Efficient Panoptic Segmentation for Dynamic Street Scene Understanding

Literature Survey and Progress Report

Youki Iijima
Supervisor: Dr Luping Zhou
Bachelor of Software Engineering (Hons)
The University of Sydney

May 28, 2020

Contents

1 Introduction **2**
1.1 Fine-grained Scene Parsing Tasks . . . 2
1.2 Panoptic Segmentation . . . 2
1.3 Delineation of Scene Understanding Tasks . . . 2
1.4 Transferability of Scene Parsing Technology . . . 4
1.5 Technical Challenges . . . 5
1.6 Summary of following Sections . . . 5

2 Image Segmentation **6**
2.1 Approaches . . . 6
2.2 Datasets . . . 6

3 Semantic Segmentation **7**
3.1 Approaches: . . . 8
3.2 Evaluation Metrics . . . 10

4 Instance Segmentation **11**
4.1 Approaches . . . 11
4.2 Evaluation Metrics . . . 13

5 Panoptic Segmentation **14**
5.1 Approaches . . . 14
5.2 Evaluation Metrics . . . 15

6 Progress Report **16**
6.1 Work Summary . . . 16
6.2 Revised Work Plan . . . 16
6.3 Gantt Timeline . . . 17

References **18**

1 Introduction

Street scene understanding is a popular category of computer vision tasks with broad applications in the domains of autonomous driving, autonomous robotic navigation, video surveillance and video editing. In computer systems, contextual understanding of a street scene is typically attained through the partitioning of 2D images into semantic categories using deep learning networks. However, significant strides have been made recently in adapting the task of street scene understanding to pre-recorded and real-time video sequences of urban environments, and understanding street scenes in 3 dimensions through the simultaenous estimation of object depth. The objective of this report is to provide a review of the literature addressing fine-grained scene understanding tasks in the paradigm of computer vision, as well as a detailed analysis of both foundational and state-of-the-art deep learning scene segmentation frameworks. Additionally, this document features a progress report that outlines the work that I have completed this semester, as well as my revised work plan for next semester, towards the synthesis of my thesis project on *Panoptic street scene segmentation*.

1.1 Fine-grained Scene Parsing Tasks

Scene understanding through fine-grained semantic categorisation is addressed differently by the tasks of *Semantic Segmentation*, *Instance Segmentation*, and *Panoptic segmentation*. The task of Semantic segmentation involves partitioning scenes into semantic object categories, achieved by allocating a class label to each pixel in an image, corresponding to the object it represents. The task of Instance Segmentation involves identifying and segmenting (at the pixel level) distinct instances of countable object instances. Panoptic segmentation combines Semantic and Instance segmentation with the unifying goal of assigning a class label to each pixel of an image and assigning an identifying instance number to each distinct instance of an object class. Notably, since these are *segmentation* tasks, they each require the classification of each individual pixel in an image in order to produce fine-grained segmentation maps.

1.2 Panoptic Segmentation

Out of these three tasks, Panoptic Segmentation frameworks deliver the most complex, global understanding of 2-dimensional scenes in terms of semantic delineations and identification of distinct object instances. The central motivation for the proposal of this task was to reconcile the objectives, methodologies and efforts directed by researchers towards the tasks of semantic and instance segmentation [1]. Typically, Semantic and Instance segmentation methodologies are distinctly characterised by divergent topologies [1, 2]. While Instance Segmentation networks commonly leverage object detection frameworks to generate object instance proposals [2], semantic segmentation networks generally adopt a fully convolutional encoder-decoder architecture [3] that performs per-pixel multi-class categorisation. Furthermore, while semantic segmentation frameworks are effective classifiers of amorphous regions of similar texture or material, instance segmenetation frameworks are effective identifiers of countable objects with distinct contours [1, 4, 5]. Ultimately, Panoptic Segmentation unifies the more myopic tasks of Semantic and Instance segmentation to produce a unified, global understanding of a scene.

1.3 Delineation of Scene Understanding Tasks

There exist numerous categories computer vision tasks with the objective of parsing or understanding a scene in some way. Historically, the paradigm of computer vision research has been focused on the detection, identification and tracking of object instances [1, 6]. These tasks are commonly addressed by Object Detection frameworks for images and Object Tracking frameworks for videos. The primary difference between these frameworks and scene segmentation frameworks is the granularity of the predictions. Object Detection and Tracking frameworks output coarse-grained bounding box localisations for target objects, while segmentation newtorks output precise segmentation masks generated by classifying each pixel.

Video Object Segmentation is another computer vision task with the similar goal of generating a fine-grained object mask computed at the pixel level. However, unlike scene parsing tasks, the purpose of VOS is to output a *binary* mask for each frame in a video, which clearly differentiates between a single target object and the background of the video [7]. Comparatively, the objective of the scene parsing tasks that are the

focus of this project, are to segment multiple object classes and instances to deliver a holistic understanding of a scene and its compositional elements.

Object Detection The goal of this class of computer vision tasks is to produce coarse-grained, bounding box localisations of distinct object instances belonging to a pre-defined set of object categories. Both object detection and tracking implicitly involve object recognition and classification [8]. Object detection frameworks are typically based on deep learning approaches [2, 8]. However, for lightweight applications or for small datasets, sufficient accuracy can be achieved through machine learning or more simplistic approaches, such as blob analysis, feature extraction and other statistical approaches. Unlike Deep Learning approaches, which automatically learn object features, Machine Learning methodologies require the manual definition of feature labels.

Comparatively, Deep Learning approaches automatically learn object features without manual definition and are better suited to large datasets, and more comprehensive and accurate object detection and classification [2, 8]. Deep Learning object detection methodologies are commonly based on Convolutional Neural Network (CNN) architectures, which are highly effective in extracting high-level and hierarchical semantic image features [2]. CNN-based deep learning approaches have become ubiquitous in computer vision applications and serve as the foundation for numerous state-of-the-art Video Object Segmentation [7] and Image Segmentation frameworks [1–3, 9, 10]. Furthermore, Deep Learning-based Object Detection architectures are directly implemented in numerous state-of-the-art Instance and Panoptic Segmentaion architectures, which commonly rely on object detection to perform segmentation. [1, 2, 11].

Object Detection methodologies can generally divided into two categories: Two-Stage and One-Stage approaches. Popular two-Stage approaches include R-CNN [12], Fast R-CNN [13], Faster R-CNN [8] and Mask R-CNN [2]. These are region-based object detectors that first generate proposals for image regions that demonstrate objects, and then classify and produce a bounding box for each detected object. Recently, frameworks like [14], [15] and [16] improve accuracy by building upon the approaches of the R-CNN variants. Hu *et al.* [14] extend Faster-RCNN with an adapated attention module for object detection that considers the intraction between object features and geometry in an image. Zhu *et al.* [15] propose a deformable convolutional network, DCNv2, which benefit from improved adaptability to geometric variations of objects [17]. Ghiasi *et al.* [16] propose NAS-FPN, which uses a Neural Architecture Search (NAS) mechanism to learn scalable features from a stacked Feature Pyramid Network (FPN) architecture [16, 18].

Comparatively, One-Stage approaches generate classifications and bounding boxes directly from the input image. State-of-the-art One-Stage frameworks include YOLOv3 [19], SSD [20], deconvolutional ssd (DSSD) [21], RetinaNet [22], RefineDet [23] and M2Det [24]. While Two-Stage frameworks are more accurate, One-Stage architectures are more efficient and suitable for real-time applications.

Common *backbone* networks for feature extraction in modern object detection frameworks include densely connected backbones such as ResNet [25], ResNeXt [26] and AmoebaNet [16]; or lightweight backbones such as MobileNet [27, 28], ShuffleNet [29], SqueezeNet [30], Xception [31], EfficientNet [32], PeeleNet [33] and [34]. While deep and densely connected networks produce higher accuracy, the lightweight networks produce efficient, real-time predictions.

Multiple Object Tracking The purpose of Multiple Object Tracking (MOT) frameworks are, intuitively, to identify multiple objects in a scene with a localising bounding box, and to track these objects continuously throughout a video sequence. MOT extends the tasks of object detection and object tracking with the objective to produce temporally continuous and consistent object instance indentifications. Notably, MOT requires consistent identity tracking to ensure the continuity of bounding box localisations [35–37]. MOT algorithms typically consist of 4 stages of: Detection, Feature extraction and motion prediction, Affinity calculation, and Association [36]. The first two stages resemble the object detection pipeline and are commonly addressed with a convolutional architecture [36]. Motion prediction algorithms are also commonly used in the second stage to predict the motion of the detected objects and inform the classifications of subsequent frames [36]. In the third stage, affinities between target objects are calculated by a distance or similarity score. Approaches to this third stage commonly employ a Long Short-Term Memory (LSTM) [38] network. In the fourth stage, Recurrent Neural Networks (RNNs) [39] are commonly adopted to associate detected features that belong to the same target. LSTMs and RNNs are both deep learning networks that are ideal

for processing sequential data [36]. MOT is extended to the task of Video Instance Segmentation by Voigtlaender *et al.* [37] and Yang *et al.* [35], who propose the task of simultaneously tracking and segmenting multiple object instances in a video sequence.

Video Object Segmentation Video Object Segmentation (VOS) is a fine-grained binary partitioning problem with the aim of precisely segmenting an object of interest from the background of a continuous video sequence by assigning a binary label to each pixel in each video frame [40]. Structurally, VOS systems are founded on deep learning models that are optimised with heuristic algorithms or human interaction [7]. Examples of commonly employed heuristic algorithms include object trajectory and optical flow algorithms [7]. Examples of human interaction include video annotation and neural network training. Principally, the three main categories of VOS frameworks are Unsupervised, Semi-Supervised and Supervised VOS – which denote the level of human interaction that is required to perform object segmentation on a video sequence. Typically, modern approaches to VOS are based on CNN architectures [41]. Supervised (or Interactive) approaches require the iterative involvement of a human annotator throughout the duration of a video sequence to continuously correct the segmentation algorithm. While Supervised frameworks produce the highest segmentation accuracy, they are only suited to a limited set of practical applications such as video editing [42]. For the task of Semi-Supervised VOS, given a set of annotated key frames, or an annotated intial video frame, a VOS network must segment the annotated object in the remaining frames [41]. Approaches to this problem are typically divided into the categories of Motion-based methods and Detection-based methods. Motion based methods, commonly inspired by MaskTrack [43], rely on the propagation of segmentation masks from previous frames that are refined with a convolutional neural network. Detection-based methods such as OSVOS [44] perform object segmentation sequentially on each video frame without consideration of temporal information [41]. Unsupervised approaches are the least accurate of the three tasks, but perform VOS automatically without the provision of a ground truth mask or a set of annotated key frames. There are a variety of approaches to this task including those that exploit salient object detection, semantic parsing, optical flow computations, motion information or object proposals to accurately and continuously segment the target object [41].

1.4 Transferability of Scene Parsing Technology

The development of highly accurate and efficient scene understanding networks have a high potential for transferability to other computer vision tasks. For example, state-of-the-art street scene understanding systems such as [45], [46], and [5] tested on urban landscape datasets [10, 47] have also proven to be highly effective with challenging general purpose datasets like [48] and [49]. Furthermore, methodologies developed for the task of scene parsing like [50] are highly influential in the development of medical imaging applications [51]. Similarly, semantic classification networks developed for medical imaging such as [52], [53] have been highly influential for methodologies focused on street scene understanding. More recently, contour-based approaches to semantic and instance segmentation (as opposed to pixel-based approaches) that are popular in medical imaging frameworks [54], have been proven to be highly effective for scene parsing tasks [46].

Moreover, due to the universality of Deep Learning architectures in modern approaches to Computer Vision tasks [7, 10] including Object Detection [2, 8], Scene Segmentation [2, 5, 45, 46], Video Object Segmentation [7] and Tracking [37], advancements of existing theory and the development of novel approaches to scene parsing tasks also have the potential to impact several computer vision paradigms simultaneously. Within the scope of scene parsing tasks, techniques developed separately for semantic and instance segmentation networks, such as multi-scale feature networks, pyramid-style networks [55, 56], active contour models [46, 57] and fully convolutional encoders [11, 50, 58, 59] have a high degree of transferability, and improve segmentation performance by incorporating both global contextual information and instance boundary information, that are typically addressed myopically by the two different tasks. Similarly, object detection frameworks and instance-based segmentation frameworks are also advanced symbiotically. Object detection methodologies such as [8, 19], structures like Feature Pyramid Networks (FPNs) and backbone architectures like ResNet [25], VGG16 [60], VoVNet [34], Xception [31] and EfficientNet [32] commonly inform the development of fine-grained segmentation topologies, or directly serve as the foundation of these frameworks [1–5, 46]. The adaptation of these object detection structures to fine-grained scene parsing tasks can lead to the advancement of existing technology through improved efficiency [5] and localisation accuracy [2], as

well as the advancement of the theoretical understanding of object detection frameworks, as coarse-grained detection and classification systems are adapted to the more complex task of fine-grained understanding [2, 3]. Notably, in the landmark instance segmentation paper [2], He *et al.* propose Mask R-CNN as an extension to Faster R-CNN [8], which is not only a more accurate object detector [2, 18] than its predecessor, but is also fine-tuned to the task of pixel-wise segmentation of object instances.

1.5 Technical Challenges

Hardware Limitations and Efficiency: Developments towards *real-time* street scene understanding (without the significant sacrifice of accuracy) will likely have the greatest applicability to a wide variety of computer vision tasks, including autonomous navigation for vehicles, as well as tracking and understanding human activity for the purpose of surveillance [5]. Thus, while pixel-level accuracy is a fundamental aim of scene segmentation tasks, one of the major challenges of these tasks, in terms of maximising its practical applicability, is to achieve an acceptable compromise between processing speed and accuracy in a minimally supervised system. Ultimately, different technologies employing Segmentation frameworks have different requirements regarding the acceptable levels of performance and accuracy [61].

Quality of Data: High quality Image Segmentation datasets require massive amounts of work to produce since they require detailed annotations, and pixel-perfect ground-truth masks defined for each high-resolution frame in a video sequence. For example, annotating a single 1024×2048 image or video frame in the Cityscapes dataset takes approximately 1.5 hours [10]. Additionally, issues caused by the quality of video capture in test and training data also complicates computer vision tasks; usually through the disruption of object detection mechanisms, or the continuity of models that rely on temporal data. Some common data capture issues include: object and scene deformation, motion blur, object scale variation, and low-resolution image quality [10, 40]. Ultimately, due to the difficulty to produce and annotate high-resolution Image Segmentation datasets, there are few high-quality options to choose from. This increases the likelihood of a Segmentation framework being overfitted to certain datasets. Furthermore, this limits the adaptability of a Scene Segmentation framework to real-life scenarios and novel datasets.

Interestingly, this limitation can be addressed by synthesising training samples based on the spatiotemporal features of video sequences. For example, Zhu *et al.* [62] proposed a novel deep learning framework based on mask propagation and label relaxation techniques that leverages temporal information extracted from video sequences to synthesise new training samples from a sparsely-labelled dataset. By augmenting existing datasets, and synthesising more training samples, they achieved state-of-the-art Semantic Segmentation accuracy on the Cityscapes Street Scene Segmentation benchmark dataset [10], using the Deeplab Atrous Convolutional network [50] as a baseline [62].

Nature of Semantic Information and Instance Representations: Scene parsing tasks are complicated by objects that are difficult to differentiate semantically and objects with ambiguous boundaries. For example, the presence of heterogeneous objects, the ambiguity of object contours, and objects with complex shapes are not easily interpreted by deep learning frameworks [40], and cause confusion on both the semantic and instance level. Furthermore, natural challenges resulting from the dynamic nature of captured subjects such as object distance, object movement on the z-axis (in video parsing tasks), and object occlusion, also increase the complexity of computer vision tasks by increasing the difficulty of object detection [10, 63, 64]. Datasets such as Mapillary Vistas [47] cultivate this complexity, and intentionally seek to maximise the variety of scene conditions, including highly varied lighting and weather conditions and street scene contexts captured in numerous locations globally, in order to augment the complexity and challenge of scene parsing tasks [47, 64].

1.6 Summary of following Sections

The remainder of this survey will feature a summative discussion of the various Computer Vision Tasks related to fine-grained Scene Understanding, including popular approaches, architectures, datasets and evaluation methods for each. It is organised as follows: Section 2 is a summative review of the paradigm of Image Segmentation tasks, including an overview of its key objectives and the datasets used to test image

segmentation framewokrs. Sections 3 to 5 are an overview of Semantic, Instance and Panoptic Segmentation in order. Each section features an overview of the foundational architectures and approaches to each task and the metrics used to evaluate the performance of a framework on each task.

2 Image Segmentation

2.1 Approaches

Image Segmentation describes a family of computer vision tasks that includes *Semantic*, *Instance* and *Panoptic* Segmentation, in order of increasing computational complexity. The common purpose of these tasks is to partition a scene (represented by an image or video frame) into multiple segments corresponding to distinct object classes, and (in the case of Instance and Panoptic Segmentation) distinct object instances. Unlike Object Detection frameworks, which output coarse-grained bounding-box localisations for detected objects, Image Segmentation requires pixel-level classification and instance identification, to generate fine-grained segmentation maps featuring precise object masks. As with Object Detection frameworks, Deep Learning-based Convolutional models are preferred universally to Machine Learning or statistical methods for their unparallelled performance and accuracy [54].

Fine-grained segmentation simulates complex understanding of scenes that feature numerous identifiable objects (typical of street scenes) by clearly distinguishing between image regions that represent different object categories and object instances. Fine-grained segmentation and identification at the semantic level more closely resembles human scene understanding than coarse-grained localisation produced by Object Detection [1, 6]. Additionally, complex scene understanding can be further augmented through the integration of other computer vision solutions such as depth estimation methods: for monocular images [65], or in combination with multiple sensors or cameras to generate 3-dimensional segmentation maps that further approach human scene understanding.

The following sections will survey the computer vision tasks of Semantic, Instance and Panoptic Segmentation, including an overview of each task's most popular and effective architectures, and a summary of the State-of-the-Art approaches to these tasks. Additionally, for each task, there will be an overview of the methods and metrics used to evaluate the accuracy and performance of the approaches to each task.

Since the focus of this project is on Street-Scene Segmentation, the effectiveness of approaches will be evaluated based on their performance on the 'Cityscapes' Benchmark street scene dataset[10]. The reason for this dataset choice will be explained in greater detail in the following section. Additionally, there will be an overview of alternative datasets and training datasets commonly used for Street-Scene Segmentation.

2.2 Datasets

Scene Parsing datasets can be grouped into 4 categories: 2D image, RGB-D (colour images with depth information) [66], 3D image datasets and Video datasets. Each viable dataset features dense, pixel-wise labels on their images that are used to evaluate performance [54]. However, since the focus of this project is on 2D Street Scene segmentation, 2D datasets focused on urban and street scene understanding will be preferred exclusively for network training and evaluation in the implementation of this project. Furthermore, only large-scale and high-resolution datasets will be considered for the implementation phase of this project. High-resolution and large-scale datasets are necessary to both evaluate and improve the performance of a proposed architecture in the task of fine-grained segmentation [62].

2D Datasets Some popular 2D image segmentation datasets include the PASCAL datasets: PASCAL Visual Object Clases (VOC), which consists of approximately 1500 training and validation images featuring 21 object classes; and PASCAL Context [67], which extends PASCAL VOC with over 400 class labels grouped into 3 coarse-grained class categories (objects, stuff, and hybrids). Here, *Objects* refer to countable object instances, *stuff* refers to amorphous objects with undefined contours, and *hybrids* refer to classifiable objects that fall into either category [6, 49]. Another popular large-scale dataset is Microsoft Common Objects in Context (MS COCO) [48], which features *common* objects in 'complex everyday scenes'. This is an extensive dataset with 328k images, and 2.5 million labeled instances grouped into 91 object classes [48]. The common theme between these datasets is the focus on the general applicability of a segmentation or object detection

network to a wide variety of 'common' objects and scenes. Similar 2D datasets include ADE20K /MIT Scene Parsing (SceneParse150) [68], Stanford background [69], Sift Flow [70], Berkeley Segmentation Dataset (BSD) [71] and Youtube-Objects [72].

Despite the popularity of these datasets, they will not be considered for the implementation phase of this project, since they are not focused on urban and street scene understanding. Comparatively, datasets such as [10], [73], and [47] are ideal for the purpose of this project, since they are large-scale, high-resolution datasets centered around dynamic street [10, 47], and road (driving) [73] scenes; including dense semantic annotations for a variety of objects commonly present in urban and road environments.

Cityscapes Cityscapes [10] is a prominent dataset for image and video segmentation (for Semantic, Instance and Panoptic tasks) focused on urban street scene understanding applications [11, 45, 54, 63]. The dataset consists of a diverse set of stereo video sequences capturing street scenes from 50 different cities. For training and validation, the dataset includes 18,000 and 2,000 weakly annotated frames respectively. For testing, the dataset includes 5,000 high quality video frames, annotated finely at the pixel level. Cityscapes features 30 classes, which are grouped in to 8 categories: flat, human, vehicle, construction, object, nature, sky and void. As the premier dataset for street scene understanding, segmentation accuracy produced on Cityscapes will be considered in this project as the primary indicator of a segmentation framework's performance, relative to its peers.

Mapillary Vistas Mapillary Vistas [47] is a challenging, large-scale street scene dataset that includes pixel-accurate, instance-specific (panoptic) annotations of 25,000 high-resolution images captured by both driving cars and pedestrians [47, 63, 64]. The dataset is divided into 18K images for training, 2K images for validation, and 5K images for testing. Vistas is challenging due to the significant variability in the captured images, including variations in the weather, time of day, lighting and sensor modality [47, 64]. The non-commercial Vistas dataset includes semantic annotations of 66 object categories, and instance-specific annotations for 37 of those categories [47]. Since Mapillary does not publish a leaderboard for Image Segmentation like Cityscapes, it is difficult to evaluate state-of-the-art networks relative to their performance on the Vistas dataset. Nevertheless, as a challenging and extensive dataset focused on street scene understanding, Vistas, along with Cityscapes, is an ideal candidate for the training and evaluation of this project's practical implementation of a scene parsing network.

KITTI: KITTI is a popular dataset for autonomous driving and robotics applications focused on 3D object detection and tracking, optical flow, and visual odometry applciations [73]. The dataset includes videos of road and traffic scenes captured in stereo with 2 colour and grayscale cameras. The KITTI dataset for semantic and instance segmentation is small, with only 200 training and 200 test images that are annotated finely with instance-specific semantic labels. However, traffic scene understanding is a highly relevant field of computer vision, and this dataset should be considered for the implementation phase of this project for its high quality road scene video sequences.

It is also worth noting that KITTI recently introduced an annotated dataset for the novel task of *Multiple Object Segmentation and Tracking* [37]. This task combines the tasks of Video Object Segmentation and Multiple Object Tracking, with the objective of tracking multiple objects in video sequences, and producing a fine-grained, pixel-level segmentation of each object in each video frame [37]. This task is functionally equivalent to a temporally continuous application of Instance Segmentation on a video sequence, where object instances are continuously tracked. A conceptually similar dataset, Youtube VIS, was proposed in [35] for the novel task of Video Instance Segmentation. Naturally, this task resembles Instance Segmentation, but emphasises temporal continuity and instance tracking in its performance evaluation [35]. While the objectives associated with these datasets are outside the scope of this project, they represent interesting, burgeoning areas of research in the field of computer vision.

3 Semantic Segmentation

The goal of Semantic Segmentation is to predict and assign a semantic class label to individual pixels in an image, in order to distinguish between different types of objects in a scene with fine (pixel-level)

granularity. The task involves partitioning an image into semantic categories through dense, per-pixel class predictions [10, 62] and producing a segmentation map that is the same size of the input, which demonstrates the boundaries between different semantic object categories in a scene. Unlike Instance Segmentation or Panoptic Segmentation, Semantic Segmentation is not concerned with identifying unique object *instances*, or with delineating the boundaries between separate instances of the same object class.

For example, in a street scene with two pedestrians, all the pixels that represent either pedestrian would be assigned a class label like "pedestrian" by a Semantic Segmentation network. No distinction is made between the two separate pedestrians. Furthermore, these pixels with the "pedestrian" label constitute a segmentation *channel* or *mask* corresponding to the semantic category with the title of "pedestrian". The "pedestrian" mask has the same spatial dimensions as the input image. Pixels that belong to pedestrians in the input image are represented in the mask with a unique identifier (like an integer). Other pixels are assigned a null value identifier in the "pedestrian" mask or channel. The output of a Semantic Segmentation sysytem, a segmentation map, is made up of multiple *masks* representing distinct semantic categories that collectively represent and partition the entire input image. In a street scene, some common categories are cars, street lights, buildings, roads, and pavements.

3.1 Approaches:

Having surveyed hundreds of papers, Minaee *et al.* [54] propose several foundational categories of deep learning approaches to Semantic Segmentation, grouped according to their main technical contributions: Fully convolutional networks (FCN) [3], Convolutional models with graphical models (e.g. Conditional Random Fields [74] and Markov Random Fields [75]), Encoder-decoder based models [3, 45, 52, 53, 76, 77], Multi-scale and pyramid network based models [78, 79], Dilated (Atrous) convolutional models [9, 80]. The survey also includes frameworks based on Recurrent Neural Networks (RNNs) [39] and General Adversarial Networks (GANs) [81]. However, neither of these architectures are featured amongst the state-of-the-art approaches tested on Street Scene Segmentation datasets; and Convolutional approaches are typically preferred for their segmentation performance in the current paradigm of Scene parsing tasks.

Fully Convolutional Networks: Conceptually, Semantic Segmentation frameworks are typically derived from Fully Convolutional Networks (FCN) [2, 82]. FCNs resemble Convolutional Neural Networks (CNNs), except that they do not include any fully-connected layers - i.e. all of the network's layers are convolutional [3]. While CNNs use fully-connected layers to perform classification, FCNs perform pixel-wise classification using convolutional layers and a cross-entropy loss function to predict the class probability of each pixel. FCNs are effective, and improve upon earlier, CNN-based approaches because they can accept an arbitrarily-sized input image and produce a segmentation map of the same size by performing pixel-wise classification [3]. Long *et al.* demonstrated this in an influential paper [3], where they adapted the standard classification networks of the time: AlexNet, VGG net and GoogLeNet to the task of Semantic Segmentation by substituting fully-connected layers with 1x1 convolutional layers (to perform pixel-wise classification). Specifically, after encoding the input via a series of convolutional layers, the low-resolution output of the network is upsampled to recover the dimensions of the input image. Additionally, skip connections between lower-level coarse layers and higher-level fine-grained layers are used to inform local predictions with global structural information [3].

Convolutional Networks with CRFs and MRFs: The FCN model proposed in [3] has several significant limitations such as its inference speed, its transferability to 3D images and its capacity to efficiently account for a scene's global contextual information [82]. One alternative to the FCN model is a CNN-based model that integrates probabilistic statisical models like Conditional Random Fields (CRFs) [74] and Markov Random fields (MRFs) [83]. The motivation of this category of approaches is to use probabilistic models to address the poorly localised predictions produced by deep CNNs, through post-processing [82]. Notably, Deeplab and Deeplab v2 [9] achieved state-of-the-art performance in 2014 and 2016 on the PASCAL VOC dataset [49] using an atrous convolutional network and a fully-connected CRF in post-processing to improve segmentation performance. However, the newer iterations of Deeplab [11, 50] no longer use CRFs, since post-processing modules increase inference time, and state-of-the-art accuracy was proven to be achievable without it [11].

Encoder-Decoder Architectures Modern state-of-the-art approaches to Street Scene Segmentation commonly adopt a fully convolutional Encoder-Decoder architecture. These approaches feature a convolutional encoder module similar to [3], or an atrous convolutional encoder similar to [9]. Additionally, they feature a *decoder* module that recovers high-resolution image representations by upsampling the low-resolution output producecd by the encoder [3, 54]. In these architecures, the encoder module is responsible for capturing contextual and semantic information, while the decoder module is responsible for recovering the spatial resolution of the input image, and the precise, pixel-wise localisation of semantic features onto the input image [3, 52, 54]. For example, in [3], Long *et al.* proposed a decoder module based on upsampling via *deconvolution* (or *backwards convolution*) that effectively reverses the forwards and backwards passes of convolution. With this decoder module, the low-resolution output produced by the convolutional encoder is upsampled using a fractional stride and spatial information is recovered through bilinear interpolation between the coarse output of shallow convolutional layers and the upsampled output [3].

In another influential paper, Ronneberger *et al.* proposed U-Net, which features a symmetrical, U-shaped Encoder-Decoder architecture, consisting of a contracting path (the encoder) and an expanding path (decoder). The contracting path follows the architecture of a convolutional network similar to [3]. It consists of several sequential blocks, which apply two 3x3 convolutional layers, followed by a 2x2 max pooling operation to downsample the input, and double the number of feature channels at each step [52]. The expanding path mirrors the contracting path. Each block in the decoder takes an input, applies two 3x3 convolutional layers, and upsamples the input with a 2x2 deconvolution (or up-convolution) operation that halves the number of feature maps, and recovers the spatial resolution of the original image at each layer. Additionally, at each expanding block, the feature map produced by the corresponding contracting block is copied and concatenated with the input, in order to recover localisation and pattern information at each stage in the image reconstruction process. Finally, a 1x1 convolutional layer is applied to the output of the decoder module to generate a segmentation map throuh pixel-wise classification. Other influential, symmetrical Encoder-Decoder architectures include DeConvNet [84] and Segnet [76], which extended the 16-layer VGG16 [60] Object Detection framework [60] with a symmetrical deconvolutional decoder module; and V-Net [53], which adapts a U-Net-like architecture to the task of 3D Semantic Segmentation.

Notably, Wang *et al.* proposed a state-of-the-art Encoder Decoder architecture, HRNet (High Resolution Network) [45], that maintains a high-resolution representation of the input image throughout the entire encoding process. Specifically, the encoder network maintains a high resolution information stream that is connected in parallel to the lower-resolution streams that are generated at each stage in the downsampling process. Information is repeatedly exchanged across the parallel resolution paths. Comparatively, typical encoder-decoder frameworks similar to [52], [76] and [84], encode the input image as a low-resolution representation first with a series of high-to-low resolution convolutions, and then recover a high-resolution representation with a decoder network. However, the downsampling process in these architectures results in a loss of spatial and global contextual information [3]. While this is addressed with skip connections in [3], and symmetrical path connections in [76] and [52], the parallel information exchange between resolution streams in HRNet results in much improved spatial precision and semantic parsing [45, 85]. Currently, HRNet is commonly incorporated as the backbone of recently proposed semantic segmentation frameworks, which are typically augmented with multi-scale contextual models and self-attention mechanisms [54, 85]. Notably, Yuan *et al.* [85] proposed the "HRNet+OCR" method, which exploits object-contextual representation in a context aggregation framework to improve the segmentation accuracy of the HRNet predictions. This combined system achieves leading, state-of-the-art performance on several benchmarks including [10] and [48].

Multi-Scale and Pyramid Models: Pyramid networks exploit multi-scale features of an image by merging high and low resolution features extracted from the inherent hierachical structure of deep CNNs [54]. Lin *et al.* proposed a Feature Pyramid Network (FPN) [86], which consists of a bottom-up pathway to compute a feature hierarchy of multi-scale feature maps and a top-down pathway that upsamples spatially coarse, but semantically rich low-resolution features. Lateral connections in the pyramid merges feature maps of the same spatial size [86] to share spatial and contextual information. While the authors initially proposed this framework for object detection, it is commonly employed in Scene segmentation frameworks.

Zhao *et al.* proposed a Pyramid Scene Parsing Network (PSPNet) [87] for semantic segmentation, which exploits global context information derived from different-region-based context aggregation that is faciliated

by a pyramid pooling module. In this network a residual network (ResNet) backbone is used to extract features in a dilated network [54, 87]. Extracted features are pooled and fused in a four-tiered pyramid at four resolution scales [87]. These pooled features are upsampled and concatenated to produce a feature representation that incorporates both local and global context information, which is then fed into a 1x1 convolutional layer to generate per-pixel semantic class predictions [87]. Building on PSPNet, Li *et al.* [80] propose a state-of-the-art approach, GALD-net, which models long-range dependencies in FCNs via a global aggregation module that are accurately localised by a Local Distribution module.

Other approaches involving multi-scale feature analysis include [88] which involves a Laplacian pyramid with skip connections and multiplicative gating, DM-Net (Dynamic Multi-scale Filters Network) [89], Context contrasted network and gated multiscale aggregation (CCN) [90], Adaptive Pyramid Context Network (APC-Net) [91], Multi-scale context intertwining (MSCI) [92], and instance-level salient object segmentation [93].

Atrous Spatial Pyramid Pooling: Atrous Spatial Pyramid Pooling (ASPP) architectures are based on Dilated (or "atrous") convolution, which is a process that increases the receptive field of a convolutional kernel [9] according to a specified dilation rate. Dilated convolution addresses the decreasing resolution of an image representation in a convonlutional network caused by pooling and striding [54]. ASPP samples a convonlutional feature layer using multiple sampling rates and fuses these feature representations to capture image context at multiple scales [9]. The Deeplab family of approaches proposed by Chen *et al.*: Deeplabv1 [9], Deeplabv2, Deeplabv3 [50] and Panoptic-Deeplab [11] are highly influential methodologies that combine dilated convolution and ASPP. Deeplabv1 [9] and Deeplabv2 use a CRF [74] in a final post-processing stage to improve localisation accuracy, which is typically compromised by downsampling in a convolutional network [9]. Deeplabv3 [50] employs atrous convolution in cascade (or parallel) to capture multi-scale context by adopting multiple atrous rates of convolution and combining these cascaded feature representations [50]. Additionally, in [50], Chen *et al.* augment the ASPP module from Deeplabv2 [9] with image-level features to encode global context into the multi-scale feature represetnations [50]. With Deeplabv3+ [94], Chen *et al.* extend Deeplabv3 with a decoder module to refine segmentation results, and use a modified Xception [31] feature extraction backbone to perform efficient feature extraction. Currently, Deeplabv3 and Deeplabv3+ are commonly employed in state-of-the-art Semantic Segmentation frameworks such as [62] and [95], as well as Panoptic Segmentation frameworks such as [58] and [11]. Other architectures that adopt atrous convolution include ENet [96] and DenseASPP [97].

3.2 Evaluation Metrics

The Jaccard index, is the standard metric for evaluating Semantic Segmentation accuracy. Also known as the Jaccard Similarity coefficient, this metric measures the similarity between the segmentation mask predicted by the segementation network and the *ground truth* mask, which represents the pixel-precise expected output. More generally, this metric represents the intersection-over-union, or the *overlap* between two sets. Mathematically, the Jaccard index is defined as follows: For two sets A and B: $J(A,B) = |A \cap B|/|A \cup B|$. $J(A,B) = 1$ if A is equal to B and $J(A,B) = 0$ if they are disjoint. In [49], an equivalent metric known as the PASCAL VOC intersection-over-union is established for the task of Semantic Segmentation: IoU $= \frac{TP}{TP+FP+FN}$. IoU computes the intersection-over-union between two sets of pixels: the predicted segmentation mask generated by the Semantic Segmentation network, and the ground truth mask that represents the expected output of the network: a precise, pixel-wise segmentation of objects belonging to a particular class. In the IoU equation, TP, FP and FN denote the number of true positive, false positive and false negative pixels, respectively, in the predicted segmentation mask compared to the ground truth mask, computed across the entire test set [10, 49].

Cordts *et al.* [10] proposed 4 augmented metrics based on the PASCAL VOC IoU to evaluate the accuracy of systems tested on the Cityscapes Dataset. Firstly, IoU is calculated separately for different classes and different *categories* of classes: $\text{IoU}_{\text{class}}$ and $\text{IoU}_{\text{category}}$. The purpose of delineating between class IoU and category IoU is to account for two semantic *granularities* of scene understanding [10]. For example, consider four semantic *classes* common to street scenes: "car", "truck", "bus" and "train". In a semantic sense, these classes could belong to the same *category* of "vehicle." Intuitively, differentiating between categories is a less complex task than differentiating between classes belonging to the same category [10].

Additionally, Cordts *et al.* introduces a second metric, iIoU, to evaluate the intersection-over-union of the predicted and ground-truth masks at the *instance level*. This metric is denoted as: $\frac{iTP}{iTP+FP+iFN}$. iIoU is similarly separated according to the two semantic granularities: $\text{iIoU}_{\text{class}}$ and $\text{iIoU}_{\text{category}}$. The motivation for this metric is to resolve the bias of IoU towards instances of objects that cover a large image area, owing to the metric's global scope [10]. Thus, in the iIoU equation, iTP and iFN represent *weighted* counts of true positive and false negative pixels respectively in the predicted mask [10]. The weight of each pixel is adjusted by the ratio of the average instance size of the pixel's class to the size of the pixel's corresponding ground truth instance [10]. Notably, the number of False Positives (FP) are not normalised in the iIoU equation, since these pixels are not associated with any object instance in the ground truth segmentation map [10].

4 Instance Segmentation

The objective of Instance Segmentation (IS) is to identify all *distinct countable* objects in a scene and to produce a precise segmentation mask for each detected object [1, 2, 46]. Conceptually, IS combines the computer vision tasks of Object Detection and Semantic Segmentation [2], since it not only involves the classification of individual pixels, but also the higher-level differentiation between object instances. Additionally, IS also requires the delineation of object instance boundaries through the assignment of an object instance identifier to individual pixels, in addition to a semantic label [46].

Notably, unlike Semantic Segmentation, Instance Segmentation is not concerned with labelling amorphous background objects with undefined boundaries such as the sky and the ground. Furthermore, it is focused on distinguishing between different object instances, whereas Semantic Segmenation does not treat instances as separate entities. The key difference between these tasks is that Instance Segmentation is only focused on identifying and segmenting *countable* objects in a scene. An Instance Segmentation map is made up of image masks corresponding only to object instances; and each pixel in these masks are assigned a single class label and an integer that identifies a distinct instance of an object category.

In a street scene, Instance Segmentation would involve identifying and distinguishing between distinct instances of critical objects of interest such as cars and pedestrians. Instance segmentation can also facilitate more complex scene understanding such as fine-grained (pixel-accurate) multiple object tracking [37]. Moreover, Instance Segmentation maps can be used to inform intelligent monitoring and identification of objects of interest with dynamic, suspicious or unpredictable movement such as cars and pedestrians in autonomous vehicles or surveillance systems.

4.1 Approaches

Instance Segmentation methods can be generally divided into three categories: *Proposal-based*, *Proposal-Free* and *Contour-based* appoaches [46].

Proposal-Based: Instance Segmentation approaches commonly employ Propsal-based (or *Detection-based*) methods based on Deep Learning Object Detection frameworks [1, 2]. Detection-based methods can also be divided into the categories of *Two-Stage* and *One-stage* architectures, consistent with the existing classes of Object Detection methodologies. Two-Stage architectures typically adopt variants of Faster R-CNN [8] and (most commonly) Mask R-CNN [2]. One-Stage architectures typically adopt single-shot frameworks like YOLOv3 [19]. While Two-Stage networks generally produce more accurate results [2, 46, 98], One-Stage architectures are considerably faster, and approach real-time segmentation performance [98–100]. Most modern approaches to Instance Segmentation feature a proposal-based, two-stage pipeline, which first identifies a set of region proposals for object instances, and then determines which proposals to segment through a voting process [2, 46]. Early approaches to this task, such as those developed by Dai *et al.* [101, 102], proposed a multi-stage detection-based pipeline where segmentation follows bounding box detection [2, 101]. Alternatively, in frameworks like [103], [104] and [102] segmentation precedes recognition [2].

Building on these works, He *et al.* proposed a landmark Instance Segmentation framework known as Mask R-CNN [2], which significantly improved upon the efficiency of earlier approaches, by performing object detection and mask prediction simultaneously in a parallel pipeline. Mask R-CNN extends the two-stage object detection framework, Faster R-CNN [8], with an additional instance mask prediction branch,

computed in parallel with the original framework's bounding box localisation branch. Furthermore, Mask R-CNN adapts the baseline framework to the task of Instance Segmentation through the introduction of an innovative Region of Interest (RoI) feature mapping module called *RoIAlign* [2]; designed to replace the *RoIPool* module in [8]. He *et al.* assert that *RoIPool*, which was designed to output coarse-grained bounding box localisations for object detection, is not suitable for fine-grained, pixelwise segmentation. *RoIPool* performs spatial quantisation for feature extraction [8], which leads to poor localisation of semantic labels and misaligned segmentation masks [2]. Comparatively, *RoIAlign* in [2] is a quantisation-free layer that preserves exact spatial locations and significantly improves segmentation accuracy over *RoIPool* [2]. Interestingly, Proposal-based detection frameworks like Mask R-CNN and its variants highlight a fundamental difference between the state-of-the-art approaches to the tasks of Semantic and Instance Segementation. Specifically, proposal-based approaches like [2] *decouple* mask and class prediction. Conversely, semantic segmentation frameworks typically perform per-pixel multi-class categorisation, which necessarily couples classification and segmentation [2, 3, 9].

Mask R-CNN serves as a baseline for numerous state-of-the-art approaches to the tasks of Instance [105, 106] and Panoptic Segmentation [4]. Notably, in [106], Liu *et al.* propose a state-of-the-art framework, which improves the performance of Mask R-CNN with a Path Aggregation Network (PANet) [106]. Specifically, PANet improves information flow in Mask-RCNN by shortening the path between lower and higher layers in the feature hierarchy and improving feature localisation [106]. In an alternative approach, Chen *et al.* [59] interleave bounding box localisation, mask regression and fully convolutional semantic segmentation to augment spatial mask alignment and improve instance segmentation performance [46, 59]. In another state-of-the-art approach, Kang *et al.* propose the integration of a novel bounding shape masks module to predict the boundary shape of objects, and substantially improve the performance of Mask R-CNN [105].

Several Proposal-based frameworks also approach real-time performance by adapting faster One-Stage Object Detection frameworks to the task of Instance Segmentation. In [107], Xu *et al.* propose a top-down network, which adapts the YOLOv3 [19] object detection framework, and uses Chebyshev polynomials to fit instance masks [107]. Bolya *et al.* propose a fully convolutional real-time framework that generates in parallel: a set of prototype mask proposals, and per-instance mask coefficient predictions, which are linearly combined to produce instance masks. Lee *et al.* [108] propose a similar network to Mask R-CNN [2], which includes a feature extraction backbone, a bounding box prediction head that uses the FCOS (Fully Convolutional One-Stage Object Detection) [109] anchor-free object detector, and an SAG-Mask module that uses a spatial attention model to guide instance mask prediction [108]. With a ResNet-101-FPN feature extraction backbone, CenterMask outperforms Mask R-CNN in both accuracy and prediction speed [108]. With a VoVNet [34] backbone, CenterMask outperforms real-time instance segmentation frameworks such as YOLACT [110] at several speed thresholds [108].

Recently, two-stage architectures have been adapted to the tasks of Panoptic Segmentation and Video Instance Segmentation. Notably, Panoptic Segmentation frameworks such as [4] and [5] that adopt a Mask R-CNN based Instance Segmentation submodule have achieved state-of-the-art performance on the Cityscapes Instance Segmentation challenge [10]. Panoptic frameworks like EfficentPS [5], UPSNet [4] and [1] all similarly include parallel Semantic and Instance Segmentation submodules inspired by [2], whose outputs are fused with a final Panoptic module.

Recently, Voigtlaender *et al.* [37] proposed a novel variation of VOS and Instance Segmentation tasks called Multiple Object Tracking and Segmentation (MOTS). MOTS extends the task of Multiple Object Tracking (MOT) by generating a binary mask of each object detected within a localising bounding box produced by a MOT framework [37]. MOTS is functionally equivalent to the task of Video Instance Segmentation, which is defined by Yang *et al.* as the simultaneous detection, segmentation and tracking of object instances in videos [35]. In both [37] and [35], Mask R-CNN [2] is used as the baseline framework. Voigtlaender *et al.* [37] extend Mask R-CNN with 3D convolutions to integrate temporal context to inform instance predictions, and an association head that links tracked objects over time [37]. Comparatively, Yang *et al.* [35] extend Mask R-CNN with a tracking branch computed in parallel to the bounding box regression and mask segmentation heads. The tracking head facilitates the continuous tracking of distinct objects through interaction with a *memory queue* that maintains representations of previously detected instances to inform predictions.

Proposal-Free Approaches: Two early approaches proposed by Zhang *et al.* featured probabilistic post-processing modules - a Markov Random Field (MRF) in [111] and a Conditional Random Field [74] in [112] - to improve the spatial consistency of instance masks. In an alternative approach, Bai and Urtasun [113] used a deep network to associate basins in a watershed transformation with object instances without post-processing [46]. Liu *et al.* [114] build upon [113] by similarly exploiting boundary prediction to separate instances, and using a neural network to construct masks with an object's compositional elements through sequential groupings [46, 114]. Building upon earlier approaches, three notable proposal-free systems achieve state-of-the-art performance on the Cityscapes dataset. Sofiiuk *et al.* [115] Sofiiuk *et al.* [115] perform class-agnostic instance segmentation with a point proposal network that generates an object mask around a point by leveraging Adaptive Instance Normalisation (AdaIN) [116] layers in a CNN [115]. Neven *et al.* [117] improve upon the accuracy of Mask-RCNN by 5% and perform fast (10FPS) segmentation in a proposal-free system, using a clustering loss function that groups the spatial embeddings of pixels belonging to the same object instances. Gao *et al.* [55] propose a single shot fully convolutional network that jointly computes pixel-pair affinities and semantic segmentation labels to predict instance and panoptic segmentation masks. Specifically, they employ a pixel-pair affinity pyramid, which learns hierarchical associations between pixels by computing the probability that two pixels belong to the same instance [55]. Finally, a cascaded graph partition module integrates the per-pixel semantic labels and pixe-pair affinities [55].

Contour-based Approaches: This set of approaches segments object instances by finding and fitting polygons (represented by a set of vertices) to object boundaries, rather than treating Instance Segmentation as a pixel-wise labelling task [46]. Contour-based frameworks exploit polygons to represent object instances [57]. In an early contour-based approach, Acuna *et al.* [118] proposed an interactive (human-in-the-loop) framework, that uses an RNN to sequentially predict polygonal vertices, and a Graph Neural Network to improve annotation accuracy on high-resolution images [118]. Building on [118] and the proposal-based approaches [2] and [4], Liang *et al.* developed a contour-based approach known as PolyTransform, which achieves the highest instance segmentation accuracy on the Cityscapes Benchmark [10]. The Polytransform network consists of 3 modules: (1) a segmentation network that generates a bounding box and polygon mask for each object instance, (2) Feature pyramid network (FPN) that extracts strong object boundary features and (3) a deforming network that fits the polygon vertices to boundaries of object instances [46]. In an alternative approach, Peng *et al.* [99] achieve real-time performance with a two-stage pipeline of contour proposal and contour deformation using a deep snake Active Contour Model (ACM) [57]. This differs from pixel-based approaches like Mask R-CNN, since contour-based representations are represented by a set of vertices, which are deformed to fit an object's boundaries [99].

4.2 Evaluation Metrics

The standard metric for Instance Segmentation is the average precision (AP) [61] of predicted instance masks computed at the region level [10]. AP is a function of precision and recall computed for a segmentation mask prediction compared to the ground truth mask. Mathematically, precision is defined as $\frac{TP}{TP+FP}$ and recall is defined as $\frac{TP}{TP+FN}$ where TP, FP and FN are the number of true positive, false positive and false negative pixels in the predicted instance segmentation mask respectively. Precision measures the number of correct predictions amongst all predictions made and recall measures the correct predictions compared to all possible correct predictions (i.e. the ground truth mask). Average Precision is defined as the area under the precision-recall (PR) curve for a set of instance mask predictions [61].

In practice, to determine whether a predicted segmentation mask is correct or not (True Positive or False Positive), the Jaccard IoU index [49] is used with a specified *overlap* threshold [10, 61]. For example, for an overlap threshold of 50% (denoted as AP_{50}): if there is an overlap of 50% or greater (i.e. $IoU \geq 0.5$) between a predicted instance segmentation mask and its corresponding ground truth mask, then it is considered a true positive prediction; else it is considered a false positive [61]. Duplicate predictions of the same ground truth mask are also penalised as false positivies [10, 61].

To assess the performance of a system based on its set of instance mask predictions, AP is computed for each class and averaged across a range of overlap thresholds [10]. The standard method of measurement for AP was established for the Microsoft COCO competition [48], which measures AP for 10 overlaps between 0.5 and 0.95 with steps of 0.05 [8, 10]. The purpose of computing AP for different overlap thresholds is

to assess the general applicability of an Instance Segmentation framework to a variety of computer vision applications that have varying requirements for segmentation accuracy [61].

Finally, a single compound score, the mean Average Precision (mAP) is obtained by computing the average AP score across the entire class label set [10]. Some common ways to denote mAP are: $IoU\epsilon[0.5 : 0.05 : 0.95]$ [8] and mAP@[.5, .95] [48].

5 Panoptic Segmentation

Panoptic segmentation essentially combines the tasks of semantic and instance segmentation by assigning a class label and an object instance identifier to each pixel in an image [1]. As with Semantic Segmentation, every pixel is assigned a class label. As with Instance Segmentation, pixels belonging to countable object instance representations are assigned an instance-identifying label in addition to the pixel's class label. Pixels belonging to amorphous *background* objects with undefined contours are not assigned an instance-identifier because they represent objects that are not countable.

In the paper that established the accepted definition of the Panoptic Segmentation task, Kirrilov *et al.* define 'panoptic' as a "unified and global view" of image segmentation [1]. Furthermore, they establish a clear delineation between the objectives of Semantic and Instance segmentation using two distinct categories of object perception and material recognition. While the former task is concerned with defining *stuff*, the latter is concerned with counting *things*. These categories of object perception were established by Adelson in [6], where he defines *stuff* as *amorphous* objects with similar textures or material, and *things* as *countable* objects that can be disntiguished from each other.

Kirrilov *et al.* emphasise that approaches to Semantic and Instance Segmentation have naturally and significantly diverged, and have become specialised to interpet either *stuff* or *things* independently. For example, while Semantic Segmentation frameworks typically adopt a bottom-up approach based on a FCN, Instance Segmentation methods are typically based on top-down proposal-based architectures. Consequently, they emphasised the need to unify Semantic and Instance interpretation frameworks to target a more holistic understanding of scenes with Panoptic Segmentation.

5.1 Approaches

There are two broad categories of architectures, into which Panoptic Segmentation approaches can be generally divided: Top-Down approaches, and Bottom-Up approaches. Top-Down approaches typically feature two *heads* (or modules) - separate Semantic and Instance Segmentation modules - and combine the outputs of these heads in some way [1, 5]. Conversely, Bottom-up approaches view the panoptic segmentation problem holistically, and propose a single panoptic segmentation module [11, 58]. Top-Down approaches treat Panoptic Segmentation as an amalgamation of Semantic and Instance Segmentation tasks, while Bottom-up approaches treat it as a single, holistic task.

Top Down Top-Down frameworks typically feature parallel Semantic and Instance Segmentation *heads* (or modules), whose outputs are fused to produce a Panoptic Segmentation map. The Semantic sub-module typically produces pixel-wise semantic labels via a FCN. The Instance Segmentation head is generally a variant of Mask R-CNN and employs a proposal-based, top-down approach to identify instance masks. In [1], Kirrilov *et al.* proposed a baseline model for Pantoptic Segmentation that independently computes semantic labels with a PSPNet (Pyramid Scene Parsing Network) [87] and Instance masks with Mask R-CNN [2], and combines the outputs of both in a heuristic panoptic module that predicts a *stuff*, *thing* or *void* label for each pixel [1, 4]. De Geus *et al.* [119] proposed a single network for Panoptic Segmentation with a Pyramid Pooling [56] semantic segmentation branch and a Mask R-CNN instance segmentation branch that are jointly trained on a shared ResNet-50 feature extraction backbone. Li *et al.* [120] proposed an Attention-guided Unified Network (AUNet) to improve the segmentation of semantic *stuff* classes using region proposal and mask attention modules computed from the Instance Segmenation stream.

The framework developed by Kirrilov *et al.* [1], and its derivative works [119, 120] produce overlapping semantic and instance segmentation masks, resulting in inconsistent classification of pixels as things or stuff [5]. Li *et al.* propose a shared backbone architecture that filters the semantic and instance modules' output

distribution through a binary mask to enforce consistent labelling of things and stuff. Liu *et al.* [121] propose a spatial ranking module to deal with occlusion between predicted instances. The state-of-the-art network UPSNet [4] introduces a parameter-free panoptic head that leverages logits from the Semantic and Instance heads to resolve overlap conflicts and integrates a mechanism to predict an *unknown* class for a pixel, to avoid making the wrong prediction. Porzi *et al.* [64] exploits a DeepLab-like semantic segmentation to integrate contextual information with multi-scale features generated by a FPN. In another state-of-the-art approach Li *et al.* exploit a dense instance affinity mechanism within a panoptic submodule to predict the probability that two pixels belong to the same instance. Mohan and Valada [5] propose a 2-way Feature Pyramid Network [32] as the shared backbone for semantic and instance segmentation heads followed by a panoptic fusion module. Their network, EfficientPS, employs a Mask R-CNN variant as its instance head and a novel semantic head that consists of dense prediction cells and residual pyramids [5] inspired by Atrous Spatial Pyramid Pooling. This network achieves first place on the Cityscapes [10] challenge leaderboard.

Bottom-Up Yang *et al.* [58] proposed the first proposal-free, single shot panoptic framework, DeeperLab, which employs a fully convolutional encoder-decoder architecture with an ASPP backbone to perform class-agnostic instance and semantic segmentation. Building upon [58] Cheng *et al.* [11] introduce a dual-ASPP (encoder), dual-decoder architecture for both Instance and Semantic Segmentation tasks. Both [58] and [11] fuse the results of the Instance and Semantic segmentation streams by a "majority vote", proposed in [58]. In an alternative approach, Gao *et al.* [55] jointly learn semantic labelling in conjunction with an Instance Segmentation head that learns pixel groupings via a pixel-pair affinity pyramid to efficiently generate instance masks.

5.2 Evaluation Metrics

The standard measurement for Panoptic Segmentation, Panoptic Quality (PQ), was established by Kirrilov *et al.* in [1]. PQ is computed independently for each semantic class and averaged over the classes to improve the metric's robustness to imbalances in class representations [1]. PQ adapts the PASCAL VOC IoU index [49], and computes the pixel sets: TP (true positives), FP (false positives) and FN (false negatives), as well as IoU, for each class by comparing the predicted segmentation mask with the ground-truth mask. Mathematically, PQ is defined in [1] as:

$$PQ = \frac{\sum_{(p,g)\epsilon TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$$

PQ can also be understood as the multiplication of two metrics: *Segmentation quality* (SQ) and *Recognition quality* (RQ) defined as follows [1]:

$$PQ = \underbrace{\frac{\sum_{(p,g)\epsilon TP} \mathrm{IoU}(p,g)}{|TP|}}_{SQ} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{RQ}$$

SQ is the average IoU score for matched segments in the predicted and ground truth mask [1]. Notably, RQ resembles the F_1 score [122] commonly used in the quality evaluation of object detection frameworks [1, 123]. In the Cityscapes Benchmark Suite, SQ and RQ are measured separately and Panoptic Quality is computed as a function of both metrics: $PQ = SQ \times RQ$.

6 Progress Report

6.1 Work Summary

Originally, I registered to undertake a thesis project focused on Video Object Segmentation (VOS). In the early stages of the semester, my initial research efforts were directed towards developing a general understanding of the topic and its applications, with a specific focus on how VOS could be applied to real-life, practical situations and applications. For my thesis proposal, I submitted a 10-page report that demonstrated a plan for investigating the practical applications of VOS frameworks, with a central focus on real-time processing and efficient computation, and implementing a VOS system that improved upon a baseline model and demonstrated the findings of my research. In the weeks following the submission of my thesis proposal, my supervisor and I mutually decided that I should focus my efforts on Street Scene Understanding, since it was an area in which we both have interest. Additionally, while researching VOS systems, I discovered and became interested in other *scene-based* segmentation tasks including Semantic, Instance and Panoptic segmentation. Naturally, street scenes feature many objects of interest. However, VOS is a binary segmentation task with the objective of tracking a single object. Thus, street scene understanding is best addressed by these other scene-based tasks since they have the objective of obtaining a global understanding of a scene featuring multiple objects of interest.

Based on these findings, my supervisor and I mutually decided that it would be more interesting to pursue and research Semantic, Instance or Panoptic segmentation frameworks, which are commonly applied to street scene understanding applications. Fortunately, much of the foundational knowledge that I obtained from researching VOS transferred to my understanding of Scene Segmentation, since the state-of-the-art frameworks in each paradigm both commonly employ deep learning convolutional network architectures. For the weeks including and following the mid-semester break, I thoroughly researched the paradigm of street scene and general scene segmentation; focusing on state-of-the-art frameworks in particular. Additionally, I also focused on Panoptic Segmentation literature, since these frameworks provide the most comprehensive and complex understanding of a scene, in terms of both semantic categorisation and object instance identification. This eventually led to my decision that I would make Panoptic Segmentation as the central focus of my project research and practical demonstration. Furthermore, during this time, I also dedicated several hours to learning and understanding the technicalities and implementation details of deep learning-based computer vision frameworks.

Finally, in weeks 10-12 of the semester, I dedicated my focus to writing a 14-page literature review. This phase of writing involved many more hours of research as I read papers, sought to truly understand scene segmentation architectures, and read the papers referenced by the state-of-the-art papers that I had prioritised. This phase culminated in the completion of my literature survey on Semantic, Instance and Panoptic segmentation frameworks and their corresponding datasets and challenges (as well as an overview of similar computer vision tasks).

6.2 Revised Work Plan

The next phase of my project can be divided into three general stages: technical research, prototype development, and thesis writing. In the first stage, I will focus firstly on identifying a baseline Panoptic segmentation system, which I will choose to extend and improve by applying the findings of my research. Alternatively, I will identify a baseline semantic and instance segmentation frameworks to combined in a unified panoptic framework, which fuses the semantic and instance information generated by the two semantic and instance segmentation modules. Throughout this stage, I will test the baseline system(s) on street scene segmentation datasets such as Cityscapes [10] and Mapillary Vistas [47] and record these results to be included in my final thesis report. Additionally, I will investigate the source code of numerous state-of-the-art panoptic segmentation architectures. Ultimately, the focus of this stage is to learn the inner workings of a panoptic segmentation framework and to learn how to implement this system through an analysis of code and corresponding literature. I will also test various state-of-the-art and baseline frameworks so that the results of these frameworks can be used for comparison in the evaluation of my own prototype.

In the second stage, I will focus on developing the prototype, and writing about the methodology that I am implementing and the baseline architecture that I am extending. This practical phase primarily involves writing code and testing. Furthermore, it will involve testing and recording the results of my system on street

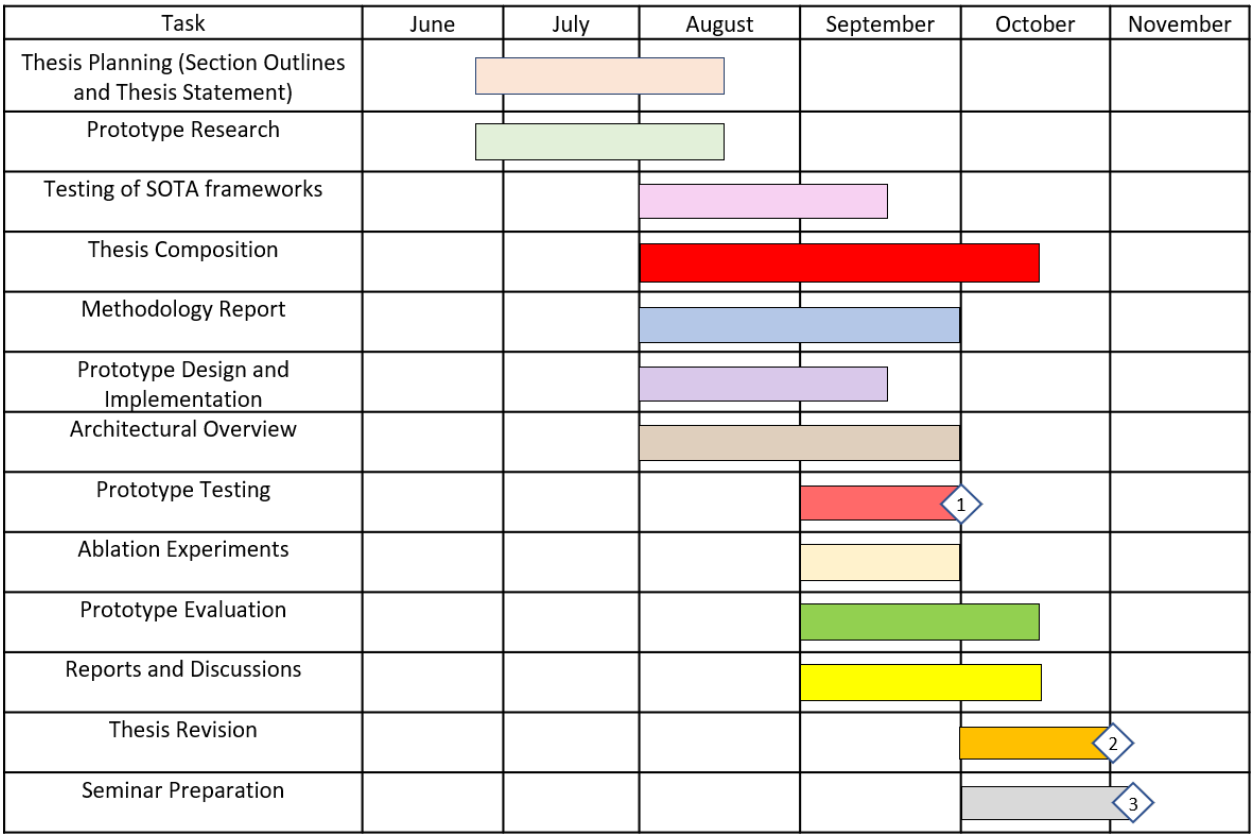
scene segmentation datasets. In this stage, I will record results, evaluate them and write a discussion on these results. Furthermore, I will test different implementation variations, such as different feature extraction backbones, or different variations of the semantic and instance segmentation modules. Finally, in the third stage, I will focus on writing out my thesis paper. Throughout the whole semester, I will be writing down dot points under section headers that I will plan out before the beginning of the semester. This final writing stage will involve expanding on those points, formulating discussions and evaluations, and connecting the sections with an underlying theoretical focus and theme.

6.3 Gantt Timeline

Start date: June 25th
End date: November 8th

Milestones:

1. October 1: Finalise system implementation and experimentation phase
2. October 30: Finalise thesis paper
3. November 1-8: Present thesis seminar



References

[1] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," *CoRR*, vol. abs/1801.00868, 2018. arXiv: 1801.00868. [Online]. Available: http://arxiv.org/abs/1801.00868.

[2] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. arXiv: 1703.06870. [Online]. Available: http://arxiv.org/abs/1703.06870.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[4] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," *CoRR*, vol. abs/1901.03784, 2019. arXiv: 1901.03784. [Online]. Available: http://arxiv.org/abs/1901.03784.

[5] R. Mohan and A. Valada, *Efficientps: Efficient panoptic segmentation*, 2020. arXiv: 2004.02307 [cs.CV].

[6] E. Adelson, "On seeing stuff: The perception of materials by humans and machines," *Proceedings of SPIE*, vol. 4299, Jun. 2001. DOI: 10.1117/12.429489.

[7] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. S. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *CoRR*, vol. abs/1809.03327, 2018. arXiv: 1809.03327. [Online]. Available: http://arxiv.org/abs/1809.03327.

[8] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. arXiv: 1506.01497. [Online]. Available: http://arxiv.org/abs/1506.01497.

[9] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016. arXiv: 1606.00915. [Online]. Available: http://arxiv.org/abs/1606.00915.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," eng, *arXiv.org*, 2016. [Online]. Available: http://search.proquest.com/docview/2078173514/.

[11] C. Bowen, M. Collins, Y. Zhu, T. Liu, T. Huang, A. Hartwig, and L.-C. Chen, "Panoptic-deeplab," eng, *arXiv.org*, 2019. [Online]. Available: http://search.proquest.com/docview/2304015185/.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. arXiv: 1311.2524. [Online]. Available: http://arxiv.org/abs/1311.2524.

[13] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. arXiv: 1504.08083. [Online]. Available: http://arxiv.org/abs/1504.08083.

[14] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," *CoRR*, vol. abs/1711.11575, 2017. arXiv: 1711.11575. [Online]. Available: http://arxiv.org/abs/1711.11575.

[15] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," *CoRR*, vol. abs/1811.11168, 2018. arXiv: 1811.11168. [Online]. Available: http://arxiv.org/abs/1811.11168.

[16] G. Ghiasi, T. Lin, R. Pang, and Q. V. Le, "NAS-FPN: learning scalable feature pyramid architecture for object detection," *CoRR*, vol. abs/1904.07392, 2019. arXiv: 1904.07392. [Online]. Available: http://arxiv.org/abs/1904.07392.

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR*, vol. abs/1703.06211, 2017. arXiv: 1703.06211. [Online]. Available: http://arxiv.org/abs/1703.06211.

[18] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, 128837–128868, 2019, ISSN: 2169-3536. DOI: 10.1109/access.2019.2939201. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2019.2939201.

[19] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, 2018. arXiv: 1804.02767 [cs.CV].

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. arXiv: 1512.02325. [Online]. Available: http://arxiv.org/abs/1512.02325.

[21] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," *CoRR*, vol. abs/1701.06659, 2017. arXiv: 1701.06659. [Online]. Available: http://arxiv.org/abs/1701.06659.

[22] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. arXiv: 1708.02002. [Online]. Available: http://arxiv.org/abs/1708.02002.

[23] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," *CoRR*, vol. abs/1711.06897, 2017. arXiv: 1711.06897. [Online]. Available: http://arxiv.org/abs/1711.06897.

[24] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," *CoRR*, vol. abs/1811.04533, 2018. arXiv: 1811.04533. [Online]. Available: http://arxiv.org/abs/1811.04533.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[26] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016. arXiv: 1611.05431. [Online]. Available: http://arxiv.org/abs/1611.05431.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. arXiv: 1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861.

[28] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018. arXiv: 1801.04381. [Online]. Available: http://arxiv.org/abs/1801.04381.

[29] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *CoRR*, vol. abs/1707.01083, 2017. arXiv: 1707.01083. [Online]. Available: http://arxiv.org/abs/1707.01083.

[30] F. Iandola, M. Moskewicz, K. Ashraf, S. Han, W. Dally, and K. Keutzer, *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and textless1mb model size*, Feb. 2016.

[31] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. arXiv: 1610.02357. [Online]. Available: http://arxiv.org/abs/1610.02357.

[32] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946.

[33] R. J. Wang, X. Li, S. Ao, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *CoRR*, vol. abs/1804.06882, 2018. arXiv: 1804.06882. [Online]. Available: http://arxiv.org/abs/1804.06882.

[34] Y. Lee, J. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and gpu-computation efficient backbone network for real-time object detection," *CoRR*, vol. abs/1904.09730, 2019. arXiv: 1904.09730. [Online]. Available: http://arxiv.org/abs/1904.09730.

[35] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," *CoRR*, vol. abs/1905.04804, 2019. arXiv: 1905.04804. [Online]. Available: http://arxiv.org/abs/1905.04804.

[36] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, 61–88, Mar. 2020, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.11.023. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2019.11.023.

[37] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," *CoRR*, vol. abs/1902.03604, 2019. arXiv: 1902.03604. [Online]. Available: http://arxiv.org/abs/1902.03604.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[40] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Computer Vision and Pattern Recognition*, 2016.

[41] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *CoRR*, vol. abs/1904.09172, 2019. arXiv: 1904.09172. [Online]. Available: http://arxiv.org/abs/1904.09172.

[42] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. Chen, "FEELVOS: fast end-to-end embedding learning for video object segmentation," *CoRR*, vol. abs/1902.09513, 2019. arXiv: 1902.09513. [Online]. Available: http://arxiv.org/abs/1902.09513.

[43] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," *CoRR*, vol. abs/1612.02646, 2016. arXiv: 1612.02646. [Online]. Available: http://arxiv.org/abs/1612.02646.

[44] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool, "One-shot video object segmentation," *CoRR*, vol. abs/1611.05198, 2016. arXiv: 1611.05198. [Online]. Available: http://arxiv.org/abs/1611.05198.

[45] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, *Deep high-resolution representation learning for visual recognition*, 2019. arXiv: 1908.07919 [cs.CV].

[46] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun, *Polytransform: Deep polygon transformer for instance segmentation*, 2019. arXiv: 1912.02801 [cs.CV].

[47] G. Neuhold, T. Ollmann, S. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," English, vol. 2017-, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 5000–5009, ISBN: 9781538610329.

[48] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[49] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, Jan. 2014. DOI: 10.1007/s11263-014-0733-5.

[50] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. arXiv: 1706.05587. [Online]. Available: http://arxiv.org/abs/1706.05587.

[51] S. A. Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh, *Deep semantic segmentation of natural and medical images: A review*, 2019. arXiv: 1910.07655 [cs.CV].

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597.

[53] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. arXiv: 1606.04797. [Online]. Available: http://arxiv.org/abs/1606.04797.

[54] S. Minaee, Y. Boykov, F. M. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *ArXiv*, vol. abs/2001.05566, 2020.

[55] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang, *Ssap: Single-shot instance segmentation with affinity pyramid*, 2019. arXiv: 1909.01616 [cs.CV].

[56] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, vol. abs/1612.01105, 2016. arXiv: 1612.01105. [Online]. Available: http://arxiv.org/abs/1612.01105.

[57] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," eng, *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988, ISSN: 0920-5691.

[58] T. Yang, M. D. Collins, Y. Zhu, J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L. Chen, "Deeperlab: Single-shot image parser," *CoRR*, vol. abs/1902.05093, 2019. arXiv: 1902.05093. [Online]. Available: http://arxiv.org/abs/1902.05093.

[59] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid task cascade for instance segmentation," *CoRR*, vol. abs/1901.07518, 2019. arXiv: 1901.07518. [Online]. Available: http://arxiv.org/abs/1901.07518.

[60] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov. 2015, pp. 730–734. DOI: 10.1109/ACPR.2015.7486599.

[61] B. Hariharan, P. Arbelaez, R. B. Girshick, and J. Malik, "Simultaneous detection and segmentation," *CoRR*, vol. abs/1407.1808, 2014. arXiv: 1407.1808. [Online]. Available: http://arxiv.org/abs/1407.1808.

[62] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. D. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," *CoRR*, vol. abs/1812.01593, 2018. arXiv: 1812.01593. [Online]. Available: http://arxiv.org/abs/1812.01593.

[63] D. de Geus, P. Meletis, and G. Dubbelman, *Single network panoptic segmentation for street scene understanding*, 2019. arXiv: 1902.02678 [cs.CV].

[64] L. Porzi, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," eng, *arXiv.org*, 2019. [Online]. Available: http://search.proquest.com/docview/2220544295/.

[65] P. R. Palafox, J. Betz, F. Nobis, K. Riedl, and M. Lienkamp, "Semanticdepth: Fusing semantic segmentation and monocular depth estimation for enabling autonomous driving in roads without lane lines," eng, *Sensors*, vol. 19, no. 14, 2019, ISSN: 1424-8220. [Online]. Available: https://doaj.org/article/d2eb36d468e04734820fd329aae4fae0.

[66] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," Oct. 2012, pp. 746–760. DOI: 10.1007/978-3-642-33715-4_54.

[67] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," Jun. 2013. DOI: 10.13140/2.1.2577.6000.

[68] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130.

[69] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1–8.

[70] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1972–1979.

[71] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, 416–423 vol.2.

[72] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3282–3289.

[73] A. Geiger, P Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231–1237, Sep. 2013. DOI: 10.1177/0278364913491297.

[74] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012. arXiv: 1210.5644. [Online]. Available: http://arxiv.org/abs/1210.5644.

[75] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," *CoRR*, vol. abs/1509.02634, 2015. arXiv: 1509.02634. [Online]. Available: http://arxiv.org/abs/1509.02634.

[76] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[77] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.

[78] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. arXiv: 1612.03144. [Online]. Available: http://arxiv.org/abs/1612.03144.

[79] X. Zhang, H. Xu, H. Mo, J. Tan, C. Yang, and W. Ren, *Dcnas: Densely connected neural architecture search for semantic image segmentation*, 2020. arXiv: 2003.11883 [cs.CV].

[80] X. Li, L. Zhang, A. You, M. Yang, K. Yang, and Y. Tong, "Global aggregation then local distribution in fully convolutional networks," in *BMVC2019*.

[81] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML].

[82] "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, ISSN: 0925-2312.

[83] N. Plath and M. Toussaint, "Multi-class image segmentation using conditional random fields and global classication," vol. 382, Jan. 2009. DOI: 10.1145/1553374.1553479.

[84] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015. arXiv: 1505.04366. [Online]. Available: http://arxiv.org/abs/1505.04366.

[85] Y. Yuan, X. Chen, and J. Wang, *Object-contextual representations for semantic segmentation*, 2019. arXiv: 1909.11065 [cs.CV].

[86] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. arXiv: 1612.03144. [Online]. Available: http://arxiv.org/abs/1612.03144.

[87] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, vol. abs/1612.01105, 2016. arXiv: 1612.01105. [Online]. Available: http://arxiv.org/abs/1612.01105.

[88] G. Ghiasi and C. C. Fowlkes, "Laplacian reconstruction and refinement for semantic segmentation," *CoRR*, vol. abs/1605.02264, 2016. arXiv: 1605.02264. [Online]. Available: http://arxiv.org/abs/1605.02264.

[89] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3561–3571.

[90] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2393–2402.

[91] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7511–7520.

[92] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *ECCV*, 2018.

[93] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," *CoRR*, vol. abs/1704.03604, 2017. arXiv: 1704.03604. [Online]. Available: http://arxiv.org/abs/1704.03604.

[94] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018. arXiv: 1802.02611. [Online]. Available: http://arxiv.org/abs/1802.02611.

[95] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Muller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.

[96] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. arXiv: 1606.02147. [Online]. Available: http://arxiv.org/abs/1606.02147.

[97] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," Jul. 2018. DOI: 10.1109/CVPR.2018.00388.

[98] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, *Yolact++: Better real-time instance segmentation*, 2019. arXiv: 1912.06218 [cs.CV].

[99] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, *Deep snake for real-time instance segmentation*, 2020. arXiv: 2001.01629 [cs.CV].

[100] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, *Solov2: Dynamic, faster and stronger*, 2020. arXiv: 2003.10152 [cs.CV].

[101] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," *CoRR*, vol. abs/1412.1283, 2014. arXiv: 1412.1283. [Online]. Available: http://arxiv.org/abs/1412.1283.

[102] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," *CoRR*, vol. abs/1603.08678, 2016. arXiv: 1603.08678. [Online]. Available: http://arxiv.org/abs/1603.08678.

[103] P. H. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," *CoRR*, vol. abs/1506.06204, 2015. arXiv: 1506.06204. [Online]. Available: http://arxiv.org/abs/1506.06204.

[104] P. H. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," *CoRR*, vol. abs/1603.08695, 2016. arXiv: 1603.08695. [Online]. Available: http://arxiv.org/abs/1603.08695.

[105] H. Y. Kim and B. R. Kang, "Instance segmentation and object detection with bounding shape masks," *CoRR*, vol. abs/1810.10327, 2018. arXiv: 1810.10327. [Online]. Available: http://arxiv.org/abs/1810.10327.

[106] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *CoRR*, vol. abs/1803.01534, 2018. arXiv: 1803.01534. [Online]. Available: http://arxiv.org/abs/1803.01534.

[107] W. Xu, H. Wang, F. Qi, and C. Lu, *Explicit shape encoding for real-time instance segmentation*, 2019. arXiv: 1908.04067 [cs.CV].

[108] Y. Lee and J. Park, *Centermask : Real-time anchor-free instance segmentation*, 2019. arXiv: 1911.06667 [cs.CV].

[109] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: fully convolutional one-stage object detection," *CoRR*, vol. abs/1904.01355, 2019. arXiv: 1904.01355. [Online]. Available: http://arxiv.org/abs/1904.01355.

[110] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, *Yolact: Real-time instance segmentation*, 2019. arXiv: 1904.02689 [cs.CV].

[111] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation with deep densely connected mrfs," *CoRR*, vol. abs/1512.06735, 2015. arXiv: 1512.06735. [Online]. Available: http://arxiv.org/abs/1512.06735.

[112] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun, "Monocular object instance segmentation and depth ordering with cnns," *CoRR*, vol. abs/1505.03159, 2015. arXiv: 1505.03159. [Online]. Available: http://arxiv.org/abs/1505.03159.

[113] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," *CoRR*, vol. abs/1611.08303, 2016. arXiv: 1611.08303. [Online]. Available: http://arxiv.org/abs/1611.08303.

[114] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3516–3524.

[115] K. Sofiiuk, O. Barinova, and A. Konushin, *Adaptis: Adaptive instance selection network*, 2019. arXiv: 1909.07829 [cs.CV].

[116] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *CoRR*, vol. abs/1703.06868, 2017. arXiv: 1703.06868. [Online]. Available: http://arxiv.org/abs/1703.06868.

[117] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," *CoRR*, vol. abs/1906.11109, 2019. arXiv: 1906.11109. [Online]. Available: http://arxiv.org/abs/1906.11109.

[118] L. Castrejón, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," *CoRR*, vol. abs/1704.05548, 2017. arXiv: 1704.05548. [Online]. Available: http://arxiv.org/abs/1704.05548.

[119] D. de Geus, P. Meletis, and G. Dubbelman, "Panoptic segmentation with a joint semantic and instance segmentation network," *CoRR*, vol. abs/1809.02110, 2018. arXiv: 1809.02110. [Online]. Available: http://arxiv.org/abs/1809.02110.

[120] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," *CoRR*, vol. abs/1812.03904, 2018. arXiv: 1812.03904. [Online]. Available: http://arxiv.org/abs/1812.03904.

[121] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, "An end-to-end network for panoptic segmentation," *CoRR*, vol. abs/1903.05027, 2019. arXiv: 1903.05027. [Online]. Available: http://arxiv.org/abs/1903.05027.

[122] C. J. Van Rijsbergen, *Information retrieval*, eng, 2d ed. London ; Butterworths, ISBN: 0408709294.

[123] D. Martin, C. Fowlkes, and J Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," eng, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004, ISSN: 0162-8828.