure 4.2: Annotation budget vs. number of images. The vertical axis is in logarithmic scale for a better visualization.

Bounding Boxes (BB): Recent techniques have cut the cost of annotating a bounding box to 7.0 s/box by clicking the most extreme points of the objects [123]. Following the same reasoning as for dense predictions, the cost of annotating a Pascal VOC image with bounding boxes is $t_{bb} = 18.5\,\text{classes/image} \times 1\text{s/class} + 2.8\,\text{bb/image} \times 7\,\text{s/bb} = 38.1\,\text{s/image}$.

Table 4.1 summarizes the average cost of the different supervision signals for a single Pascal VOC image.

Given a certain annotation budget, the amount of annotated images will depend on the chosen level of supervision. The lower the level of supervision, the more images will be annotated. Figure 4.2 shows the total cost of annotating a variable amount of images with different types of supervision. The central research question of our work is how to use an annotation budget: whether in few but fully supervised annotations, or in weaker labels for a larger amount of images.

4.4 BASIS

Our semi-supervised scheme BASIS is a self-learning methods and consists of two different networks. A first fully supervised model f_θ is trained with strong-labeled samples

	#Strong	#Unlabeled	val mIoU	test mIoU
DeepLab-v3+ Ours	∼1.4k		79.22	77.26
DeepLab-v3+ Ours	∼1.4k	∼9k	79.41	78.71
DeepLab-v3+ Ours	∼10k		80.42	80.29
DeepLab-v3+ [30]	∼10k		81.21	-

Table 4.2: Performance of DeepLab-v3+ for the validation and test set of Pascal VOC 2012 with different supervision setups.

from the ground truth $(X, Y) = \{(x_1, y_1), ..., (x_N, y_N)\}$, being N the total number of strong samples. The network f_θ is an annotation network used to predict pseudo-labels $Y' = \{y'_1, ..., y'_M\}$ for M unlabeled samples $X' = \{x'_1, ..., x'_M\}$. A second segmentation network g_φ is trained with $(X, Y) \cup (X', Y')$, as depicted in Figure 4.1. Depending on the task (semantic or instance segmentation), we will choose different architectures for the networks. It is important to remark that the proposed pipeline is independent to the network architecture used, so it is possible to leverage off-the-shelf networks.

We present experiments for both the semantic and instance segmentation tasks for the Pascal VOC 2012 benchmark [48]. The standard semi-supervised setup adopted for this dataset consists of using the Pascal VOC 2012 train images (1464 images) as strong-labeled images, and the Pascal VOC additional set (9118 images) [61] as unlabeled/weak-labeled. Note that in fact, all these images (1464 + 9118) are annotated with strong annotations, but for purposes of working on a semi-supervised setup, we will consider the second set as weakly-labeled by only working with the class category annotations.

In this section, we vary N to analyze the performance at different annotation budgets, and consider M to be the total size of the training dataset minus N ($M = 10582 - N$). Note that these M samples are unlabeled, free of annotation cost.

4.5 Semantic Segmentation

For semantic segmentation, we consider f_θ and g_φ to have the same architecture, a DeepLab-v3+ [30] with an Xception-65 [36] encoder, with output stride of 16 for both training and evaluation. We used the official TensorFlow implementation from [30]. Following the setup described in Section 4.4, we run experiments with the standard semi-supervised setup for Pascal VOC. Table 4.2 shows the results for different levels of supervision in terms of mean Intersection Over Union (mIoU). The first row sets the baseline of 79.22 when training the annotation network f_θ (a DeepLab-v3+) with only the 1.4k images from the Pascal VOC 2012 train set. The next row, reports a mIoU of 79.41 when we train g_φ, also a DeepLab-v3+, with both the strong-labels Y and the pseudo-labels Y' obtained with f_θ, which represents a small improvement. Finally, we trained a DeepLab-v3+ with all labels strongly-annotated (fully-supervised case), and obtained a mIoU of 80.42, close to the reference figure (81.21) reported in [30].

To assess the impact of fixing different annotation budgets, we trained several DeepLab-v3+ f_{θ_N} with a varying number of strong-labeled training samples $N \in \{100, 200, 400, 800, 1464\}$. These networks are used to obtain pseudo-annotations for the M samples without labels. Then, we train a corresponding g_{φ_N} for each f_{θ_N}. Notice that the pseudo-annotations are obtained for free, as no supervision signal is required.

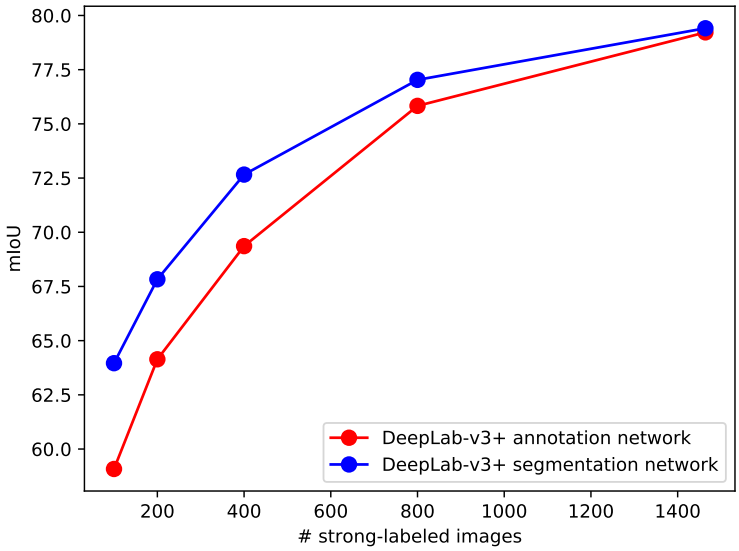

Figure 4.3: Semantic segmentation performance of the annotation and segmentation networks for an increasing budget for the validation set of Pascal VOC.

	100	**200**	**400**	**800**	**1464**
Annotation Network (AP th=0.5)	58.3	66.3	70.4	75.9	79.0
Segmentation Network (AP th=0.5)	64	67.8	72.7	77.0	79.4
Budget (days)	0.28	0.55	1,11	2.22	4.06

Table 4.3: Mean Intersection Over Union (mIoU) and annotation budgets when changing N for semantic segmentation.

Figure 4.3 plots the obtained mIoU by the annotation network f_{θ_N} and segmentation network g_{φ_N} for different annotation budgets. The same information is depicted in Table 4.3. We observe that, given a certain budget, the mIoU of g_{φ_N} is always higher than the one obtained with the f_{θ_N} alone, and therefore the extra pseudo-labels improve the performance. Notice that the gap between ef_{θ_N} and g_{φ_N} becomes larger for lower budgets, given that g_{φ_N} is trained with a larger proportion of images compared to its corresponding f_{θ_N}. This suggests that pseudo-annotations can increase the quality of the segmentation tool at no additional cost, being this increment more relevant for low budgets.

Figure 4.4 compares our results with recent works of both weakly-supervised that rely on coarse labels and semi-supervised approaches for semantic segmentation. The plot on the left shows the mIoU metric with respect to the annotation cost in days. We propose this analysis as a unified benchmark that allows a fair comparison between both weakly-supervised and semi-supervised pipelines. We observe that our results obtained with DeepLab-v3+ outperform all previous methods at same or lower annotation budgets, setting a new state of the art of 79.41 mIoU for semi-supervised segmentation at the time of publication, using strong supervision only. In order to compensate for the different network backbones used in the related works, Figure 4.4 (right) normalizes the mIoU

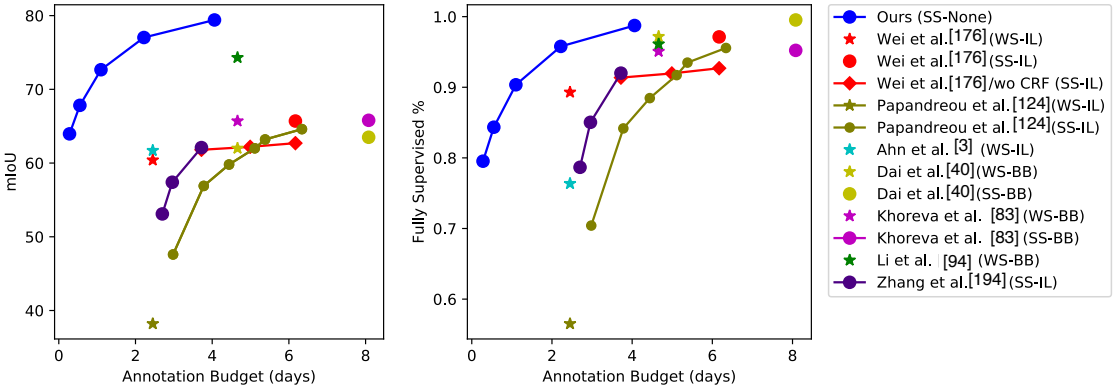

Figure 4.4: Semantic segmentation comparison, in terms of mean IoU and the normalized IoU considering the fully-supervised performance, for the validation set of Pascal VOC with other semi-supervised (SS) and weakly-supervised (WS) methods that use image-level labels (IL) or bounding box labels (BB). All methods are trained with the Pascal VOC dataset. Note that in this Figure we only show our results with the *segmentation* network (not the *annotation* one), as it is the one that gets better performance.

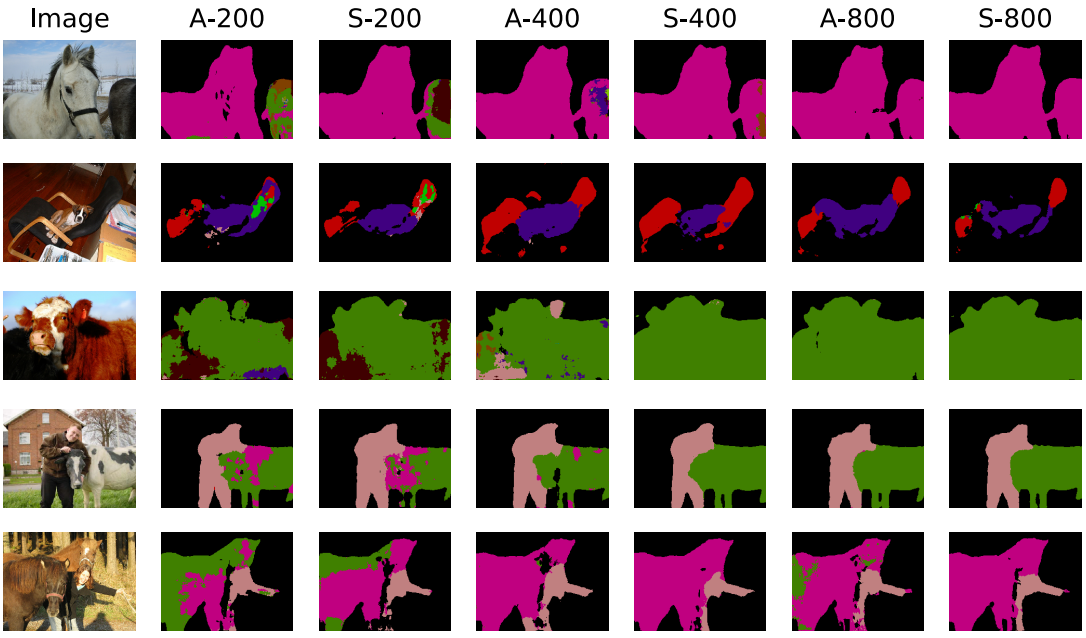

Figure 4.5: Visualization of Pascal VOC validation set for the annotation f_{θ_N} *(A-)* and segmentation networks g_{φ_N} *(S-)*, depending on the number of strong labels used $N \in \{200, 400, 800\}$. The color map legend is in Figure 4.6.

Figure 4.6: Pascal VOC color map for semantic segmentation [47]

.

	#Strong	#Unlabeled	val AP 50
RSIS Ours	∼1.4k		32.4
RSIS Ours	∼1.4k	∼9k	48.3
RSIS Ours	∼10k		56.4
RSIS [154]	∼10k		57.0

Table 4.4: Performance of RSIS for the validation set of Pascal VOC 2012 with different supervision setups. The difference between the result obtained in the original RSIS paper (*AP 50* of 57.0), and the result we got when re-training RSIS for this work (*AP 50* of 56.4), is due to variance in the model performance.

scores with the ones obtained by their fully-supervised counterparts. Our method with DeepLab-v3+ still reaches a closer number to the fully-supervised case compared to the other works at a fixed annotation budget. We want to highlight that our approach outperforms all methods that rely on weak labels when matching the annotation cost. Therefore, we conclude that it is preferable to invest the budget into collecting fewer fully supervised samples, than a larger amount of weakly-labeled ones. Figure 4.5 depicts some examples of semantic segmentation predicted by f_{θ_N} and g_{φ_N} when using different number of strong labels. As expected, g_{φ_N} obtains better segmentation results than its counterpart f_{θ_N}. We can also observe that at low annotation budgets ($N = 200$), the segments produced are able to accurately outline some contours, although the results are still far from the ones obtained with a higher N.

4.6 Instance Segmentation

The approach for instance segmentation follows the semi-supervised pipeline described in Section 4.5: training an annotation network f_θ and a segmentation network g_φ. This is the same scheme as in the semantic segmentation task presented in Section 4.5 but, in this case, we use the recurrent architecture for instance segmentation RSIS [154], described in Section 3.3.1 from Part I, for both f_θ and g_φ. The results in Table 4.4 show a similar behaviour to the semantic segmentation case from Table 4.2, although there is a more significant improvement of performance when the segmentation network g_{φ_N} is trained with $(X, Y) \cup (X', Y')$, the union of the strong-labeled set and the pseudo-annotated set. Figure 4.7 shows the Average Precision at 0.5 threshold for different budget scenarios, in which f_{θ_N} is trained with $N \in \{100, 200, 400, 800, 1464\}$ strong labels. The same information is depicted in Table 4.5. The setup is the same as in the semantic segmentation case of Section 4.5, but the performance gap between f_{θ_N} and g_{φ_N} is more significant for the instance segmentation task (Figure 4.7) than for the semantic segmentation one (Figure 4.3). In the later, both curves converge when all available 1464 strong labels are used to train the annotation network, which indicates that the segmentation network does not learn anything new from the unlabeled images. We hypothesize that learning instance segmentation is a more complex task, and more samples would be needed for both curves to converge.

Figure 4.8 compares our approach with related works that tackle weakly-supervised instance segmentation. For low annotation budgets there is the work from [198], that addresses weakly-supervised instance segmentation with image-level labels. This task is clearly very challenging for the instance segmentation problem, and we demonstrate

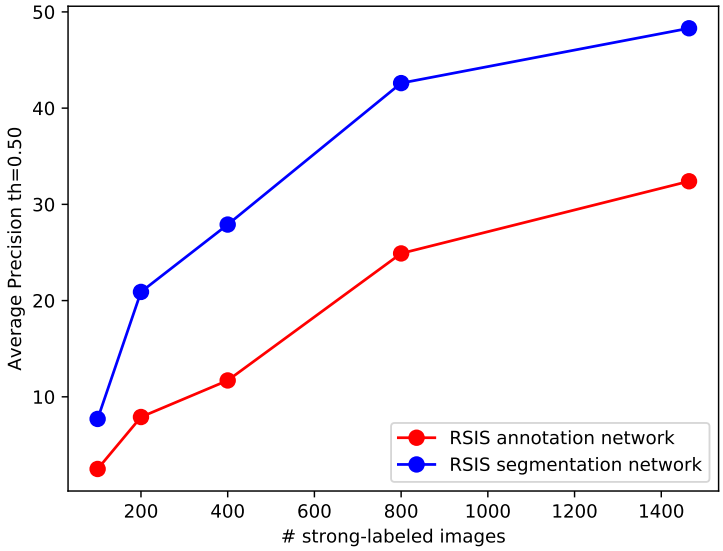

Figure 4.7: Instance segmentation performance of the annotation and segmentation networks for an increasing budget for the validation set of Pascal VOC.

	100	**200**	**400**	**800**	**1464**
Annotation Net. (AP th=0.5)	6.6	12.9	18.5	26.9	31.7
Segmentation Net. (AP th=0.5)	14.9	23.7	35.7	42.9	46.8
Budget (days)	0.28	0.55	1,11	2.22	4.06

Table 4.5: Average Precision (AP) at $th = 0.5$ and annotation budgets when changing N for instance segmentation.

that when matching the annotation cost, our semi-supervised approach reaches significant better performance. We believe that working with a semi-supervised setup for low-annotation budgets is convenient for instance segmentation, as cheap labels such as image-level ones barely relate to distinguishing between different instances of an image. Bounding boxes, on the other hand, scale down the problem to separate the foreground from the background, but at the cost of more expensive annotations and thus at higher budgets [83, 94]. Figure 4.9 depicts some examples predicted by the segmentation network g_{φ_N} when varying N. The higher the N, the better the network distinguishes between different instances.

4.6.1 Training with heterogeneous annotations

Heretofore, we have been assuming a semi-supervised setup where some samples are strongly-labeled and others are unlabeled. For instance segmentation, we observe in Figure 4.7 that the Average Precision for annotation networks f_θ trained with very few strong samples is very low (an annotation network trained with $N = 100$ reaches a low figure of 2.5 of AP). In this section we propose to use heterogeneous annotations, i.e., strong and weak annotations, instead of strongly-labeled samples alone. The main

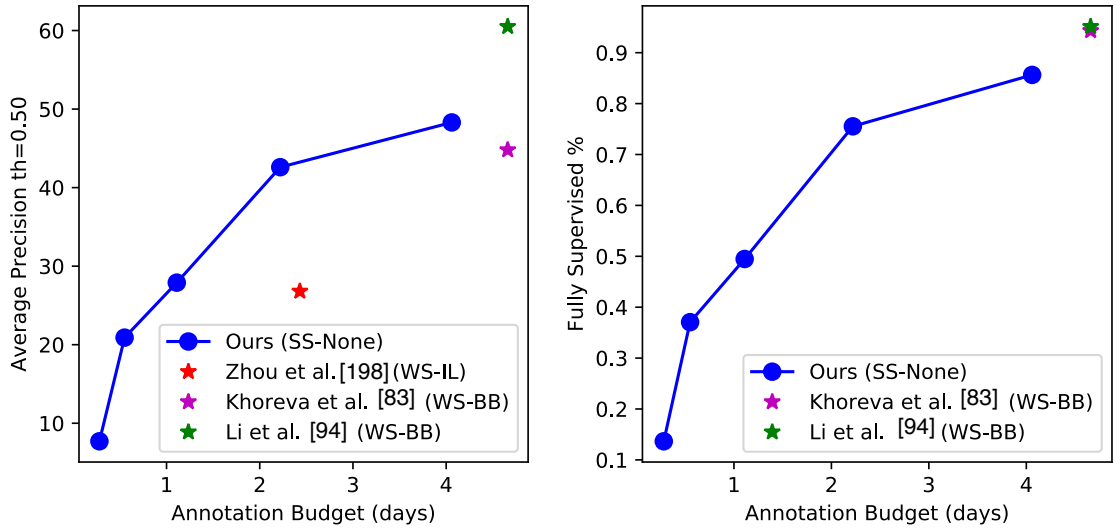

Figure 4.8: Instance segmentation comparison, in terms of Average Precision (AP) and the normalized AP considering the fully-supervised performance, for the validation set of Pascal VOC with other weakly-supervised (WS) methods that use image-level labels (IL) or bounding box labels (BB).

difference to our previous setup, is that now the annotation cost will come from two sources: the N strong-labeled samples, and the M weak-labeled ones.

As weak labels, we choose image-level labels, in addition to knowing how many instances of each class category appear in an image (IL+C). This supervision signal was first employed for weakly-supervised object localization [54], and its annotation cost is almost the same as using simply image-level labels, as explained in Section 4.3. To the best of our knowledge this is the first time that this supervision signal is used for instance segmentation. As strong annotations, as in the previous setup, we consider pixel-level annotations.

The setup is similar to the one explained in Section 4.4. For a better understanding, we will keep the same notation. Let Z be the IL+C labels for the strongly-annotated subset (X, Y, Z), and Z' the IL+C labels for the weakly-annotated subset (X', Z'). To exploit the weak-labels Z', now f_θ during training will receive as input (X, Z), and will be optimized to predict Y. In order to infer the pseudo-annotations Y', (X', Z') will be fed into f_θ. The segmentation network g_φ works as in Section 4.4. The architecture of the annotation network f_θ is a modified version of RSIS [154], described in Section 3.3.1 from Part I, and the architecture for the segmentation one corresponds to the original RSIS model.

Annotation network. RSIS consists in an encoder-decoder architecture (Figure 4.10). The encoder is a ResNet-101 [66], and the decoder is formed by a set of stacked ConvLSTM's [178]. At each time step, a binary mask and a class category for each object of the image is predicted by the decoder. The architecture also has a stop branch that indicates if all objects have been covered. The main property of this architecture is that its output does not need any post-processing (as it happens with proposal-based methods, where proposals need to be filtered), so that the pseudo-annotation is the output of the network

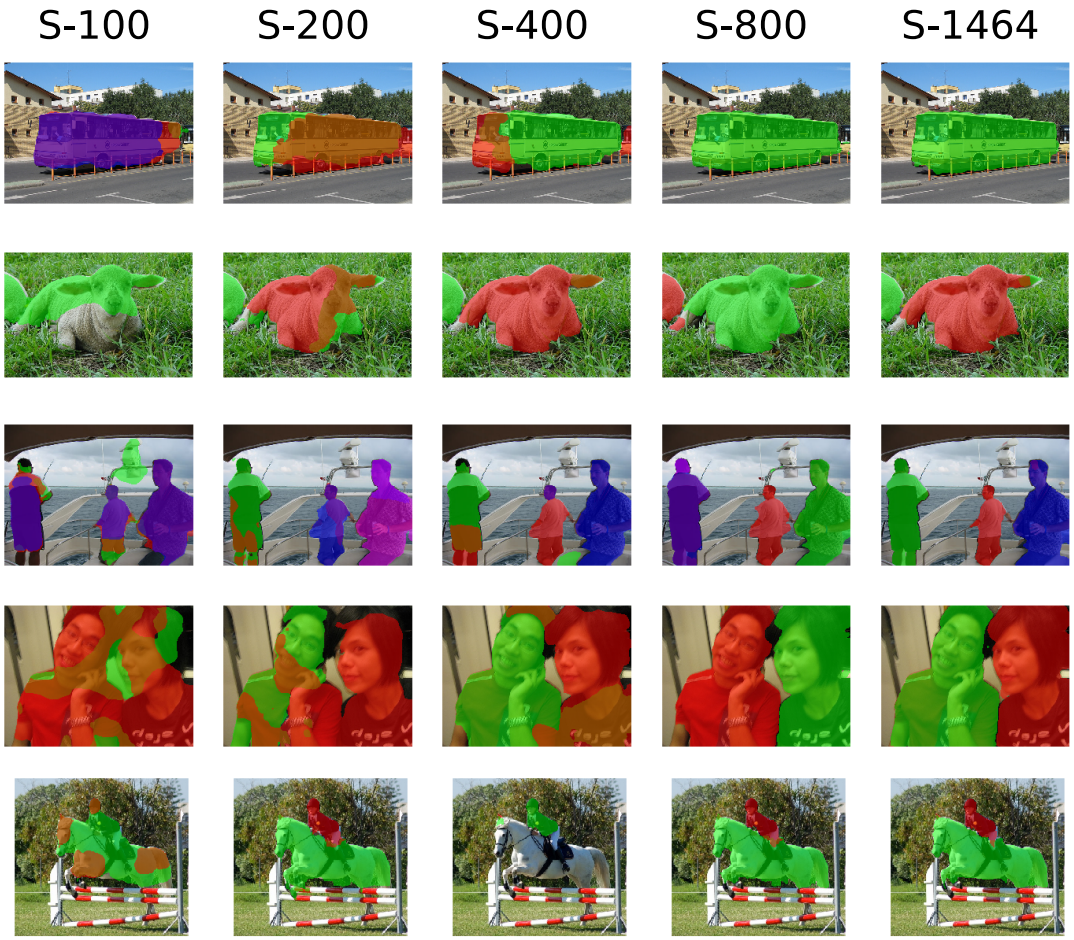

Figure 4.9: Visualization of Pascal VOC validation set for the instance segmentation network g_{φ_N} *(S-)* with $N \in \{100, 200, 400, 800, 1464\}$ and $M = 10582 - N$. The AP (th=0.50) for each configuration is, from left to right, of 7.7, 20.9, 27.9, 42.6 and 48.3.

itself. More details of this architecture are explained in Section 3.3.1 from Part I of this thesis. Our modified RSIS architecture for weak labels (W-RSIS) is also depicted in Figure 4.10. The main difference is that, besides the features extracted by the encoder, the decoder receives at each time step a one-hot encoding of a class category representing each of the instances of the image. If there are several instances belonging to the same class, a one-hot encoding of that class will be given as input at several time-steps.

Table 4.6(a) presents an ablation study to analyze the impact of the different modifications included in W-RSIS. We use the standard semi-supervised setup for Pascal VOC (1464 strong labels and 9118 weak labels). The first row in Table 4.6(a) corresponds to the original RSIS, which annotates samples without using the weak labels. The *+ IL* term means that the output of the *softmax* class predictor is masked at inference time, thus constraining the possible classes predicted for the pseudo-labels. The option *+ C* assumes that the count of instances n in the image is known, and post-processes the pseudo-labels accordingly by keeping the first n objects. Finally, in W-RSIS the IL+C labels are an input of the network f_θ, instead of simply being used as a post-processing step. The ablation study shows how the proposed W-RSIS architecture maximizes the information contained in the IL+C weak labels.

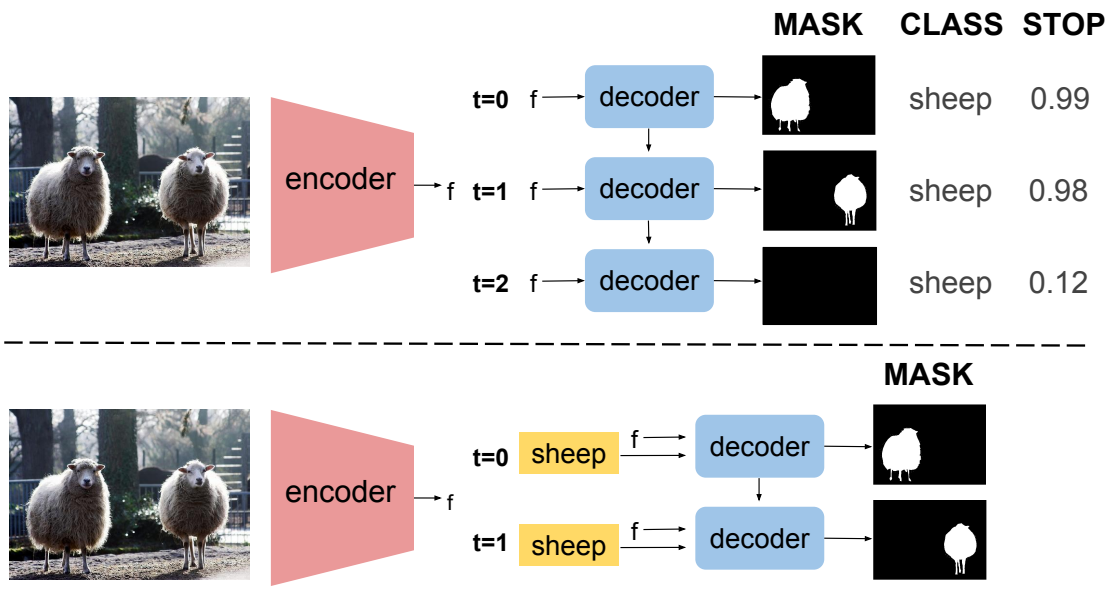

Figure 4.10: RSIS architecture in the first row, and W-RSIS architecture in the second. RSIS has three different outputs, the mask, class, and stop score. When the score is below a fixed threshold (e.g. th = 0.5), no more masks are produced. W-RSIS receives as input a token for each object in the image, so it only has the segmentation output.

RSIS does not impose any order on the sequence of predicted masks. The permutation of the ground truth masks that leads to a lower loss with the predicted sequence is found with the Hungarian algorithm. As in RSIS, we use the soft intersection over union loss (sIoU) as the cost function between the mask predicted by our network and the ground truth mask. Notice that now we have some restrictions in the sequence order, as we want an alignment between the input class category and the output, so in order to train W-RSIS, the Hungarian matching is performed only between ground truth instances of the same category.

Table 4.6(b) includes a second ablation study, in this case, about the masking of the Hungarian algorithm to just allow some permutations, constrained to class categories. The first row corresponds to the basic case *Hungarian*, but we observed that this did not constrain that our input classes were aligned with the classes of the predicted masks. Afterwards, we applied the Hungarian algorithm among objects of the same category only, hence forcing an alignment between the input class categories (which correspond to ground truth) and the actual class category of the prediction. This last *Masked Hungarian* solution resulted to be the best option.

Figure 4.11 proves how W-RSIS generates better annotations compared to RSIS at different annotation budgets. We train multiple W-RSIS models f_{θ_N} with a varying number of strong-labeled samples $N \in \{100, 200, 400, 800, 1464\}$, and compare them to the baseline RSIS. We notice that for any number N of strong samples, W-RSIS outperforms RSIS, being the improvement more notable at low N.

Figure 4.12 shows qualitative results of the pseudo-annotations obtained by both configurations. The first three pairs of images correspond to cases in which RSIS (first row) misses some of the instances because they were predicted with a low confidence score

	AP 50
RSIS	32.4
RSIS + IL	33.6
RSIS + IL + C	37.3
W-RSIS	**38.8**

(a)

	AP 50
Hungarian	34.8
Masked Hungarian	**38.8**

(b)

Table 4.6: (a) Ablation study of IL+C (Image-level Labels plus Counts) as inputs with the Pascal validation set. (b) Ablation study of different losses with the Pascal validation set.

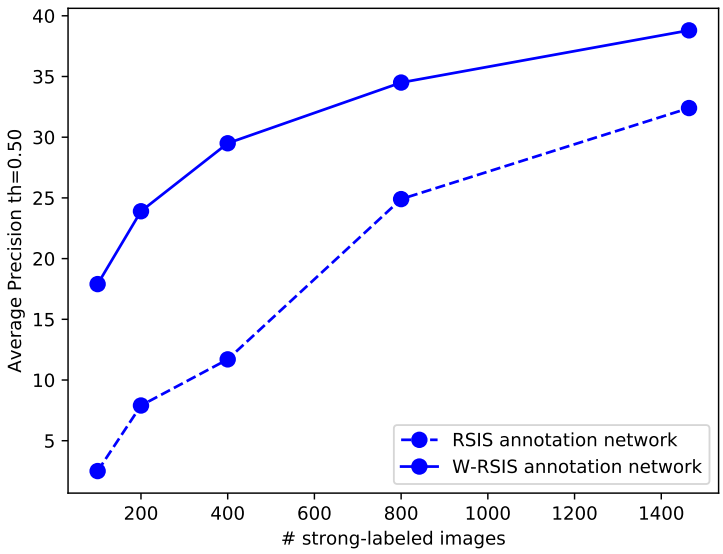

Figure 4.11: Comparison of RSIS annotation network, whose input are only the images to be annotated, and W-RSIS annotation network, whose input are the images and the IL+C information.

that does not reach the minimum detection threshold. W-RSIS (second row) does not present this limitation because the amount of instances for each class is provided by the weak annotation, so the confidence score is ignored. The last pair of images corresponds to the case in which RSIS predicts a wrong class, a problem that W-RSIS does not have either as the category is already provided by the weak label. The additional knowledge about the category of the pseudo-annotation provided by the class label facilitates the task, resulting in better quality masks.

Segmentation network. We analyze the final performance of the segmentation network g_φ in terms of the annotation cost when using the RSIS or W-RSIS annotation network. Notice that the segmentation network g_φ is the same for both configurations (RSIS), just f_θ changes.

Figure 4.12: Comparison of pseudo-annotations obtained by RSIS (first row) and W-RSIS (second row) with $N = 800$. The class category predicted for each pseudo-annotation is written underneath, with the same color code. The performance of each annotation network is 34.5 and 24.9 of AP (th=0.50), for W-RSIS and RSIS respectively.

In Section 4.4 annotating samples was cost-free, as the pseudo-annotations were done on unlabeled images, so varying the number M did not impact the annotation budget. Consequently, we always considered M to be the total size of the training set of Pascal VOC minus N ($M = 10582 - N$). In the heterogeneous setup that we are considering now, there is a cost involved in annotating samples, as weak-labels are fed into the annotation network.

In this section we will vary the number of weak-labeled samples $M \in \{912, 2279, 4459, 6838, 9118\}$, corresponding to the $\{10, 25, 50, 75, 100\}\%$ of the additional Pascal VOC set from [61]. In Table 4.8 we fix three different budget scenarios ($\sim$0.55, $\sim$1.1 and $\sim$2.2 days) and we compare RSIS and W-RSIS. As now the weak-labeled samples have an associated cost, for the W-RSIS setups, we reduce the number of strong samples N, and use part of the budget for M weak-labels. We observe that for very low annotation budgets ($\sim$0.55, $\sim$1.1 days), it is more convenient to spend some budget on weak-labels and reduce the number of strong ones. We even reach better performance at lower annotation budgets (0.79 vs. 1.1 days), as the performance of the W-RSIS annotation network f_θ is significant superior to the baseline one. On the other hand, for higher annotation budgets, the RSIS annotation network f_θ is strong enough, and does not benefit from the weak labels. We observe that in this second case, the baseline reaches better performance (42.6 vs. 40.9 or 36.5 of AP).

The last row in Table 4.8 includes the results of [198], the only previous work addressing the problem of instance segmentation with a low annotation budget. W-RSIS obtains a much better performance at half the budget required by [198] (33.5 of AP at 1.14 days, vs. 26.8 of AP at 2.43 days). Figure 4.13 shows qualitative results for W-RSIS for different numbers of weak-labeled samples.

Table 4.7(a) presents the Average Precision obtained at each of the configurations we tested varying N and M. Note that the first row indicates the results with the W-RSIS annotation network f_θ, while the rest include results for the segmentation network g_φ when trained with a varying number of M samples. We observe the expected behaviour, the higher N and M, the better the results. We can also see how changing N has a higher

	100	**200**	**400**	**800**	**1464**
W-RSIS	19.0	22.7	27.1	34.5	40.0
10%	25.2	27.3	32.5	40.4	46
25%	27.7	30.8	34.8	41.2	47.1
50%	28.0	30.7	32.6	43.1	47.2
75%	28.4	32.7	36.8	43.7	48.4
100%	29.4	33.3	36.8	43.8	48.4

(a)

	100	**200**	**400**	**800**	**1464**
W-RSIS	0.28	0.55	1.11	2.22	4.06
10%	0.51	0.79	1.34	2.45	4.30
25%	0.86	1.14	1.70	2.80	4.65
50%	1.45	1.73	2.28	3.39	5.23
75%	2.03	2.31	2.87	3.98	5.82
100%	2.62	2.90	3.45	4.56	6.40

(b)

Table 4.7: (a) Average Precision (AP) at $th = 0.5$ and (b) Annotation budget (in days) when changing M (rows) and N (columns) for instance segmentation.

	#Strong	**#Unlabeled**	**#Weak**	**Budget**	**AP 50**
RSIS	200	10382		0.55 days	20.8
W-RSIS	100		912	0.51 days	**22.7**
RSIS	400	10182		1.1 days	27.9
W-RSIS	200		912	0.79 days	30.5
W-RSIS	200		2279	1.14 days	**33.5**
RSIS	800	9782		2.22 days	**42.6**
W-RSIS	400		4559	2.28 days	40.9
W-RSIS	200		6838	2.31 days	36.5
[198]			10582	2.43 days	26.8

Table 4.8: Results of the segmentation network when the annotation network changes (RSIS vs. W-RSIS) at different fixed annotation budgets.

impact to the final performance than M, as N indicates the number of strongly-labeled samples. In this configuration, weak labels add a cost to the annotation budget, which we report in Table 4.7(b). In comparison to Table 4.5, we observe better performance at a cost of slight increase in the annotation budget, specially for low values of N.

4.7 Training Details

This Section presents the training details for the different models.

Semantic Segmentation. The model used for all semantic segmentation configurations is the DeepLab v3+ [30] with an Xception65 [36] encoder pretrained on ImageNet [88]. We used the official code in TensorFlow. The different atrous rates that we used were $6, 12, 18$. The output stride chosen is of 16. We used the decoder module explained in the original work [30], with an output stride of 4. We trained the models on 2 Tesla V100 (4 GPU devices in total), with a total mini-batch size of 28. As in the original work [30], we optimized the models with SGD with Momentum of 0.9, with a base learning rate of 0.007. The image resolution used both for training and evaluation is of 513×513 pixels.

Instance Segmentation. For instance segmentation we trained two different models, the RSIS and our modified version W-RSIS. We used the PyTorch code (https://github.com/imatge-upc/rsis). We trained our models with 2 Tesla V100 (4

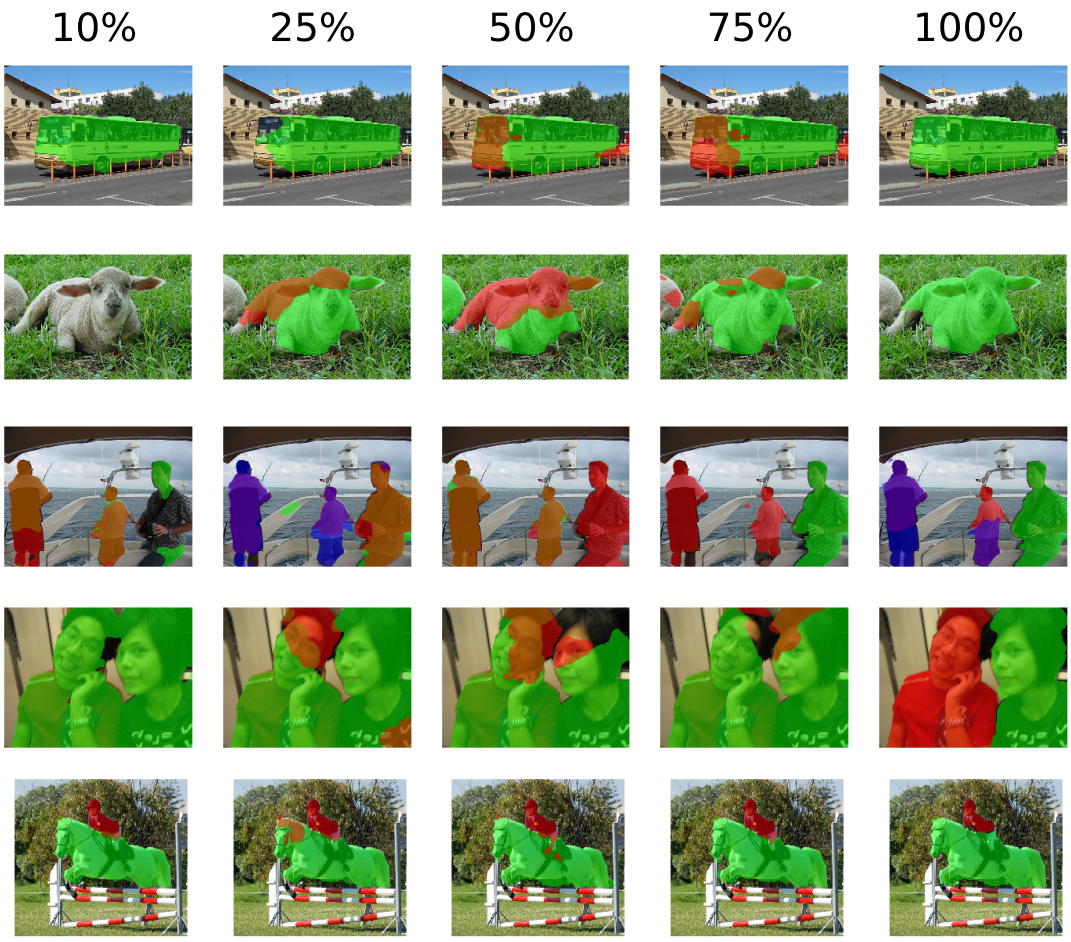

Figure 4.13: Visualization of Pascal VOC validation set for the instance segmentation network g_φ when the annotation network is W-RSIS. The setup consists of $N = 200$ and $M \in \{912, 2279, 4459, 6838, 9118\}$, being N the number of strongly-labeled data, and M the number of weakly-labeled samples. The percentages on top of the figure indicate the fraction of M compared to the total set [61]. The AP (th=0.50) for each configuration is, from left to right, of 30.5, 33.5, 35.4, 36.5 and 43.0.

GPU devices) with a mini-batch size of 60. The encoder is pretrained on ImageNet [88]. The image resolution we used in this case was of 256×256 pixels. As in the original work, we used an Adam [86] optimizer, with learning rate of 10^{-3} for the decoder and of 10^{-6} for the encoder.

4.8 Conclusion

The main contribution of this Chapter is a unified benchmark for image segmentation structured around the annotation cost in days, allowing to compare fairly weakly and semi-supervised methods. This budget-aware benchmark has allowed us to demonstrate that semi-supervised setups are preferable to weakly-supervised setups that rely on only coarse labels. In other words, that fewer but strong labels achieve better results than a larger amount of weak labels. This fewer labels paradigm is especially suitable in those domains in which collecting images is cumbersome (e.g. for the medical field).

Moreover, the time to outline segments can be alleviated even further by modern interactive annotation tools [2, 112]. Therefore, at a restricted annotation cost, more strong labels can be obtained, aiming at closer figures compared to the fully-supervised case.

In next Chapter we exploit active learning to select which samples to strongly-annotate for the BASIS semi-supervised pipeline.

5 Sample Selection for Semi-Supervised Segmentation

5.1 Introduction

This Chapter extends the semi-supervised scheme BASIS presented in Chapter 4. Given a low annotation budget, BASIS outperforms previous works on weakly and semi-supervised semantic and instance segmentation. The setup in BASIS consists of using a limited amount of strong labels, and a larger amount of unlabeled or weakly-labeled data. In this initial approach, the subset of strongly-annotated samples was chosen randomly. In this Chapter, we propose an alternative selection scheme based on active learning, which leads to an improved performance given a fixed annotation budget.

Our active learning method for sample selection consists of firstly training the *annotation* network with a random subset of very few strongly-annotated images. This model is later used to obtain pseudo-annotation masks, as in BASIS, but in this case a confidence score for the masks is predicted, too. This additional information is leveraged to select more images to be strongly-annotated by humans, allowing a more efficient usage of the available annotation budget. Our main contribution is the definition of a novel way to estimate the confidence score. Specifically, our model is trained to predict an estimation of the Intersection Over Union (IoU) of the pseudo-labels and their corresponding ground truth masks. IoU prediction has been used in previous works on object detection for filtering object proposals [80]. To the best of the authors' knowledge, our work is the first one to use IoU as a selection criterion for active learning. We name our selection strategy *Mask-guided sample selection*.

The summary of our contributions in this Chapter is as follows: (a) a novel method to estimate the mask confidence score based on IoU score, being the first work to leverage IoU prediction for active learning, (b) a study of the selected images, which concludes that the best images to annotate are those that are neither the easiest nor the most complicated of our dataset. Finally, (c) with the Mask-guided sample selection strategy we reach higher performance compared to our BASIS baseline, leading to state-of-the-art results at low annotation budgets.

5.2 Related Work

Active learning [158]: consists in recursively selecting which samples to annotate to train a network. The goal of this approach is the reduction of the annotation cost, by only annotating those samples that will have more impact to the learning of the model. This

acquires special relevance in contexts where annotating samples is very expensive, e.g., in image segmentation problems. Common active learning methods select samples according to two main criteria: how *uncertain* and *representative* a sample is. The *uncertainty* is related to how informative a sample is with respect to the learning process.

There are several methods that estimate the *uncertainty*, e.g., dropout has been used to sample from the approximate posterior of a Bayesian CNN to calculate the uncertainty of predictions when varying the model [52]. This quantified metric can be used to request the annotation of subsequent training batches of data [53][59]. More recent methods have also used Bayesian CNNs to calculate the informativeness of images generated by a Generative Adversarial Network (GAN) [109] in order to add these samples to the training set. Another method [46] is based on bootstrapping, and consists in training several networks with different subsets and calculate the variance in predictions across the different networks in order to estimate uncertainty [183].

Some of the aforementioned methods not only base their selection on the *uncertainty* criterion, but also on the *representativeness* of a sample. This criterion is relevant to promote diversity among samples and to avoid redundancy. One strategy used in computer vision is to extract image descriptors with a CNN, and compare images with a cosine similarity metric [183] to avoid picking very similar samples. Maximizing set coverage has also been studied [49]. Other metrics, such as *content distance* have been used to quantify the distance between images to maximize content information [121][122].

Most of the above methods focus on image recognition and region labeling. The first works that handled active learning for large scale object detection [170] used as active learning criterion the *simple margin* selection method for SVMs [166], which seeks points that most reduce the version space. More recently, methods rely on modern object detectors [138][105], but still are based on uncertainty indicators like least confidence or 1-vs-2 [16][150]. Notice that object detection is very close to the instance segmentation task addressed in this work. However, our sample selection criterion is based on the estimated quality of the different masks predicted for each image, instead of using classification scores as the previous approaches. We want to highlight that our method is the first one that proposes active learning for semi-supervised instance segmentation for Pascal VOC benchmark [48], and the first one to explore mask quality prediction as an alternative to classification scores for active learning. Our claim is that classification scores are suitable for object detection pipelines, but do not reflect the quality of the actual pixel-wise annotation used to train instance segmentation models.

IoU prediction: IoU prediction has been used in previous works for filtering object proposals in object detection tasks [80]. In particular, in [80] the IoU between predicted bounding boxes and ground truth bounding boxes is estimated, and the authors argue that this score, in comparison to a class confidence score, considers the localization accuracy. In their work they show how their approach leads to improved performance. Similarly to this work, [75] estimate the IoU between the predicted masks and the ground truth masks, and use this score to better filter object proposals for instance segmentation. In this direction, we propose to also predict the Intersection Over Union of the predicted masks with respect to the ground truth as a measure of the confidence of the prediction.

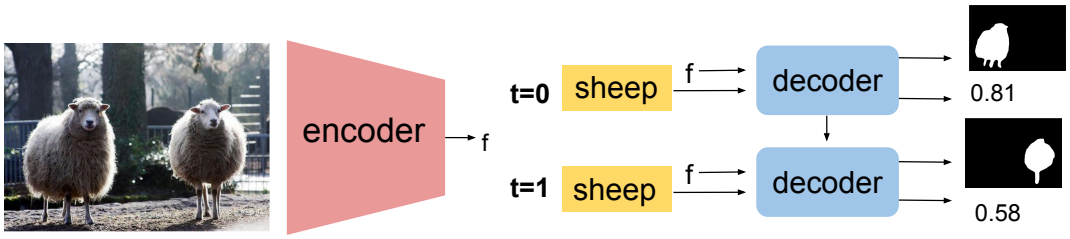

Figure 5.1: *IoU-W-RSIS* model with the IoU branch. The model consists of an encoder for the image, that is a ResNet-101, and a recurrent decoder that receives at each time step a class category label, in this example, it receives at the first time step the label *sheep*, and in the second time step it receives the same label, as in the image there are two instances of the *sheep* category. The decoder also receives the features obtained by the encoder, and at each time step produces a binary mask with the segmented instance, and a prediction of the IoU of the produced mask.

5.3 IoU Quality Prediction

We study a selection criterion mechanism for the semi-supervised setup with heterogeneous annotations presented in Section 4.6.1. This setting combines two types of annotations: strongly-annotated samples (with pixel-level annotations) and weakly-annotated samples with image-level plus counts (IL+C). This type of weak annotation consists in indicating the class labels in each image, and the counts of how many times each category appears. In this Chapter we decided to work with this setup, instead of using unlabeled samples, because with the IL+C weak labels we know beforehand how many objects there are in each image, which facilitates the study of predicting a confidence value for each instance. However, the strategy that we propose could be easily adapted for the setup in which no weak labels are provided.

Compared to what we presented in the previous Chapter 4, the only change in the model architecture is an additional output to the *W-RSIS annotation* network, already presented in Section 4.6.1. This output estimates the quality of each predicted mask, which can be used to guide an active learning algorithm in choosing which images should be strongly-annotated given a limited budget. Our proposed method to estimate the mask quality is to predict the Intersection over Union (IoU) of the predicted mask over a hypothetical ground truth. The IoU measures the intersection between two regions divided by its union, and it is a common metric to assess segmentation performance (Equation 5.1). For binary segmentation, the IoU is computed on the foreground pixels of the predicted and the ground truth masks.

$$IoU(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{5.1}$$

We call this new architecture *IoU-W-RSIS*, and it is depicted in Figure 5.1. The model can be trained in a fully-supervised setup, as the ground truth masks are available for the training data. *IoU-W-RSIS* will segment an object mask of the category fed in the input and predict a confidence score of the segmentation quality at each time step.

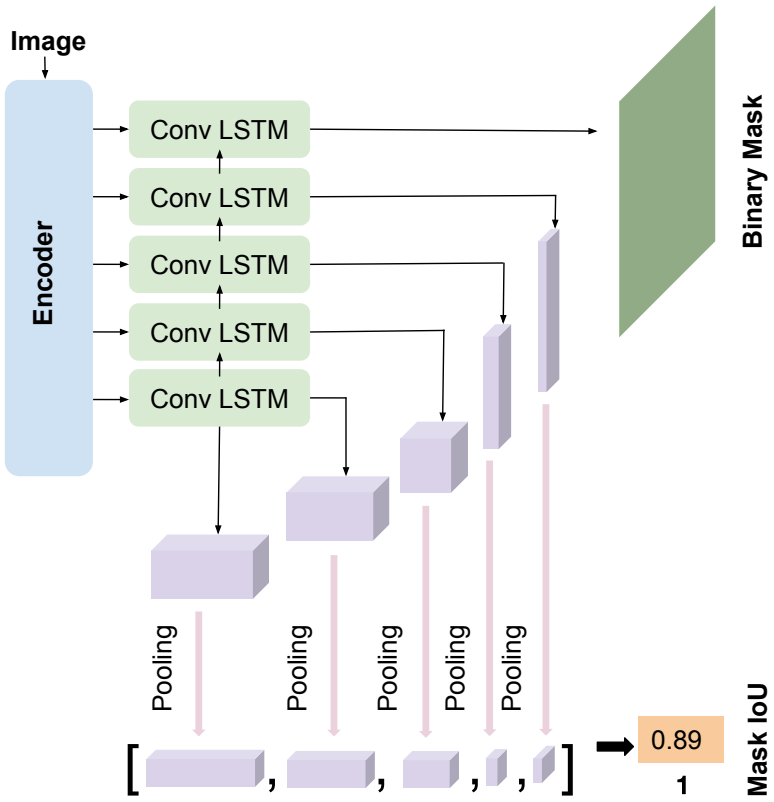

Figure 5.2: *IoU-W-RSIS* model with the IoU branch for a single time step. The class label is omitted in this figure for clarity. The input image is fed into the encoder and the features at different resolutions are fed at different levels of the decoder. Each level of the decoder has a convolutional LSTM (Conv LSTM) layer that receives a hidden state from the previous time step, and produces features for the current time step. The features of the different levels are averaged-pooled and concatenated, and are the input of a fully connected layer that predicts the mask IoU. On the other hand, the features of higher resolution are the ones that produce the binary mask that corresponds to the segment.

The architecture that predicts the IoU is depicted in Figure 5.2. A branch for IoU prediction is added to the decoder of the network, indicated in the Figure as *Mask IoU*. This branch aggregates features of the decoder at different spatial resolutions, concatenates them, and computes global average pooling. Afterwards, we add a fully connected layer that predicts the IoU using an L1 regression loss. This loss term is introduced once the segmentation loss has already converged. At that point, the network weights are frozen and only the additional IoU branch is trained for a few epochs. To give more relevance to the predictions of low IoUs, we predict the squared IoU, as suggested in other scenarios in which small values have important relevance, as bounding box offset regression for object detection [138].

With the proposed architecture, an *IoU Score* for each mask is predicted. In our methodology we use an overall IoU per image instead of individual IoU scores per object. This means that a human annotator will be asked to annotate all object instances from the selected images. Therefore, to compute the *IoU Score* for an image with M objects, we

simply average the scores predicted per each object, as seen in Equation 5.2.

$$IoU\ Score\ =\ \frac{1}{M}\sum_{i\epsilon M} IoU_i \tag{5.2}$$

5.4 Experiments

The *IoU-W-RSIS annotation* network presented in Section 5.3 is tested considering one active learning iteration for the task of instance segmentation. Our experiments aimed at measuring the gain of a IoU-guided selection of the images to strongly-annotate, compared with a baseline of random selection as in Section 4.6.1, and with baseline techniques for active learning based on Dropout [52]. We present experiments for the instance segmentation task for the Pascal VOC 2012 benchmark [48]. The standard semi-supervised setup adopted for this benchmark consists in using the Pascal VOC 2012 train images (1464 images) as strong-labeled images, and an additional set (9118 images) from [61] as unlabeled/weak-labeled. For our study, as done in the previous Chapter 4, we select which samples to strongly-annotate from the Pascal VOC 2012 train images. The additional set of Pascal is used to obtain pseudo-annotations for the semi-supervised pipeline.

Two sets of experiments are presented, first we focus on the IoU prediction task (Section 5.4.1), and then we study how to use this score for tackling sample selection (Section 5.4.2).

5.4.1 IoU Prediction

As a first experiment, we try several configurations to train the IoU branch of the *IoU-W-RSIS* architecture. We train our proposed annotation network *IoU-W-RSIS* with $N \in \{100, 200, 400, 800, 1464\}$, where N is the amount of strongly-annotated samples. These N samples are randomly selected from the Pascal VOC 2012 train set (that has a total of 1464 images). Table 5.1 contains the Mean Absolute Error (MAE) computed as the mean of the MAE of *IoU Scores* (Eq. 5.2) of the dataset for the different configurations. The *Baseline* configuration consists in training the IoU branch at the same time as the segmentation branch. In the next row, we freeze the weights of the segmentation network after 150 epochs and only train the IoU branch (for 250 epochs). Finally, we optimize the squared root of the IoU and this option leads to the best results. As expected, the MAE tends to decrease from left to right in the table, which corresponds to considering more strongly annotated images.

To better understand the kind of segmentation that our model obtains, Figure 5.3 shows results from the network trained with the minimum number of images, only 100 random samples. Note that this model gets a performance of 19.0 Average Precision (threshold=0.5). When there is a single object in the scene, the model gets decent performance, but as there are more objects, the masks obtained are worse. However, the goal of this network is not to get accurate segments, but to estimate which is the performance of this model for a given image. With this estimation, we can identify for which images the current model is struggling to obtain good performance, and for which images the current model is good enough already. This information is crucial to define our mask-guided selection criterion.

	100	**200**	**400**	**800**	**1464**
Baseline	31.1	39.8	49.3	47.7	51.0
+ Freeze Seg. Network	24.8	**16.7**	19.0	17.1	**16.6**
+ Sqrt Loss	**23.6**	19.5	**18.0**	**17.0**	**16.6**

Table 5.1: Mean Absolute Error (MAE) of IoU prediction. Each column indicates the number of samples used to train the IoU prediction branch, and each row is a different configuration that we test. The one that yields best performance is when we freeze the segmentation network and when the prediction of the model is the squared root of the IoU.

Figure 5.3: Segmentations predicted by the *IoU-W-RSIS* model trained on only 100 images. Each color indicates a different mask.

5.4.2 Mask-guided sample selection

The second set of experiments exploit the estimated IoU to select which images should be strongly-annotated and used as supervision to train the *annotation* network in the BASIS pipeline.

Considering a fixed set of 1464 images from Pascal VOC 2012, our proposal firstly trains an *IoU-W-RSIS annotation* network with a few randomly selected samples (100), and later the remaining samples (1364) are processed through the trained model. Together with the predicted masks for these 1364 samples, the *IoU-W-RSIS annotation* network predicts also a IoU score for each input sample. We explore different approaches to select which subset of images should be strongly-annotated based on the estimated IoU for each predicted mask. Next, the chosen samples are manually annotated with pixel-level labels and added to the training of the *annotation* network. This pipeline is depicted in Figure 5.4. We follow the classic active learning setup, in which the samples to be annotated are iteratively selected. In our case, we experiment with a single iteration, but it could be easily extended to a looped pipeline.

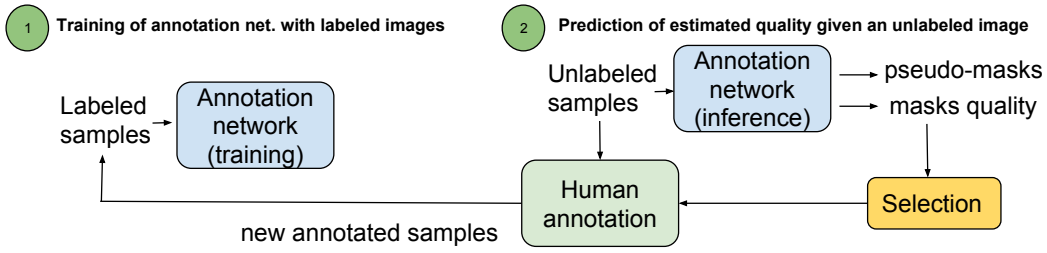

Figure 5.4: Active Learning pipeline to select next samples to be labeled by a human annotator. The first step consists in training the annotation network with few strongly-labeled samples. The second step consists in using the annotation network to obtain pseudo-masks for the unlabeled samples, together with the masks quality score. Based on this score, some samples are selected to be manually annotated by a human, and added to the pool of labeled samples to re-train the annotation network.

	200	**400**	**800**
Random subset	22.7 ± 1.8	27.1 ± 0.8	34.5 ± 2.0
Dropout Baseline (highest)	21.4 ± 1.4	23.9 ± 1.3	28.1 ± 1.9
Dropout Baseline (lowest)	20.0 ± 0.9	24.8 ± 1.5	32.2 ± 1.4
$\beta = 0.0$	20.9 ± 1.5	24.1 ± 0.7	29.1 ± 1.3
$\beta = 0.1$	22.3 ± 1.5	23.8 ± 0.6	28.6 ± 0.7
$\beta = 0.2$	23.3 ± 0.8	24.4 ± 0.3	31.6 ± 1.1
$\beta = 0.3$	**23.9 ± 0.8**	26.5 ± 2.6	32.9 ± 1.4
$\beta = 0.4$	23.4 ± 2.7	**29.0 ± 1.3**	35.0 ± 0.6
$\beta = 0.5$	22.2 ± 1.1	28.9 ± 0.7	35.1 ± 0.9
$\beta = 0.6$	22.2 ± 2.4	28.6 ± 1.3	**35.4 ± 2.4**
$\beta = 0.7$	22.3 ± 1.2	26.7 ± 1.3	**35.4 ± 1.4**
$\beta = 0.8$	21.9 ± 2.0	25.3 ± 1.2	33.4 ± 3.1
$\beta = 0.9$	20.4 ± 1.1	25.9 ± 1.1	34.8 ± 1.9
$\beta = 1.0$	20.3 ± 1.1	25.2 ± 2.3	34.5 ± 1.3

Table 5.2: Oracle: mean Average Precision (th=0.5) for different selection criteria (5 runs for each configuration). Each column indicates the number of images used to train the segmentation models. The first row shows the results obtained with random selection of samples, the second and third rows show a baseline sample selection method, whereas the following rows show different selection criteria with our method. If $\beta = 0.0$ and the number of training samples is 200, means that the first 100 samples are randomly-selected and the next 100 are the ones that have a IoU closer to 0.0.

5.4.2.1 Criterion for sample selection based on IoU

In this Section we explore a criterion for selecting which images should be strongly-annotated by a human given their estimated *IoU Scores*. As we would like our analysis to focus on the selection criterion only, in this Section we will not use the IoU value predicted by our model but the real ground truth value (*oracle*).

Our experiments start with an *IoU-W-RSIS annotation* network trained with only 100

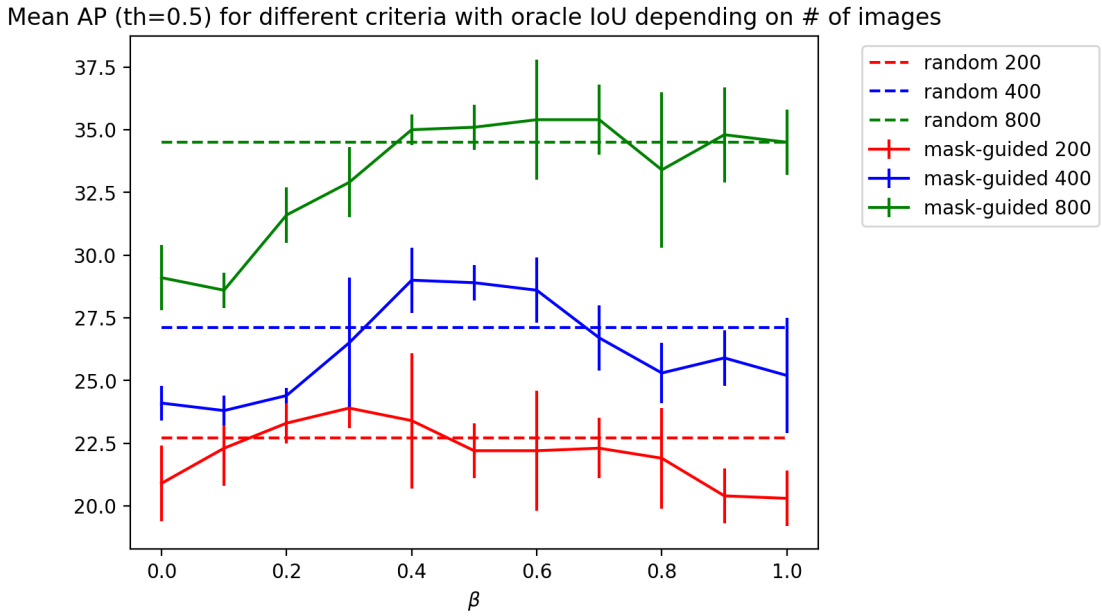

Figure 5.5: Oracle: mean Average Precision (th=0.5) for different selection criteria (5 runs for each configuration). We illustrate two criteria: the mask-guided and the random one. We see how for some *IoU scores* our proposed method surpasses the random one. For the mask-guided criteria, the figure shows the variance of each configuration. We compare three different scenarios depending on the number of training images, as in Table 5.2.

samples, which obtains a performance of 19.0 Average Precision (threshold=0.5). After that, we select another N' samples to be manually annotated, being $N' \in \{100, 300, 700\}$ to make a total of $N \in \{200, 400, 800\}$ strongly-annotated samples. The criterion used to select these N' samples consists in first defining a set of *IoU Scores* (from 0 to 1.0 in steps of 0.1), that we name β, and select the N' images (being $N' \in \{100, 300, 700\}$) whose *IoU Scores* are closest to these β values. Finally, the samples used to train the *annotation* networks are the 100 initial random images plus these N' selected images. The performance obtained with these different subsets is presented in Table 5.2, which reports the Average Precision (threshold=0.5). All configurations have been trained five times, and the reported results are the average with the standard deviation of the performance of these different models. Notice that we compare our approach to a random selection and to two baseline selection criteria. These baselines consist in adding a dropout layer at the end of the encoder of our model, with 50% of probability of dropping out the neurons. Following, we test each of the trained models five different times with the dropout, and obtain the predicted masks for each run. We compute the standard deviation of the pixels from the masks predicted, to see which samples vary significantly between different runs when different neurons are dropped, as a way to estimate the uncertainty of the predictions. Finally, we select the images related to the lowest standard deviation values (Dropout Baseline lowest) or the highest values (Dropout Baseline highest), similarly to previous works [53][59]. We illustrate the same results in Figure 5.5, comparing our mask-guided criteria, and the random one.

The results in Table 5.2 and Figure 5.5 show that there are multiple subsets that out-

perform the random and the baseline selections. This means that our selection strategy based on IoU is effective to reach better performance. We also notice that the optimal predefined *IoU Score* is not fully consistent across different subsets sizes (at $N = 800$ the optimal score is 0.6, whereas at $N = 200$ the optimal score is 0.3). Interestingly, these optimal *IoU Score* values suggest that the best options are the ones that select images that are neither the most challenging of the dataset nor the easiest ones. We also observe that the dropout baselines do not surpass our method and are even worse than random selection. As our results with *IoU Score* indicate, the best options for the dropout baselines may be related to neither the highest nor lowest standard deviations. However, choosing a mid-range option for standard deviation is not as intuitive as it is with *IoU Score*. In the latter case, we only need to select an IoU which directly reflects the quality of the predicted masks. Moreover, our method does not need to run the model several times to select the samples, as it happens with the dropout baselines, thus it is computationally more efficient.

5.4.2.2 Predicted IoU-selection

The experiments in Section 5.4.2.1 with the real ground truth IoU (the *oracle* experiment) showed that choosing samples based on the IoU quality metric leads to better results than performing a random selection or a baseline active learning selection.

In this Section, we address the realistic case in which the IoU is predicted by the same annotation network, instead of using the ground truth value as in Section 5.4.2.1. Table 5.3 and Figure 5.6 shows that for the three set sizes ($N = 200, 400, 800$) better results are also obtained by selecting with the IoU criterion instead of performing a random selection or using the dropout baseline defined previously. The optimal *IoU scores* are between 0.3 and 0.6. In fact, we observe a tendency that for smaller subsets, a lower *IoU score* is optimal, whereas for larger subsets, a higher *IoU score* works better. We also observe there is no significant difference in the maximum performance between the results obtained with the *oracle* and the predicted IoU configurations, but the curves (Figure 5.6) from the latter are noisier.

5.4.2.3 Sets analysis

In this Section we will analyze the properties of the *N'* samples selected based on the selection criterion when considering different *IoU Scores* predefined values. We compare the subsets obtained from the *oracle* and the predicted IoU configurations. In Figure 5.7 we depict an histogram of the average number of objects per image and the mean size of objects per image for each of the subsets, depending on the predefined *IoU Scores*. The plot has two different columns, the first one belongs to the *oracle* configuration and the second one to the predicted IoU configuration. For both the *oracle* and the predicted IoU configurations, we observe that lower *IoU scores* are related to images with more objects per image and smaller objects. These two scenarios correspond to very challenging cases in object detection, as pointed out by previous works [56]. Finally, we can observe that the subsets created by the predicted IoU follow a similar distribution to the *oracle* one.

As we already found in Section 5.4.2.2, the optimal *IoU Scores* are between 0.3 and 0.6. In Figure 5.7 we can see how images associated to these values tend to have a close to the average number of objects per image (2.8 objects/image). Regarding object size, we

	200	**400**	**800**
Random subset	22.7 ± 1.8	27.1 ± 0.8	34.5 ± 2.0
Dropout Baseline (highest)	21.4 ± 1.4	23.9 ± 1.3	28.1 ± 1.9
Dropout Baseline (lowest)	20.0 ± 0.9	24.8 ± 1.5	32.2 ± 1.4
$\beta = 0.0$	21.5 ± 1.1	23.7 ± 0.6	30.1 ± 1.7
$\beta = 0.1$	21.8 ± 1.6	23.7 ± 0.7	30.3 ± 1.7
$\beta = 0.2$	22.6 ± 0.9	25.0 ± 0.8	29,9 ± 2.2
$\beta = 0.3$	**24.0 ± 1.3**	26.9 ± 3.2	33,5 ± 3.1
$\beta = 0.4$	23.2 ± 0.4	24.8 ± 2.2	35.3 ± 0.9
$\beta = 0.5$	20.9 ± 3.1	25.0 ± 0.9	**37.0 ± 2.0**
$\beta = 0.6$	20.6 ± 1.2	**27.5 ± 2.7**	34.8 ± 3.0
$\beta = 0.7$	20.3 ± 1.0	26.2 ± 3.1	36.3 ± 1.1
$\beta = 0.8$	20.7 ± 2.1	26.9 ± 1.6	35,9 ± 2.5
$\beta = 0.9$	20.8 ± 0.8	26.1 ± 1.2	35,5 ± 1.1
$\beta = 1.0$	21.1 ± 1.5	24.8 ± 1.5	34.6 ± 2.1

Table 5.3: Predicted IoU: mean Average Precision (th=0.5) for different selection criteria (5 runs for each configuration). Each column indicates the number of images used to train the segmentation models. The first row shows the results obtained with random selection of samples, the second and third rows show a baseline sample selection method, whereas the following rows show different selection criteria with our method. If $\beta = 0.0$ and the number of training samples is 200, means that the first 100 samples are randomly-selected and the next 100 are the ones that have a predicted IoU closer to 0.0.

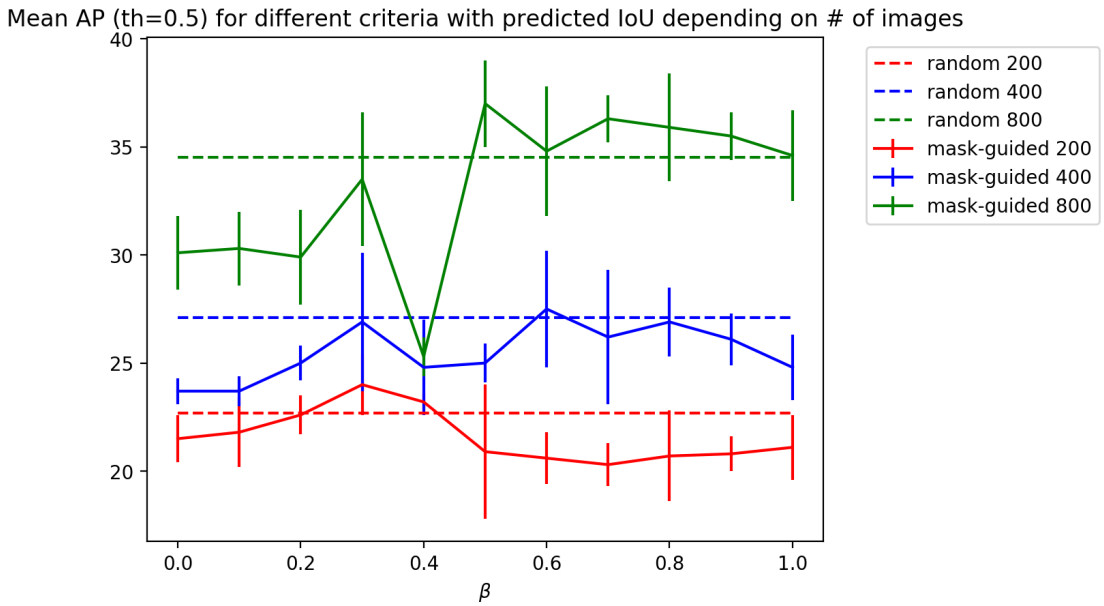

Figure 5.6: Predicted IoU: mean Average Precision (th=0.5) for different selection criteria (5 runs for each configuration). We illustrate two criteria: the mask-guided and the random one. We see how for some *IoU scores* our proposed method surpasses the random one. For the mask-guided criteria, the figure shows the variance of each configuration. We compare three different scenarios depending on the number of training images, as in Table 5.3.

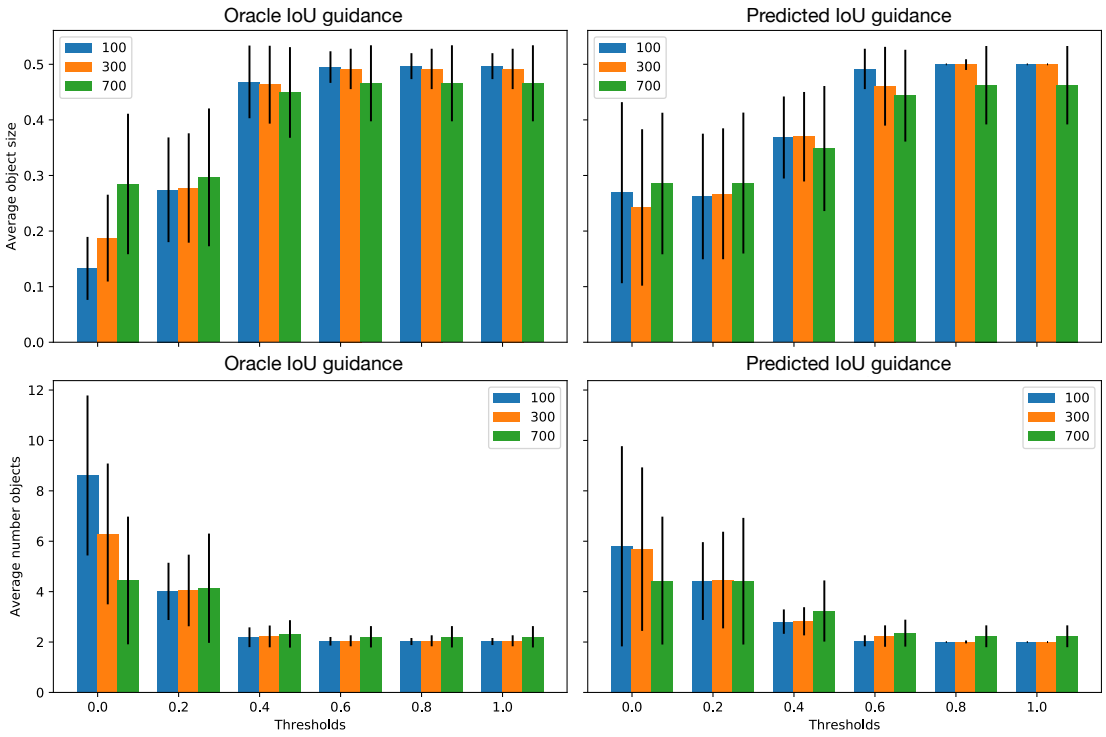

Figure 5.7: Analysis of the mean object size (first row) and number of objects (second row) of the selected images when considering the oracle and the predicted IoU. We consider three different scenarios, when 100, 300 or 700 samples are added to the initial 100 randomly-selected set of samples.

observe that objects tend to be neither the largest ones nor the smallest.

Figure 5.8 shows some of the selected images when different *IoU Scores* are considered. We observe that at high *IoU Scores* values (0.8 or 1.0), images selected are easy, with only one or two large objects in the image. On the other hand, at low *IoU Scores* (0.0 or 0.2) images have multiple, rather small, instances. As our results indicate, the optimal selected samples to be strongly annotated are those in the middle of the range. These are images that have multiple instances but that are not too complicated to segment. We hypothesize that training with very difficult images can be inefficient if the model is not capable to learn from them, while easy cases do not add much value to the learning process.

5.4.2.4 Training of segmentation network

In this Section we focus on the final goal of the pipeline: training the *segmentation* network. As a first step, an *annotation* network of $N = 100$ is trained with 100 random samples, and following 100 more samples are selected (the ones that are closest to the *IoU score* of 0.3, which is the optimal for this set size), so we get an *annotation* network trained with $N = 200$. The same procedure applies for $N = 400$ and $N = 800$, always starting from $N = 100$ random samples, and applying thresholds 0.6 and 0.5 respectively, as they are the optimal ones for each set size. Once the *annotation* network has been

Figure 5.8: Examples of images of each subset. Each column are images related to different predicted IoUs. For instance, the first column belongs to the images that have a mean IoU closer to 1, and we can see that these images are simple, with a single and big instance appearing.

trained with the best selection of samples given our mask-guided criterion, we use this *annotation* network to pseudo-annotate the additional Pascal set from [61], a total of 9118 images. Lastly, we train the *segmentation* network with the obtained pseudo-annotations and the available strongly-labeled samples.

Figure 5.9 shows the Average Precision (threshold 0.5) achieved by the *annotation* network when the samples are selected with the mask-guided criterion, compared to a random selection, depending on the final number of strongly-labeled samples used. We see how the proposed selection strategy outperforms the random one for all data points.

Following, Figure 5.10 depicts the Average Precision (threshold 0.5) of the *segmentation* network depending on the total annotation budget in days. We test two configurations, pseudo-annotating all the additional set of Pascal [61], consisting of 9118 images, or only annotating half of it, 4559 samples. The three data points plotted per curve consist in using $N = 200$, $N = 400$ or $N = 800$ strongly-labeled samples. We see that all configurations of the mask-guided selection require a slight higher budget compared to the random selection configurations. That is, because in order to choose which samples to strongly-annotate, we need extra IL+C weak labels for the Pascal 2012 train set as inputs to the network, whose budget is added to the total cost. As we can observe in the plot, this additional annotation cost is worth it for the $N = 400$ and $N = 800$ strongly-labeled samples configurations, as the performance is higher compared to the random one.

The final results obtained by the *segmentation* network are presented in Tables 5.4

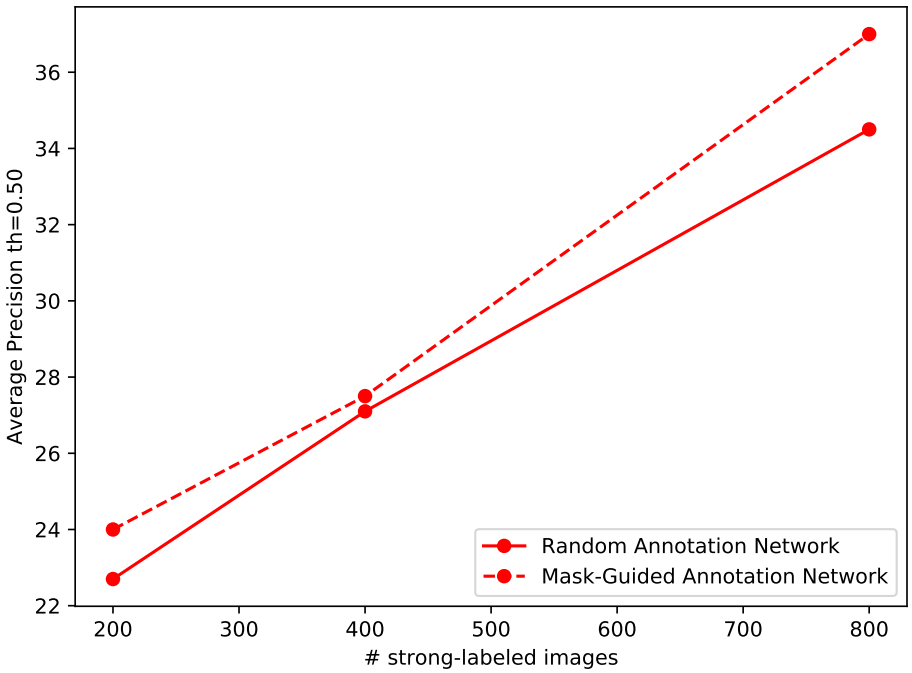

Figure 5.9: Annotation networks comparison, in terms of Average Precision (AP) th=0.5, for the validation set of Pascal VOC, when samples are chosen based on random selection and with the mask-guided selection strategy.

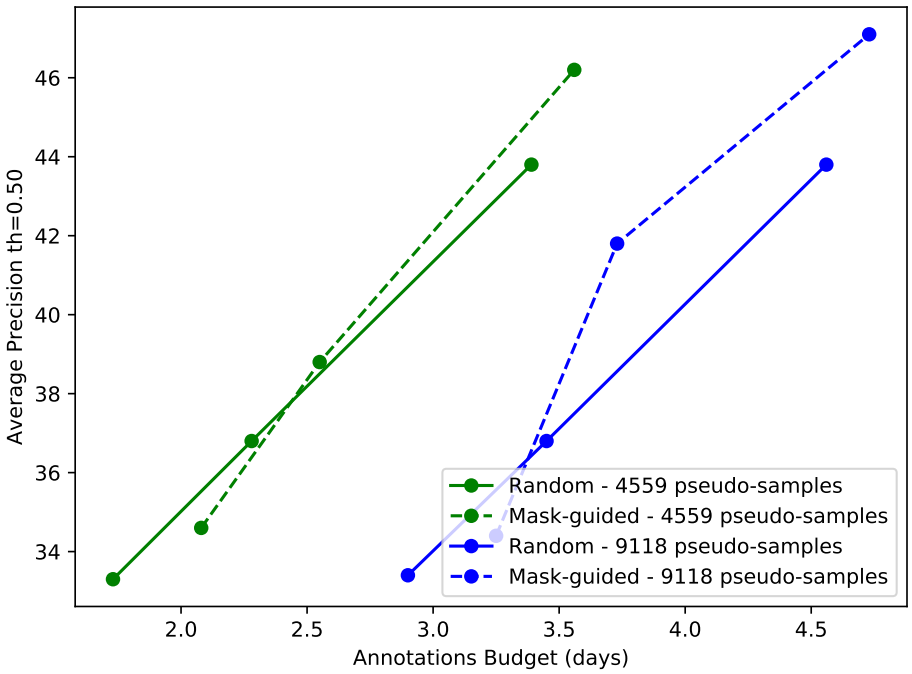

Figure 5.10: Segmentation networks comparison, in terms of Average Precision (AP) th=0.5, for the validation set of Pascal VOC, when samples are chosen based on random selection and with the mask-guided selection strategy.

	AP @[0.5:0.95]	AP @[0.5]	AP @[0.7]	AR @[0.5:0.95]	AR @[0.5]	AR @[0.7]	F@[0.5]	SSIM
200	18.7	34.4	20.6	26.1	41.6	28.7	37.7	84.0
400	24.8	41.8	28.2	33.6	50.2	37.5	45.6	83.6
800	29.2	47.1	32.7	38.7	55.4	42.9	50.9	85.8

Table 5.4: Average Precision and Average Recall at different thresholds, F measure and SSIM for the *segmentation* networks with the Mask-guided criterion. The samples pseudo-labeled are the complete additional set of Pascal (9118 images).

	AP @[0.5:0.95]	AP @[0.5]	AP @[0.7]	AR @[0.5:0.95]	AR @[0.5]	AR @[0.7]	F@[0.5]	SSIM
200	19.5	34.6	21.7	27.3	42.8	30.1	38.3	84.4
400	22.8	38.8	26.4	32.5	48.2	37.0	43.0	83.7
800	29.0	46.2	33.0	37.9	54.2	42.4	49.9	85.9

Table 5.5: Average Precision and Average Recall at different thresholds, F measure and SSIM for the *segmentation* networks with the Mask-guided criterion. The samples pseudo-labeled are 50% of the additional set of Pascal (4559 images).

and 5.5 with three complementary metrics: Average Precision and Recall at different thresholds, the F measure at threshold 0.5, which corresponds to $F = 2 * (precision * recall)/(precision + recall)$, and the Structural Similarity Index (SSI), as an effort to have a metric that considers the overall structure of the mask instead of pixel-wise errors. Observing these metrics, we see that for $N = 400$ and $N = 800$, using more pseudo-labels (9118 vs. 4559) leads to better performance, while for $N = 200$ it is not the case. This may be produced by the different ratio of pseudo-labeled samples vs. strongly-labeled samples, which is significantly larger for $N = 200$. Having a large ratio of noisy labels compared to reliable ones, could damage the training of the model. We also observe that when varying the number of pseudo-labels (9118 vs. 4559), the SSIM does not change as much as we see for the other metrics. Interestingly, the SSIM for $N = 200$ is higher than for $N = 400$. The reason could be that with $N = 200$ the blobs obtained in the masks are coarser and this could favour this metric because it is based on structural similarity. Nevertheless, the difference is not very significant between both configurations.

5.5 Conclusions

In this Chapter we have proposed a novel method to select which samples to strongly-annotate in our semi-supervised instance segmentation setup. Our method, based on IoU prediction, outperforms the baseline random selection and a solution based on neural dropout to estimate pixel-wise uncertainty. We guided a detailed analysis of which samples are best to annotate given the confidence score of the predictions, and we observe that the best samples are those that fall in the mid-range of the IoU scores. With our pipeline, we present a very simple but effective manner to perform sample selection to improve performance at a negligible annotation cost. As future work we think that IoU prediction for sample selection can be exploited in other tasks, such as semantic segmentation or object detection. Regarding instance segmentation, which is a very challenging task in the field of scene understanding, we think that our experimental validation proves that the task can be addressed with low annotation budgets, and that exploiting few but interesting samples can lead to better results.

Summary of Part II

Part II has presented two different techniques for weakly-supervised segmentation, both of them leveraging the same semi-supervised framework. Chapter 4 has presented BASIS, a semi-supervised pipeline for semantic and instance segmentation, and Chapter 5 has introduced a sample selection mechanism for the instance segmentation case. The main contribution of this Part is that we analyze how, at matching annotation costs, semi-supervised pipelines surpass methods that rely on weak labels only. Hence, having a few strong labels is key for good performance, and we studied how selecting wisely which samples to strongly-annotate can increase further the accuracy of our semi-supervised models.

The aforementioned strategies focus on lowering the annotation cost during training. In Part III, we concentrate on the inference part of the system, concretely when addressing video object segmentation. In this task, a pixel-level mask of the object to be segmented is typically required at inference, as we saw in Part I with one-shot video object segmentation. In Part III, instead of relying on pixel-level masks, we explore natural language expressions as cues to indicate which object to segment throughout a video sequence.

Part III

Language-guided Video Object Segmentation

6 Referring Expressions for Video Object Segmentation

6.1 Introduction

Part II of this thesis addressed how to train segmentation models with low annotation budgets. This budget only considered the time to annotate the training data. Another manner to leverage human effort, is to center on the human interaction required at inference time.

One-shot video object segmentation [131, 181], a task already addressed in Part I of this thesis, requires significant human interaction at inference, as a pixel-level mask for each object of interest in the video has to be provided for the first frame. The goal of the system is to follow the objects along the video sequence. In Part I we also addressed zero-shot video object segmentation, meaning that no object mask is provided and the system must discover different objects in the video clip without any initialization cue. Nonetheless, we analyzed that the performance dropped significantly compared to the one-shot case. In Part III of this thesis we study a trade-off that consists in lowering the effort required at inference time compared to the one-shot case, but still providing some initialization cue to the system to identify the target object.

We aim at improving the human computer interaction by allowing linguistic expressions as initialization cues, instead of interactive segmentations under the form of a detailed binary mask, bounding box, scribble or point. In particular, we focus on referring expressions (REs) that allow the identification of an individual object in a discourse or scene (the *referent*). For instance, Figure 6.1 depicts REs related to one of the objects contained in a video sequence, which is highlighted in green.

Language-guided Video Object Segmentation (LVOS) was first addressed by Khoreva et al.[85], and tackled later by Gavrilyuk et al. and Wang et al. [55, 173]. Compared to related works on still images [188, 25], REs for video objects may be more complex, as they can refer to variations in the properties of the objects, such as a change of location or appearance. The particularities of REs for videos were initially addressed by Khoreva et al. [85], who built a dataset of REs divided in two categories: REs for the first frame of a video, and REs for the full clip. We propose another approach for analyzing the performance of the state of the art in VOS with REs. We identify seven categories of REs and use them to annotate existing datasets.

We address both the language-guided image segmentation and the language-guided video object segmentation tasks with *RefVOS*, our end-to-end deep neural network that lever-

ages BERT language representations [44] to encode the phrases. RefVOS achieves results comparable to previous works for the RefCOCO dataset of still images [82], and improves state-of-the-art works over the DAVIS-2017 [135] and Actor-Action datasets (A2D) [179] for video with the phrases collected by Khoreva et al. [85] and Gavriluk et al. [55], respectively. We also identify the categories of REs which are most challenging for RefVOS.

The main contributions presented are summarized as follows: (a) an end-to-end model, *RefVOS*, that achieves state of the art performance with available phrases for DAVIS-2017 and A2D benchmarks, (b) a novel categorization of REs tailored to the video scenario with an analysis of the current benchmarks, and (3) an extension of A2D with additional REs with varying semantic information to analyze the limitations and strengths of our model according to the proposed linguistic categories.

6.2 Related Work

Language-guided Image Segmentation: The task, also known as referring image segmentation, was first tackled by Hu et al. [72]. They use VGG-16 [159] to obtain a visual representation of the image, and a Long-Short Term Memory (LSTM) network to obtain an embedding of the RE. From the concatenation of visual and language features, the segmentation of the referred object is obtained. Posterior work [102] explored how to include multi-scale semantics in the pipeline, by proposing a Recurrent Refinement Network that takes pyramidal features and refines the segmentation masks progressively. Liu et al. [103] argued to better represent the multi-modality of the task by jointly modeling the language and the image with a multi-modal LSTM that encodes the sequential interactions between words, visual features and the spatial information. With the same purpose of better capturing the multi-modal nature of this task, long-range correlations between the visual and language representations can be reinforced by learning a cross-modal attention module (CMSA) [186] or by learning a visual-textual co-embedding (STEP) [25]. Additionally, STEP iteratively refines the textual embedding of the RE

Figure 6.1: Video sequences for DAVIS 2017 with language queries and our results. The first column shows a reference frame, the second to third columns depict the masks produced by our model when given the language query shown on top. Finally, the fourth to fifth columns show the results for the language query shown on top of these columns, which refers to another object of the video sequence.

with a Convolutional Recurrent Neural Network in a collaborative learning setup to improve the segmentation. An alternative consists in using off-the-shelf object detectors, like MAttNet [188]. In this case, a language attention network decomposes REs into three components: subject, location, and relationships, and merges the features obtained for each into single phrase embeddings. Given the object candidate by an off-the-shelf object detector model and a RE, the visual module dynamically weights scores from all three modules to fuse them.

RefVOS is a simpler model trained end-to-end that obtains a performance comparable to previous works on still images.

Language-guided Video Object Tracking: Object Tracking is a similar task to Video Object Segmentation as it also follows a referent across video frames, but in the tracking case the model localizes the object with a bounding box instead of a binary mask. Li et al. [96] and Feng et al. [50] tackle the object tracking problem given a linguistic phrase instead of using the bounding box at the first frame.

Our network provides pixel-wise segmentation masks that could be easily converted into bounding boxes, and at the same time avoid the annotation ambiguities present when bounding boxes overlap.

Language-guided Video Object Segmentation (LVOS): VOS [131, 181] has traditionally focused on semi-supervised setups in which a binary mask of the object is provided for the first frame of the video. Khoreva et al. [85] propose to replace the mask supervision with a linguistic expression. In their work, they extend the DAVIS-2017 dataset [135] by collecting referring expressions for the annotated objects. They provide two different kinds of annotations from two annotators each: *first frame* annotations are the ones that are produced by only looking at the first frame of the video, whereas *full video* annotations are produced after seeing the whole video sequence. They use the image-based MAttNet [188] model pretrained on RefCOCO to ground the localization of the referred object, and then train a segmentation network with DAVIS-2017 to produce the pixel-wise prediction. Temporal consistency is enforced, so that bounding boxes are coherent across frames, with a post-processing step. To the authors' knowledge, Khoreva et al. [85] is the only work previous to ours that focuses on REs for video object segmentation. Related work by Gavriluk et al. [55] addresses a similar task by segmenting video objects given a natural language query. They extend the Actor-Action Dataset (A2D) [179] by collecting phrases, but some of them may be ambiguous with respect to the intended referent, as they were not produced with the aim of reference, but description. The authors propose a model with a 3D convolutional encoder and dynamic filters that specialize to localize the referred objects. Wang et al [173] also leverages 3D convolutional networks, adding cross-attention between the visual and the language encoder.

We propose a simpler model trained end-to-end that treats each video frame independently and outperforms all previous works.

Referring Expression Categorization: RefCOCO, RefCOCO+ [189] and RefCOCOg [113] are datasets that provide REs for the still images in MSCOCO [101]. The datasets focus on different aspects related to the difficulty of REs: the REs for RefCOCO and RefCOCO+ were collected using the interactive ReferIt two-player game [82], designed to crowdsource expressions that uniquely identify the target referents. However, for

RefCOCO+, *location* information was disallowed. RefCOCOg, in turn, collected non-interactively, only contains *non-trivial* instances of target objects, that is, there is at least one other object in an image of the same class. The CLEVR dataset [81] contains objects of certain shapes, attributes such as sizes and colors, and spatial relationships. CLEVR uses synthetic images and phrases designed to test VQA systems, while our work focuses on human-produced language and natural videos.

Khoreva et al. [85] categorize the REs they collected for DAVIS-2017 in order to analyze the effectiveness of their proposed model. This is similar to our work, however, while they distinguish REs according to their length and whether they contain spatial words (e.g., *left*) or verbs, we propose a more fine-grained, semantic categorization that also distinguishes between different aspects of verb meaning related to motion and object relations. Khoreva et al. [85] furthermore analyze the REs in DAVIS-2017 with respect to the parts of speech they contain, while we use our *semantic* categories for dataset analysis.

6.3 Model

We address the task of language-guided image segmentation with the deep neural network depicted in Figure 6.2, that we call RefVOS. This model operates at the frame level, i.e., treats each frame independently, and is thus applicable for both images and videos. It uses state of the art visual and language feature extractors, which are combined into a multi-modal embedding decoded to generate a binary mask for the referent.

Visual Encoder: To encode the images we rely on DeepLabv3, a network for semantic segmentation based on atrous convolutions [29]. We use DeepLabv3 with a ResNet101 [66] backbone and output stride of 8. The Atrous Spatial Pyramid Pooling (ASPP) has atrous convolutions with 12, 24 and 36 rates.

Language Encoder: In contrast to previous works addressing language-guided image segmentation, our work was the first one to leverage the bidirectional transformer model BERT [44] as language encoder. For our pipeline, we use BERT to obtain an embedding for the linguistic phrases. First of all we fine-tune BERT with the REs of RefCOCO with the masked language modelling (MLM) loss for one epoch, which consists in randomly masking a percentage of input tokens and then predicting them, following the common fine-tuning procedure for BERT. We then integrate BERT into our pipeline and fine-tune it specifically towards the language-guided image segmentation task: to this end we tokenize the linguistic phrase and add the [CLS] and [SEP] tokens at the beginning and end of the sentence respectively. BERT produces a 768-dimensional embedding for each input token. We adopt the procedure of Devlin et al. [44] and extract the embedding corresponding to the [CLS] input token, i.e., the *pooled output*, as it aggregates a representation of the whole sequence.

Multi-modal Embedding: To obtain a multi-modal embedding, the encoded linguistic phrase is first converted to a 256-dimensional embedding with a linear projection and then element-wise multiplied with the visual features extracted by the ASPP from DeepLabv3. We noted that the multiplication yielded better performance than addition or concatenation, as depicted in Table 6.1. A convolutional layer then predicts two maps, one for the *foreground* and another for the *background* class. We employ the cross entropy loss commonly used for segmentation.

	val	testA	testB
Concatenation	55.12	58.88	49.59
Addition	56.60	60.87	51.29
Multiplication	**59.45**	**63.19**	**54.17**

Table 6.1: Comparative study about different fusion strategies between visual and language features on RefCOCO.

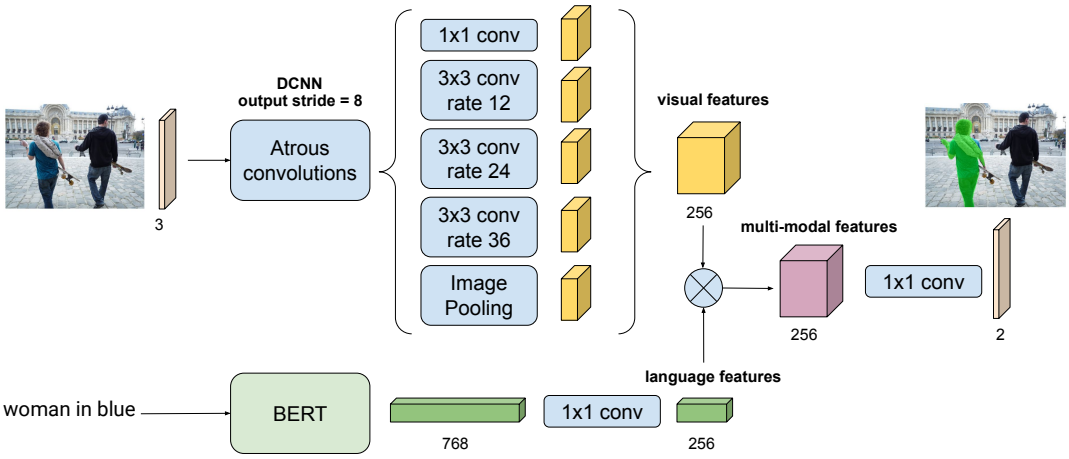

Figure 6.2: Architecture of our model, namely RefVOS.

6.4 Referring Expression Categorization

We propose a novel categorization for referring expressions (REs), i.e., linguistic phrases that allow the identification of an individual object (the *referent*) in a discourse or scene. This categorization is adapted to the challenges posed by the VOS task. This categorization will later be used to assess VOS benchmarks and also to analyze the performance of video models. We follow the commonly adopted definition of REs put forward by computational linguistics and natural language processing (e.g., [139]), and consider a (noun) phrase as a RE if it is an accurate description of the referent, but not of any other object in the current scene. Likewise, in Vision & Language research, visual RE resolution and generation has seen a rise of interest, especially in still images [37, 113, 104, 190, 116], and more recently also on videos [4, 31]. The task is formulated as, given an instance comprising an image or video with one or multiple objects, and a RE, identify the *referent* that the RE describes by predicting, e.g., its bounding box or segmentation mask. The difficulty of the task increases with the number of objects appearing in the scene, and the number of objects of the same class. Such cases require more complex REs in order to identify the referent.

In order to make progress on VOS with REs and allow for a systematic comparison of methods, benchmark datasets need to be challenging from both, the visual and linguistic perspective. However, for example, most video sequences in the DAVIS-2017 dataset used in Khoreva et al. [85] show a single object in the scene or, at most, different objects from different classes. In these cases, the actual challenge is that of predicting accurate object masks for the RE. On the other hand, the existing datasets for VOS with REs

do not focus on the particularities that video information provides either, and often use object attributes which can be already captured by a single frame, or are not even true for the whole clip (e.g. the A2D dataset provides phrases for only a few frames per clip).

Our novel categorization of REs for video objects allows the analysis of datasets with respect to the *difficulty* of the REs and the kind of *semantic information* they provide. We apply it to label and analyze existing REs of DAVIS-2017 and A2D. In addition, we use this categorization to extend a subset of the A2D test set with REs which contain semantically varying information to analyze how our model behaves with respect to the different categories.

6.4.1 Difficulty and Correctness of Datasets

We first assess the validity and visual difficulty of a subset of DAVIS-2017 and A2D, by classifying each instance (an object and its RE) into *trivial* or *non-trivial*: if the referent is not the only object of a certain object class in the video we consider it *non-trivial*, otherwise *trivial*. A trivial case would be a video with a single *elephant*, because the class category is enough to indicate the target object. A non-trivial case would be if there is more than one *elephant* in the video, as then a more complex description is required to uniquely identify each instance. We further label each phrase according to its linguistic ambiguity and correctness: we mark it as *no RE* if its referent is not the only object in the video which could be described by the phrase, and as *wrong object* if it does not match the referent.

Data and Annotation Procedure: Annotation was performed on the DAVIS-2017 validation set (61 REs provided by *annotator 1* [85]) in the full video setup (see Section 6.2), as well as on the subset of the A2D test set which contains at least two annotated objects (856 instances). Each instance contained therein was annotated by one out of four persons (all co-authors). Note that we assume the instances in A2D videos with only a single annotation as *trivial*, and automatically labeled them as such (439 instances).

Results: Figure 6.3 shows the proportion of phrases in the DAVIS-2017 and A2D sets with respect to their difficulty and correctness. Despite being collected in a (non-interactive) referential two-player game setup, DAVIS-2017 contains a considerable proportion of ambiguous phrases (*no RE*, 8%). The proportion in A2D is slighlty higher (11%), but note that A2D was designed to contain descriptive phrases in contrast to unique identifiers (as defined above). About 52% in DAVIS, and 35% in A2D are *non-trivial* phrases, that is, more challenging for language-guided VOS from both, the linguistic and visual perspective, since the object class itself is not sufficient to identify the correct referent.

6.4.2 Semantic Categorization of REs

Our categorization is inspired by semantic categories of situations and utterances in linguistics [92, 60], tailored to the situations found in video data. Specifically, we analyze the REs with respect to the type of information they express, by assigning them categories assumed to be relevant for reference to objects in visual scenes. We focus on information relevant for both, objects in still images and videos, namely the *class category*, *appearance*, and the *location* of the referent, and distinguish between information assumed to be more relevant for videos only, namely *motion* vs. *static* events. If, according to the

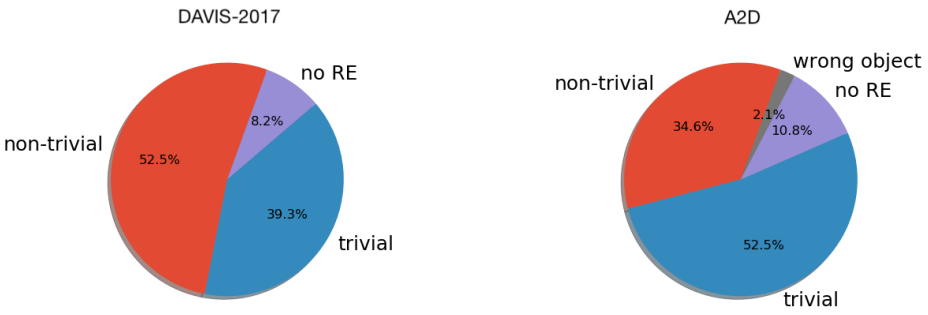

Figure 6.3: Proportion of expressions in the val set of DAVIS-2017 and the test set of A2D by the difficulty and correctness of the REs.

Semantic Categories	**Q: Does RE tell you about referent r...**	**Example**
appearance	how r looks like?	...*in a yellow dress*...
class category	r's name or category (noun)	...*seagull*...
location	where r is located? (rel. to image/other object)	...*near tractor*...
motion	if r moves or changes its location?	...*walking*...
obj-motion	if r moves or changes another object's location?	...*riding a bike*...
static	what r is doing (if not moving)?	...*eating*...
obj-static	if r acts on another object (no motion)?	...*holding a bike*...

Table 6.2: The semantic categories used for annotation.

RE, the referent acts upon other objects in the scene, we distinguish between whether an object is moved by the referent or not (*obj-motion* vs. *obj-static*). This information may be particularly valuable for models that reason over object interactions.

(Psycho)linguistic studies have observed a tendency of REs to contain redundant nondiscriminating information, i.e., logically more information than required to establish unique reference, arguably because this reduces the effort needed for identification (e.g., [68, 92]) In particular the kind (class category) of the object and salient properties such as color (e.g., [151]) have been found to be used redundantly. To assess whether the phenomenon of redundancy is born out in the video datasets, we additionally label instances as *redundant* or *minimal*.

Data and Annotation Procedure: We collect annotations for the same 61 instances of the validation set of DAVIS-2017 as above, and for a subset of the test set of A2D, which we call *A2Dre* henceforth. We obtain A2Dre by selecting only instances that were labeled as *non-trivial*, which are 433 REs from 190 videos. We do not use the *trivial* cases as the analysis of such examples is not relevant, as *referents* can be described by using the *class category* alone. Each annotator was presented with a RE, a video in which the target object was marked by a bounding box, and a set of questions paraphrasing our categories (see Table 6.2). Three annotators (all co-authors of the paper) individually labeled all instances of the DAVIS-2017 val set, then jointly discussed their disagreements, and again individually revised their annotations for possible errors or other unclear cases. The inter-annotator agreement can be considered substantial for all categories, with Davies & Fleiss' kappa coefficients [42] between $\kappa = .83$ and .97 (except *obj-static*, $\kappa = .35$, which has only

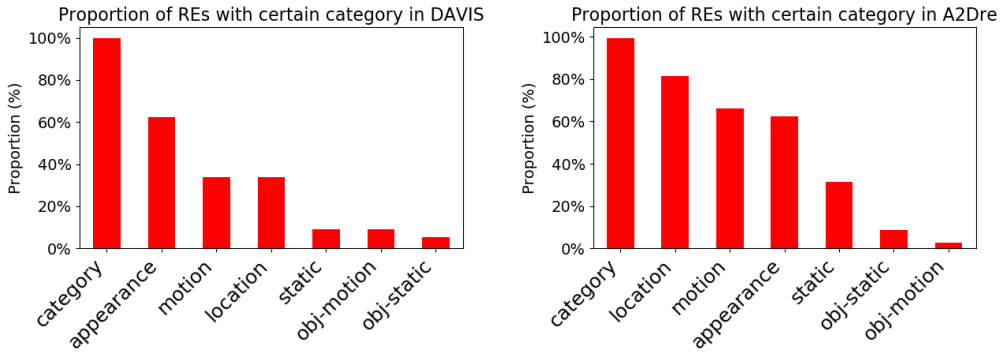

Figure 6.4: REs in the validation set of DAVIS-2017 and A2Dre with respect to their categories. The *class category* is referred as *category* for reduced space in the plot.

5 positively labeled instances by at most 2 annotators, and *class category* which obtained perfect agreement). A2Dre was subsequently annotated by the same 3 annotators. Our final set of category annotations used for analysis was derived by means of majority voting: for each *non-trivial* RE, we kept all category labels which were assigned to the RE by at least two annotators.

Results: What kind of information do REs express? First of all, we found 99% of the REs for non-trivial instances in A2Dre, and 66% in DAVIS-2017 val (74% including trivial), respectively, to contain redundant information. Recall that only the REs in DAVIS-2017 were obtained in a referential setup, thus relatively larger proportion of redundant REs in A2D is not surprising.

Figure 6.4 shows the proportion of instances in the two datasets (DAVIS-2017 val and A2Dre) that were labeled with the individual categories. As expected, the name or *class category* of the referent is virtually always expressed. The visual properties of the referent, i.e., *appearance*, is prominent in both datasets, too (approx. 60%). Taken together with their high redundancy ratio, this confirms what has been found in psycholinguistic studies on reference [92]. The remaining categories, however, are rare in both datasets, or are only highly frequent in A2Dre, with *location* and *motion* being used in the majority of REs. That A2Dre comprises more complex REs than DAVIS-2017 may be not only due to their collection as descriptive, instead of discrimininative phrases, but also due to the much higher complexity of the video scenes. Note that information about referent-object interactions (*obj-static* and *obj-motion*) is neglectable, which illustrates the datasets' limited usefulness for research on reasoning over object interactions [174, 193, 185]. In the experiments we report in Section 6.5, we discard these categories, and focus on the remaining categories only, for which we augment the A2Dre dataset.

6.4.3 Extending A2D with REs

As explained above, A2Dre is a subset from the A2D test set including 433 *non-trivial* REs. Due to its highly unbalanced distribution across the 7 semantic categories (Figure 6.4), we select the 4 major categories *appearance, location, motion and static*. The four categories have in common that in most cases, for a given referent, a RE can be provided that expresses a certain category, and one that does not. We use these categories to

augment A2Dre with additional REs, which vary according to the presence or absence of each them. Specifically, based on our categorization of the original REs, for each RE re and category C, we produce an additional RE re' by modifying re slightly such that it does (or does not) express C. For example, if we have the last RE from Figure 6.6, i.e. *girl in yellow dress standing near the woman*, which could be categorized as *appearance*, *location*, no *motion* and *static*, we produce new REs for each category: *girl standing near the woman* (no *appearance*), *girl in yellow dress standing* (no *location*), *girl in yellow dress walking* (*motion*) and *girl in yellow dress near the woman* (no *static*). We do not apply this procedure for *class category*, since it is expressed in almost all REs, and its removal may be difficult in many cases.

6.5 Experiments

We report results with our model on two different tasks: *language-guided image segmentation* and *language-guided video object segmentation*. The results for still images are obtained on RefCOCO and RefCOCO+ [189], while those for video correspond to DAVIS-2017 and A2D.

6.5.1 Language-guided Image Segmentation

The impact of BERT embeddings in our model on both RefCOCO and RefCOCO+ can be assessed in Table 6.3, compared with a bidirectional LSTM similar to Chen et al. [25] for encoding the linguistic phrase. In particular, we average the GloVe embeddings [128] of each token and concatenate the mean embeddings of the forward and backward pass. This baseline is compared to two configurations that use BERT. The first fine-tunes BERT for the language-guided image segmentation task, and significantly boosts performance over using GloVe embeddings. The second has an additional step, that consists in first training BERT with the masked language modelling loss with the REs from RefCOCO, as explained in Section 6.3, and then fine-tuning BERT on the language-guided image segmentation task (as in the previous configuration). We see that this configuration brings an additional gain.

Table 6.3 also compares our model with the state of the art on language-guided image segmentation. STEP [25] yields the best performance for both datasets. It consists in an iterative model that refines the RE representation to improve the segmentation. Note that the model must be run for each iteration. Our model surpasses STEP (1-fold) on RefCOCO val and testA, which corresponds to a comparable computational cost, and is still slightly better than STEP (4-fold). Compared to STEP (5-fold), the performance of our method is slightly lower.

Qualitative results generated with our best model on RefCOCO are depicted in Figure 6.5. We note how our model distinguishes properly the referred instance and generates an accurate mask.

We conclude that our approach is competitive with the state of the art for language-guided image segmentation. Hence, *RefVOS* is a valid model to be exploited for language-guided VOS.

	RefCOCO			RefCOCO+		
	val	testA	testB	val	testA	testB
Ours with Bi-LSTM	48.46	52.90	44.43	35.35	40.72	28.43
Ours with BERT	58.65	62.28	54.28	42.07	46.46	34.23
Ours with BERT Pre-train	59.45	63.19	54.17	44.71	49.73	36.17
MattNet	56.51	62.37	51.70	46.67	52.39	40.08
CMSA	58.32	60.61	55.09	43.76	47.60	37.89
LANG2SEG	58.90	61.77	53.81	-	-	-
STEP (1-fold)	56.58	58.70	55.39	-	-	-
STEP (4-fold)	59.13	-	-	-	-	-
STEP (5-fold)	**60.04**	**63.46**	**58.97**	**48.18**	**52.33**	**40.41**

Table 6.3: Overall IoU for RefCOCO and RefCOCO+.

Figure 6.5: Visualizations of RefCOCO *testA* and *testB* sets.

6.5.2 Language-guided VOS

Our model is assessed for LVOS on DAVIS-2017 and A2D. In both cases, each video frame is treated separately, so we use the same architecture as in the still image experiments from Section 6.5.1.

Our experiments on the DAVIS-2017 validation set are reported in Table 6.4. All models are pre-trained on RefCOCO. Results are provided with the J&F metric adopted in the DAVIS-2017 challenge for the two different types of REs collected by Khoreva et al. [85] explained in Section 6.2. J&F is the average between a region-based evaluation measure (J) and a contour-based evaluation measure (F). The region-based evaluation measure J is computed in the same way as IoU. On the other hand, the contour-based evaluation measure F is computed as follows: first, pixel boundaries for both predicted and ground truth masks are obtained. Then, these pixel boundary masks are dilated using a morphological operation. Finally, precision and recall measures are computed on these dilated boundary masks and F-measure is computed as $2 * precision * recall / (precision + recall)$.

Model	+Ft DAVIS segms.	+Ft DAVIS REs 1st frame	+Ft DAVIS REs full video	J&F 1st frame	J&F full video
Khoreva et al. [85]	✓			39.3	37.1
URVOS [156]	✓	✓		44.1	-
RefVOS				39.8	40.8
	✓			42.0	42.0
	✓	✓		44.5	**45.1**
	✓		✓	42.7	**45.1**

Table 6.4: J&F on DAVIS-2017 validaton set.

	Prec @0.5	Prec @0.9	IoU Overall	IoU Mean
Gavriluk et al. [55]	50.0	0.4	55.1	42.6
Wang et al. [173]	55.7	2.0	60.1	49.0
RefVOS with A2D	49.5	6.4	59.9	43.0
RefVOS with RefCOCO	27.9	3.4	41.4	25.6
+ finetuned on A2D	**57.8**	**9.3**	**67.2**	**49.7**

Table 6.5: *Precision*, *overall IoU* and *mean IoU* on A2D.

Our experiments indicate that our baseline model trained only with RefCOCO already outperforms the best model by Khoreva et al. [85], despite the latter being fine-tuned on the same DAVIS-2017 dataset (+Ft DAVIS segms.). The difference increases when our model is fine-tuned with the segmentations provided in the training set, but freezing the language encoder. This is the configuration comparable to Khoreva et al. [85] in terms of training data, and brings gains of 2.7 and 4.9 points for the *first frame* and *full video* REs, respectively. Finally, we also fine-tune the BERT language encoder, obtaining a significant extra gain in performance. We want to highlight that our frame-based model does not rely on any post-processing to add temporal coherence, or optical flow, in contrast to Khoreva et al. [85], so our method may be more efficient computationally. We also compare our model to URVOS [156], a concurrent work to ours. URVOS is a model for language-guided video object segmentation, and is composed of a cross-modal attention module for the visual and lingual features, and a memory attention module to leverage information from past predictions in a video sequence. Compared to ours, their architecture is more complex due to the cross-attention and memory network. Our model performs slightly better when trained with the same amount of annotated data. Qualitative results for full video REs for our model are presented in Figure 6.1. When the multiple objects belong to different class categories, the model works properly and produces accurate masks from the language query, whereas it is more challenging to properly segment the referent in cases where there are multiple instances of the same class in the sequence (3rd row). The fine-tuning is done with the *full video* REs, and the REs shown in Figure 6.1 are of the same kind. We note how the referred object is in general identified and properly segmented.

The results for A2D are shown in Table 6.5, using the evaluation methods that allow us a comparison with previous works [55, 173]. Following we provide a brief description of

the metrics for this benchmark:

- Overall IoU: Intersection area of all test data over the total union area.
- Mean IoU: Average over the IoU of each test sample so that large and small regions are treated equally.
- Precision@X: Given a threshold X, e.g. $X = 0.5$, a predicted mask for an instance is counted as true positive if the IoU is larger than X, and as false positive otherwise. Then, Precision@X is computed as the ratio between the number of true positives and the total number of instances.

For each benchmark we report the evaluation metrics commonly used. Therefore, Overall IoU is reported for RefCOCO and RefCOCO+ datasets, J&F is reported for DAVIS-2017 dataset, and Precision@{0.5,0.6,0.7,0.8,0.9}, Overall IoU and Mean IoU are reported for A2D dataset.

Our model trained only with A2D already outperforms Gavriluk et al. [55] in *Precision* at a high threshold and at the *Overall* and *Mean Intersection Over Union (IoU)*. Moreover, our model significantly increases its performance when it is first trained on RefCOCO and later fine-tuned on A2D, both its visual and language branches. In this setup, it achieves state of the art results in all metrics by significant margins. Note that both Gavriluk et al. [55] and Wang et al. [173] leverage an encoder pre-trained on the Kinetics dataset, which includes 650,000 video clips [22]. Hence, these models see a large amount of annotated data for action recognition in videos. We also want to stress our high *Precision* values at high thresholds, which indicates that our model is able to produce very accurate masks. Visualizations with our model are illustrated in Figure 6.6.

In conclusion, RefVOS outperforms all previous works for DAVIS-2017 and A2D on the LVOS task, although it is a frame-based model. This motivates the analysis of our model when tested with different types of REs, based on the categorization and difficulty analysis proposed in Section 6.4.

Referring Expressions Analysis: Firstly, we analyze the performance on *trivial* and *non-trivial* linguistic phrases for both the A2D test and DAVIS-2017 validation sets. The *mean IoU* per referent obtained for *trivial* and *non-trivial* for DAVIS-2017 is 48.7 *vs.* 46.2, and for A2D is 53.9 *vs.* 33.2. We observe that the performance is worse for the *non-trivial* cases for both datasets as expected, with a major drop on A2D.

Secondly, we study the effect of RE categories in relation to the performance of RefVOS. The A2Dre+ dataset described in Section 6.4.3 allows us to have the same number of referents for all major categories: *appearance, location, motion* and *static*. Each of our referents is annotated with highly similar REs (two for each category) and thus are directly comparable. In contrast, Khoreva et al. [85] split the videos into two different subsets with non-comparable referents. Table 6.6 compares the performance of RefVOS depending whether each of the categories is present in the RE. The results show that the presence of image-based categories, such as *appearance* and *location*, yields significantly higher results compared to their absence. Regarding video-based categories, we observe a drop in performance when the *static* category is present, which indicates that the model struggles at identifying a referent based on static actions such as *holding, sitting,*

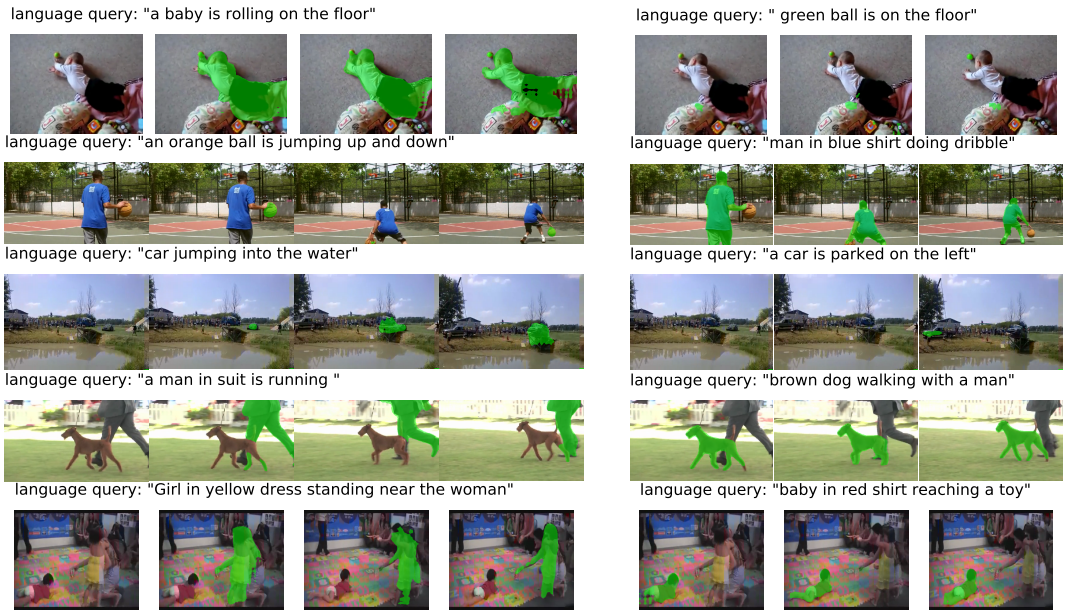

Figure 6.6: Video sequences for A2D with language queries and the results of our model. The first column shows a reference frame, the second to fourth columns depict the masks produced by our model when given the language query shown on top. Finally, the fifth to seventh columns show the results for the language query shown on top of these columns, which refers to another object of the video sequence.

Image-based				Video-based			
+App	-App	+Loc	-Loc	+Motion	-Motion	+Static	-Static
33.90	30.15	**34.15**	30.78	**35.58**	35.60	34.28	**36.21**

Table 6.6: Effect of the presence of categories in REs.

eating. In contrast, the presence or absence of the *motion* category does not affect the performance, which actually means that the model is unable to benefit from this type of REs.

Following we further analyze visually the results obtained with RefVOS depending on the categories appearinig in REs. Figure 6.7 includes examples of the results of our model with A2Dre+. Each column is a first frame of a video sequence with a *non-trivial* case, and each row is a different RE that has or has not a certain category. As we concluded with the numerical results, the performance when the *appearance* and *location* categories are present is higher compared to when these categories are absent. Regarding the *motion* and *static* categories, we first notice that some REs are not created as the annotators considered that it was not possible. We indicate those examples with the "Not Applicable" label. We see how the presence or the absence of the *motion* and *static* categories has a minimal impact to the results. In fact, adding these categories even leads to worse segmentations, as it happens with the example *"a man eating a big sandwich"* from the second column.

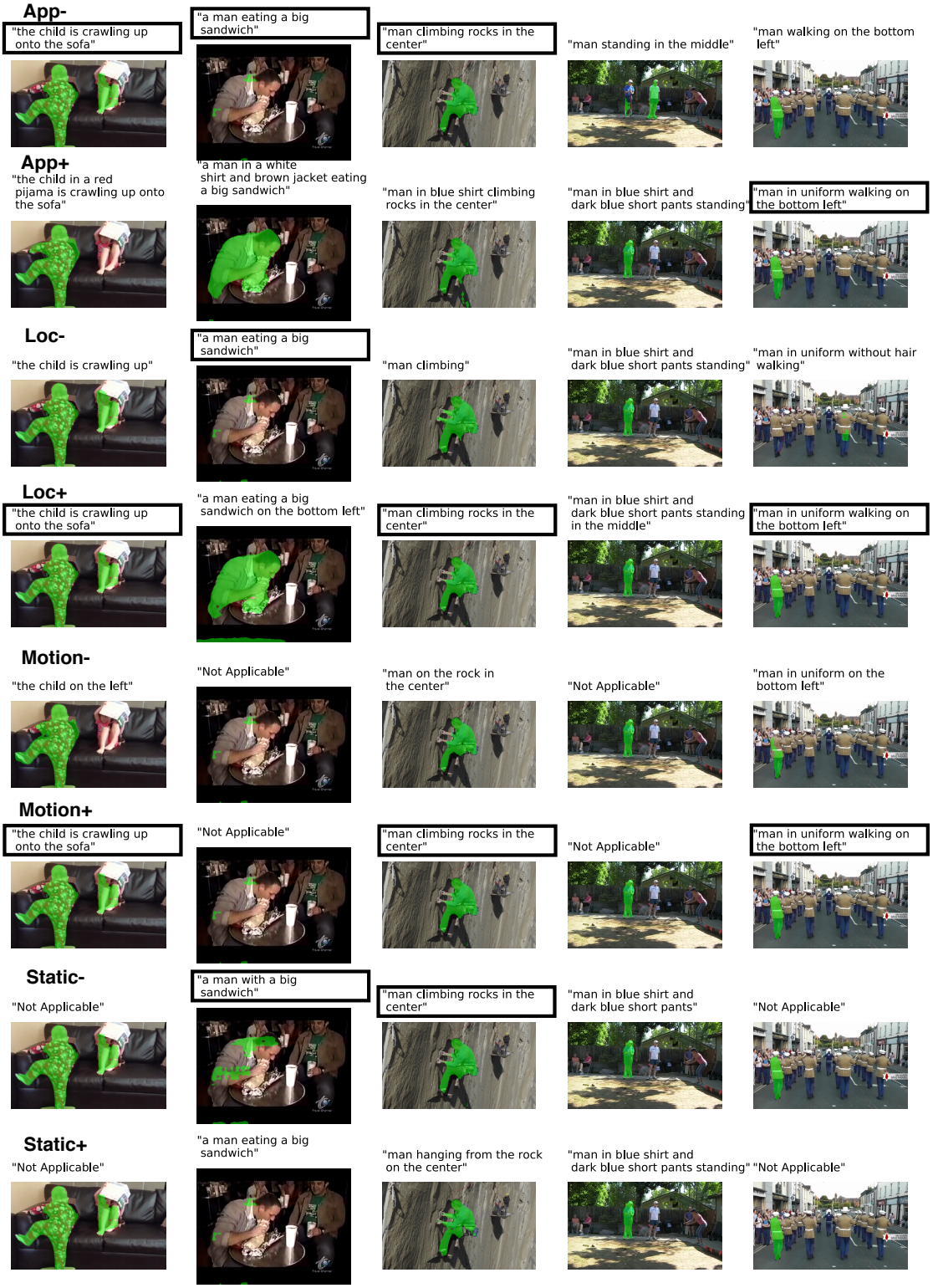

Figure 6.7: Each column is the first frame of a video sequence of A2D. Each row indicates if the RE that produces the depicted result contains or not a certain category from our proposed categorization. The REs that are inside a box are the original REs in the dataset [55]. For the example in the fourth column, the natural expression in the annotations from Gavrilyuk et al. [55] is *man standing*, which is not a RE as it does not uniquely identify the target object. For this reason, for this example all REs shown in the Figure are our own annotations.

	Overall IoU			Mean IoU		
	Trivial	Non-Trivial	All	Trivial	Non-Trivial	All
Generic	45.6	18.1	41.6	34.6	10.0	29.6
Only Actor	65.6	34.8	60.8	51.5	22.8	45.7
Only Action	56.3	30.7	52.6	43.0	18.5	38.0
Actor + Action	66.6	37.3	62.2	51.3	24.8	45.9
Full phrase	**70.2**	**47.5**	**67.2**	**53.9**	**33.2**	**49.7**

Table 6.7: *Overall* and *Mean IoU* on A2D for different levels of information in REs.

Finally, in Table 6.7 we study the effect of feeding the model with only the *actor*, the *action*, or the *actor and action*, without formulating any RE, for all the test set of A2D. These *actor* and *action* terms are obtained from the dataset collected by Gavriluk et al. [55]. In most cases these expressions are not REs as they do not unambiguously describe the referent in the video. Additionally, we consider a generic phrase *thing*. We distinguish between the *trivial* and *non-trivial* cases. Results show that RefVOS works significantly better when the *actor* is provided than when the *action* is. Furthermore, performance improves when using both. Finally, having the full linguistic phrase is still the best model. Remarkably, our configuration with *actor and action* reaches higher *Overall IoU* than previous works that use complete linguistic phrases (see Table 6.5). Notice that using the full phrase improves performance especially for the *non-trivial* cases, as these require complete linguistic expressions to identify the *referent*. We also want to stress that the aggregated performance, i.e., considering *all* cases, is dominated by the performance of the *trivial* ones, as they represent most of the dataset.

6.6 Training Details

We used PyTorch framework to develop the models. DeepLabv3 model is the one provided by torchvision, and BERT model is the one provided by HuggingFace's transformer library [177]. In order to train the model on RefCOCO and RefCOCO+, we use pre-trained weights of Imagenet for the backbone model.

We use SGD optimizer with 0.9 of momentum and 1e-6 of weight decay. We train our model with batch size of 7 and 480x480 of resolution. The learning rate scheduler for RefCOCO and RefCOCO+ consists in first starting with learning rate of 0.01 and decrease it 0.0004 every epoch until reaching 3e-3 of learning rate. Then we increase the learning rate again to 6e-3 and decrease it 2.5e-4 every epoch until reaching 1e-3. Increasing the learning rate after some iterations has been proven to increase performance in previous works [108]. Finally we set a final state with fixed learning rate of 3e-5. To later fine-tune on DAVIS, we set an initial learning rate of 1e-3 and decrease 1e-5 every epoch for 20 epochs. To train the model with A2D we set a learning rate of 0.1 and decrease it 0.004 for 15 epochs. We used a single Tesla V100 GPU to train the models. The average time for training the model with RefCOCO is about 3 days, 2 days for A2D and 1 day for DAVIS-2017 with our machines. The approximate number of parameters of our model, considering also the BERT encoder, is 170M.

6.7 Conclusions

In this Part we study the task of language-guided video object segmentation, as a manner to alleviate the struggle to delineate the objects to be segmented as it is required in the one-shot VOS task. Thus, we reduce the human effort required at inference in the VOS task.

We focus on studying the difficulty of REs from benchmarks on LVOS, and propose seven semantic categories to analyze the nature of such REs. We introduce RefVOS, a novel model that, compared to previous works, is competitive for language-guided image segmentation, and state of the art for language-guided VOS. However, our analysis shows that benchmarks are mainly composed of trivial cases, in which referents can be identified with simple phrases. This indicates that the reported metrics for the task may be misleading. Thus, we focus on the non-trivial cases. We extend A2D with new REs with diverse semantic categories for non-trivial cases, and test our model with them, which reveals that it struggles at exploiting motion and static events, and that it mainly benefits from REs based on appearance and location. We reckon that future research on LVOS should focus on non-trivial cases describing motion and events, as they present a challenge for language grounding on videos. Concurrent to our work, Seo et al. [156] collected Refer-Youtube-VOS, a large-scale benchmark for language-guided video object segmentation built on top of Youtube-VOS [182]. We believe that, as future work, our categorization for REs could be used to classify the provided language expressions by this benchmark. Thus, models could be evaluated based on the non-trivial cases and the different categories in order to analyze which REs are more challenging when using a large-scale dataset.

The presented models, source code and extended dataset of REs are publicly available and can be found in https://github.com/miriambellver/refvos.

7 Conclusions

This thesis has addressed the task of instance segmentation for both image and video, being the guiding thread the level of supervision applied either at training or at inference. Our main objective is to lower the human effort required. Following, we summarize the different goals and contributions for each part of the thesis:

The first research question formulated in Part I of the thesis was **if it was possible to train fully end-to-end instance segmentation architectures leveraging Recurrent Neural Networks**, specially focusing on video systems. Our main contribution is the development of RVOS, a recurrent model for video object segmentation that is end-to-end trainable and that is trained in a fully-supervised way, and thus it does not require any post-processing step. Our model can segment multiple objects and frames in a single stage. This project was open-sourced, which has enabled the research community to build on top of our architecture, contributing to the general interest towards end-to-end trainable models. Another goal for Part I was **to design a video system that did not rely on any initialization cue to discover objects along the video sequence**. In this dissertation we present the first solution for zero-shot video object segmentation, i.e., a model that segments objects from a sequence that is completely unsupervised at inference, and thus does not require any effort at test time.

The tools developed for Part I of the thesis were crucial to identify the main challenge of image segmentation models: advancing towards systems that demand less annotation effort. In Part II the main goal was to reduce the annotation time required during training. Hence, we shift our focus from fully-supervised systems to semi-supervised ones. The research question addressed was if **it is possible to train semi-supervised systems for segmentation on very low annotation budgets**. As a solution, we propose BASIS, a semi-supervised pipeline based on self-learning that leverages a limited amount of strongly-labeled data and larger amounts of unlabeled or weakly-labeled samples. Our experimental validation is tested on two different tasks for segmentation, and for varying annotation budgets. Our main contribution is that we experimentally show that, when considering matching annotation costs, having few but accurate strong labels leads to better results than having a larger amount of weakly-annotated data. We consider that this outcome can be relevant for the community, as it can be used to assess how to spend some pre-defined budget when annotating a new dataset. Furthermore, another contribution of our experimental validation is that, compared to previous works, we show results for image segmentation when the annotation cost is remarkably low, which can be a referent for future research. Additionally, in Part II of the dissertation we explore the effect of using an active learning mechanism to select which samples to strongly-annotate. Given a pool of weakly-labeled data, we want to know which samples are better to further annotate with pixel-wise labels. We contribute with a novel mechanism that predicts the

model confidence about a prediction of a given sample, by estimating the intersection over union of the predicted masks with the ground truth. Moreover, we conclude that for our pipeline, the best samples to choose are those that are neither the most complex, nor the easiest ones of the dataset. In other words, those images that are not trivial for the model to resolve, as they do not bring any extra gain, and also discarding those samples that are so complex that could be considered outliers.

In Part III we target to ease the human intervention required at inference time, in contrast with Part II, where we focused on the annotation cost of the training data for segmentation models. Particularly, our first research question was **if language could help to reduce the human effort required at inference time in semi-supervised VOS systems**. As a solution, we present RefVOS, a model that addresses language-guided video object segmentation. Our method surpasses previous works and is the first method to exploit BERT language model [44] for this task. The second research question was **if current benchmarks are suitable for the video task**. The main contribution of Part III is an analysis on current benchmarks addressing language-guided video object segmentation, and our proposal on novel semantic categorization of referring expressions that attend to the intrinsic challenges of video. To lead a thorough study, we augment the phrases provided by these benchmarks, and analyze which type of expressions are more challenging for video objects based on our categorization. We concluded that object descriptions based on motion and static events were the hardest to comprehend by our model. We believe that our novel categorization can be a valuable contribution to the community to assess video models, in order to identify their performance for different types of language expressions. We open-sourced this project, and we hope that this helps to reproduce our results and build on top of our model for future research on the topic.

Future Work

As a conclusion, in this thesis we explored different supervision scenarios for instance segmentation models, distinguishing between supervision when training and at inference. We believe that, due to the wide range of applications of image segmentation and the expensive cost of pixel-level annotations, future work will still focus on lowering the annotation cost using semi-supervised or weakly-supervised pipelines, or even with fully-unsupervised systems through self-supervised learning. However, as we saw in our work, accurate annotations, even if only having a few of them, are crucial for good performance in current models. In the last Part of the thesis we concluded that language is a powerful input for semi-automatic systems. Nevertheless, we observed that there is still potential of improvement to make our models comprehend descriptions of objects in videos based on their dynamics. An interesting line of research to explore would be to fully exploit video, by also extracting features from the audio signal. Therefore, we would approximate to a system that is able to understand vision, language and audio all together.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. 18

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 50, 67

[3] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 49, 52

[4] Hazan Anayurt, Sezai Artun Ozyegin, Ulfet Cetin, Utku Aktas, and Sinan Kalkan. Searching for ambiguous objects in videos using relational referring expressions. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 91

[5] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 30

[6] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[7] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. *arXiv preprint arXiv:2003.08429*, 2020. 43

[8] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[9] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 27, 30, 39

[10] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 47, 49, 52, 53

[11] Míriam Bellver, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Xavier Giró-i Nieto, Jordi Torres, and Luc Van Gool. Detection-aided liver lesion segmentation using deep learning. *Machine Learning 4 Health Workshop in Neurips*, 2017. 9

[12] Miriam Bellver, Amaia Salvador, Jordi Torres, and Xavier Giro-i Nieto. Mask-guided sample selection for semi-supervised instance segmentation. *Multimedia Tools and Applications*, 79(35):25551–25569, 2020. 8

[13] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i Nieto. Refvos: A closer look at referring expressions for video object segmentation. *arXiv preprint arXiv:2010.00263*, 2020. 8

[14] Míriam Bellver Bueno, Amaia Salvador Aguilera, Jordi Torres Viñals, and Xavier Giró Nieto. Budget-aware semi-supervised semantic and instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019*, pages 93–102, 2019. 8

[15] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, et al. The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056*, 2019. 9

[16] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. Active learning for deep object detection. *arXiv preprint arXiv:1809.09875*, 2018. 70

[17] Míriam Bellver Bueno, Xavier Giró-i Nieto, Ferran Marqués, and Jordi Torres. Hierarchical object detection with deep reinforcement learning. *Deep Learning for Image Processing Applications*, 31(164):3, 2017. 9

[18] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 27, 30, 37, 38, 39

[19] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019. 43

[20] Victor Campos Camunez, Francesc Sastre, Maurici Yagües, Míriam Bellver, Xavier Giró Nieto, and Jordi Torres Viñals. Distributed training strategies for a computer vision deep learning algorithm on a distributed gpu cluster. In *Procedia Computer Science*, pages 315–324. Elsevier, 2017. 10

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 5, 43

[22] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 98

[23] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 50

[24] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. Semi-supervised learning, vol. 2. *Cambridge: MIT Press. Cortes, C., & Mohri, M.(2014). Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science*, 519:103126, 2006. 19, 20, 52

[25] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 87, 88, 95

[26] Liang-Chieh Chen, Sanja Fidler, Alan L Yuille, and Raquel Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 50

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 14

[28] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 14

[29] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 14, 15, 90

[30] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 55, 65

[31] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. *arXiv preprint arXiv:1812.03426*, 2018. 91

[32] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 16

[33] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019. 51

[34] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 27, 30, 38, 39

[35] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 27, 30, 31

[36] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 55, 65

[37] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. *AAAI*, 2018. 91

[38] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 50

[39] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 13

[40] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2015. 47, 49, 52

[41] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 13

[42] Mark Davies and Joseph L. Fleiss. Measuring Agreement for Multinomial Data. *Biometrics*, 38(4):1047–1051, 1982. 93

[43] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. In *Workshop in conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019. 23, 88, 90, 104

[45] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997. 20

[46] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. 70

[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge. *IJCV*, 2010. 7, 57

[48] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010. 50, 52, 53, 55, 70, 73

[49] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. 70

[50] Qi Feng, Vitaly Ablavsky, Qinxun Bai, Guorong Li, and Stan Sclaroff. Real-time visual object tracking with natural language description. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 700–709, 2020. 89

[51] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013. 21

[52] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016. 70, 73

[53] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017. 70, 76

[54] Mingfei Gao, Ang Li, Ruichi Yu, Vlad I Morariu, and Larry S Davis. C-wsl: Count-guided weakly supervised localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 50, 53, 60

[55] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 87, 88, 89, 97, 98, 100, 101

[56] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 12, 77

[57] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 12

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 17

[59] Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017. 70, 76

[60] Birgit Hamp and Helmut Feldweg. GermaNet - a lexical-semantic net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997. 92

[61] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 55, 64, 66, 73, 80

[62] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 16

[63] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 16

[64] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 12, 16

[65] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 52

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 17, 60, 90

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 32

[68] Raquel Hervás and Mark Finlayson. The prevalence of descriptive referring expressions in news and narrative. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 49–54, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 93

[69] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 18, 22, 23

[70] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in neural information processing systems (NeurIPS)*, 2015. 49, 52

[71] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. 2018. 52

[72] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016. 88

[73] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 325–334, 2017. 30

[74] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 30, 31

[75] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 70

[76] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 52

[77] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 27, 30, 31

[78] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 30

[79] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 30

[80] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 69, 70

[81] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 90

[82] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 88, 89

[83] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 47, 49, 52, 59

[84] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for multiple object tracking. *arXiv preprint arXiv:1703.09554*, 2017. 30

[85] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision (ACCV)*. Springer, 2018. 6, 87, 88, 89, 90, 91, 92, 96, 97, 98

[86] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 66

[87] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 30, 31

[88] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 17, 65, 66

[89] Dong Lao and Ganesh Sundaramoorthi. Extending Layered Models to 3D Motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 30, 31

[90] Issam H Laradji, David Vazquez, and Mark Schmidt. Where are the masks: Instance segmentation with image-level supervision. *arXiv preprint arXiv:1907.01430*, 2019. 52

[91] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 17

[92] William J. M. Levelt. *Speaking: From Intention to Articulation*. MIT Press, Cambridge, MA, 1989. 92, 93, 94

[93] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 30

[94] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 49, 52, 59

[95] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 31

[96] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 89

[97] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Zequn Jie, Jiashi Feng, Liang Lin, and Shuicheng Yan. Reversible recursive instance-level object segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 13, 16

[98] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 47, 52

[99] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 13

[100] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 50

[101] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, Cham, 2014. Springer International Publishing. 89

[102] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 88

[103] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 88

[104] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2017. 91

[105] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the IEEE conference European Conference on Computer Vision (ECCV)*. Springer, 2016. 13, 70

[106] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 14, 32

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 15

[108] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 101

[109] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2018. 70

[110] K Maninis, S Caelles, Y Chen, J Pont-Tuset, L Leal-Taixe, D Cremers, and L Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 27, 30, 31, 39

[111] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Deep retinal image understanding. In *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2016. 9

[112] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 50, 67

[113] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 89, 91

[114] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 2010. 50

[115] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013. 22

[116] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 91

[117] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 30

[118] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014. 30

[119] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 43

[120] Christopher Olah. Understanding lstm networks. 2015. 18

[121] Firat Ozdemir, Philipp Fuernstahl, and Orcun Goksel. Learn the new, keep the old: Extending pretrained models with new anatomy and images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2018. 70

[122] Firat Ozdemir, Zixuan Peng, Christine Tanner, Philipp Fuernstahl, and Orcun Goksel. Active learning for segmentation by optimizing content information for maximal entropy. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 183–191. Springer, 2018. 70

[123] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 50, 54

[124] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision (CVPR)*, 2015. 47, 49, 52

[125] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 18

[126] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015. 51

[127] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014. 51

[128] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 22, 95

[129] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 27, 30, 37

[130] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 27, 30

[131] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 87, 89

[132] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 22

[133] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 51

[134] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 12, 52

[135] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 88, 89

[136] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 3, 7, 27, 29, 30, 31, 34

[137] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI*, 2018. 23

[138] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 13, 70, 72

[139] Ehud Reiter and Robert Dale. A fast algorithm for the generation of referring expressions. In *COLING 1992 Volume 1: The 15th International Conference on Computational Linguistics*, 1992. 91

[140] Tal Remez, Jonathan Huang, and Matthew Brown. Learning to segment via cut-and-paste. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 52

[141] Mengye Ren and Richard S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[142] Mengye Ren and Richard S Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 32

[143] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 5, 12

[144] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 16

[145] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 32

[146] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 15

[147] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 32

[148] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 51

[149] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, 2004. 50, 52

[150] Soumya Roy, Asim Unmesh, and Vinay P Namboodiri. Deep active learning for object detection. In *British Machine Vision Conference*, pages 3–6, 2018. 70

[151] Paula Rubio-Fernández. How redundant are redundant color adjectives? an efficiency-based analysis of color overspecification. *Frontiers in Psychology*, 7:153, 2016. 93

[152] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3, 32

[153] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision*, pages 86–103. Springer, 2018. 51

[154] Amaia Salvador, Miriam Bellver, Victor Campos, Manel Baradad, Ferran Marques, Jordi Torres, and Xavier Giro-i Nieto. Recurrent neural networks for semantic instance segmentation. *Deep Vision Workshop of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8, 16, 27, 30, 31, 32, 35, 58, 60

[155] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 52

[156] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. 97, 102

[157] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. *arXiv preprint arXiv:2007.08270*, 2020. 43

[158] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 20, 69

[159] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 17, 88

[160] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 30, 31

[161] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014. 11

[162] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. 50

[163] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 47, 52

[164] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017. 30, 31

[165] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 27, 30

[166] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001. 70

[167] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 12

[168] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 22, 23, 43

[169] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5277–5286, 2019. 8

[170] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014. 70

[171] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 27, 30, 31, 37, 39

[172] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 23

[173] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 87, 89, 97, 98

[174] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. 2018. 94

[175] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 52

[176] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 49, 52, 53

[177] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. 101

[178] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 802–810, 2015. 19, 30, 32, 33, 60

[179] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7, 88, 89

[180] Jia Xu, Alexander G Schwing, and Raquel Urtasun. Learning to segment under various forms of weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 47, 52

[181] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 27, 30, 31, 42, 87, 89

[182] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 7, 27, 29, 31, 34, 36, 37, 42, 102

[183] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017. 70

[184] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 27, 30, 37, 38, 39, 42

[185] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 94

[186] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 88

[187] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 14, 15

[188] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 87, 89

[189] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016. 89, 95

[190] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 91

[191] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *Unpublished draft. Retrieved*, 3:319, 2019. 14

[192] Cha Zhang, John C Platt, and Paul A Viola. Multiple instance boosting for object detection. In *Advances in neural information processing systems*, pages 1417–1424, 2006. 20

[193] Chao Zhang, Weiming Li, Wanli Ouyang, Qiang Wang, Woo-Shik Kim, and Sunghoon Hong. Referring Expression Comprehension with Semantic Visual Relationship and Word Mapping. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1258–1266, Nice, France, 2019. Association for Computing Machinery. 94

[194] Tianyi Zhang, Guosheng Lin, Jianfei Cai, Tong Shen, Chunhua Shen, and Alex C Kot. Decoupled spatial neural attention for weakly supervised semantic segmentation. *arXiv preprint arXiv:1803.02563*, 2018. 49, 52

[195] Xiangyun Zhao, Shuang Liang, and Yichen Wei. Pseudo mask augmented object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 49, 52

[196] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 52

[197] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016. 50

[198] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 49, 52, 58, 64, 65

[199] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018. 20

[200] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 20, 52