

***GAN Mask R-CNN: Instance semantic
segmentation benefits from generative adversarial
networks***

by

Quang H. Le

S.B., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2020

©2020 Quang H. Le. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and to
distribute publicly paper and electronic copies of this thesis document
in whole and in part in any medium now known or hereafter created.

Author

Department of Electrical Engineering and Computer Science

January 29, 2020

Certified by

Kamal Youcef-Toumi

Professor

Thesis Supervisor

Accepted by

Katrina LaCurts

Chair, Master of Engineering Thesis Committee

GAN Mask R-CNN: Instance semantic segmentation benefits from generative adversarial networks

by

Quang H. Le

Submitted to the Department of Electrical Engineering and Computer Science
on January 29, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

In designing instance segmentation ConvNets that reconstruct masks, segmentation is often taken as its literal definition –assigning label to every pixel– for defining the loss functions. That is, using losses that compute the difference between pixels in the predicted (reconstructed) mask and the ground truth mask –a template matching mechanism. However, any such instance segmentation ConvNet is a generator, so we can lay the problem of predicting masks as a GANs game framework: We can think the ground truth mask is drawn from the true distribution, and a ConvNet like Mask R-CNN is an implicit model that infers the true distribution. In GANs terms, Mask R-CNN is the generator who reconstructs a mask as the fake one. We then send the fake mask and the real (ground truth) one to a discriminator (critic). By playing a min-max game, we want Mask R-CNN to fool the critic, and the critic to distinguish between real and fake masks. In this way, we take the advantage of a region proposal network (implemented in Mask R-CNN) to design a generator, and the benefit of a critic network to design a better loss function as opposed to a template matching one. We discuss how we utilize the GANs training stability regiments in practice to make this concept works. We show this GANs framework performs better than the original Mask R-CNN. Furthermore, we show the results give crisper boundaries – a traditional challenge of ConvNets where there is a trade-off between having higher level of semantics and finer boundaries.

Thesis Supervisor: Kamal Youcef-Toumi
Title: Professor

Acknowledgments

I would like to express my special thanks of gratitude to my supervisor, Prof. Kamal Youcef-Toumi for providing such a huge opportunity to work and research at Mechanics Research Lab. Thanks to Kamal's guidance, I have gained lots of professional experiences and valuable skills in my research.

Thanks also to my great labmate, Ali Jahanian, for helping me develop my research ideas and giving reliable feedback.

Contents

1	Introduction	13
2	Related Works	15
3	Method	17
3.1	GAN Overview	17
3.2	Network Architecture	18
3.2.1	Box Head	18
3.2.2	Mask Head	19
4	Experiments	21
4.1	Experimental settings and Evaluation Metrics	21
4.2	Phone Recycling Dataset	22
4.3	Cityscapes Dataset	24
4.4	COCO Dataset	26
4.5	Gland Segmentation Dataset	27
5	Conclusion	31

List of Figures

1-1	An Introduction to Instance Semantic Segmentation	14
3-1	Box Head Architecture	18
3-2	Mask Head Architecture	20
4-1	Instance Segmentation Results on Phone Recycling Dataset	23
4-2	Instance Segmentation Results on Cityscapes dataset	26
4-3	Instance Segmentation Results on 2017 COCO dataset	27
4-4	Instance Segmentation Results on Gland Segmentation dataset	29

List of Tables

4.1	Object Detection result on Phone Recycling dataset	23
4.2	Mask Segmentation result on Phone Recycling dataset	23
4.3	Object Detection result on Cityscapes dataset	25
4.4	Mask Segmentation result on Cityscapes dataset	25
4.5	Mask Segmentation result on 2017 COCO dataset	27
4.6	Object Detection result on Gland Segmentation dataset	28
4.7	Mask Segmentation result on Gland Segmentation dataset	28

Chapter 1

Introduction

Inference of instance image segmentation that includes details of crisp boundaries is still a challenging task. While the idea of using generative adversarial networks – GANs – [9] to produce segmentation mask has been explored, the results are suffering from losses that are based on averaging pixel-wise differences. We suggest to tackle this problem by using a GAN loss merged with Mask R-CNN [10] and get the best of the both worlds: power of multi-tasking on region-based networks and high-accuracy generations from the GAN loss. One can think of the Mask R-CNN as a mask generator. We can then take these generations and optimize for making them more accurate by using the min-max strategy used in GANs where there is another network – discriminator – that tries to distinguish them from the real ground truth masks. We show this strategy outperforms the pixel-wise L-norm losses.

Our main contribution is to design a complementary GAN loss which help training the model better. Results through extensive experiments show consistent performance gain in both object detection and mask segmentation tasks over state-of-the-art methods. Although we only implement our design on Mask R-CNN, this GAN loss can be extended and applied on future state-of-the-art method. For our experiments, we choose different datasets: phone recycling [13] , Cityscape autonomous driving [3], COCO objects [17], and medical gland dataset [26]. Figure 1-1 illustrates some results. Note that without bells and whistles, we simply train our model on these datasets and show our design of GAN loss quantitatively and qualitatively improves

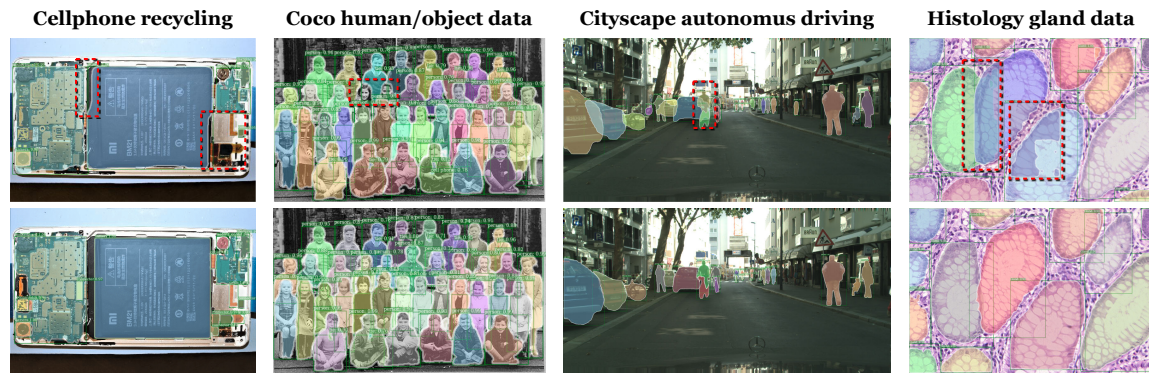


Figure 1-1: Instance semantic segmentation has applications in many domains, and each domain may have a specific goal. Using a better loss on only one generic architecture, and regardless of the kind of underlying domain, we show a GAN loss generally improves the accuracy and quality of the task. Top row: results of the baseline Mask R-CNN. Bottom row: results of replacing the baseline’s loss with a GAN loss. For ease of comparison, we highlight (with the red dotted-boxes) some failures in the baseline results.

the baseline model.

Chapter 2

Related Works

We are interested in designing a new framework for training Convnets more effectively on object detection and localization, semantic segmentation, instance segmentation and crips boundaries detection, all together. Multiple researches on optimizing these ConvNets architectures have been explored separately and benchmarked on public datasets, and yet it is still and active topic of research.

For example, for semantic segmentation, fully convolutional networks, FCN [18, 21], SegNet [1] and U-Net [25] architectures have been successful. Since these networks do not perform well in details and boundaries, several improvements have been suggested [5, 23, 15].

FPN: Driven by the idea that incorporating additional connectivity helps the different types of information to flow across different scales of network depth, recent architectures are based on skip connections from encoder to decoder for using details from different feature maps' level (Laplacian reconstruction in LLR [6], SharpMask [23]). One such architecture is FPN [16] which demonstrates significant improvement as a generic feature extractor in several visual applications. FPN utilizes a pyramidal structure (i.e., multi-level) for encoder and decoder, similar to human visual system in multi-scale visual tasks [2]. FPN further uses an adaptive pooling mechanism for aggregating the feature maps of corresponding encoder/decoder levels together (through lateral skip-connections as well as the inherited layer skip-connections from ResNet [11]) with losses at each level of the decoder. This architecture gives state-of-

the-art for any ConNet’s backbone. FPN is also used as the backbone in Mask R-CNN [10] which does object detection, localization, and instance semantic segmentation together.

Mask R-CNN is currently state-of-the-art for object detection and instance segmentation, and part of its strength is due to region-based detection mechanism . In fact this network is an evolution of prior work – RCNN [7], Fast RCNN [6], and Faster RCNN [24]. The core idea in this family of networks is to scan over the predefined regions called *anchors*. For each anchor, the Region Proposal Network (RPN) does two different type of predictions: the score of it being foreground, and the bounding box regression adjustment to best fit the object. RPN then chooses a fixed number of anchors with highest score, and apply the regression adjustment to get the final proposal for objects prediction at the network head. There are 3 type of prediction branches at the head: bounding box regression, class detection, and mask segmentation. Each head branch extracts the Regions of Interest (RoIs) from the output feature maps of the FPN backbone based on the final proposals from RPN, then feeds them through a combination of fully connected and convolution layers for final predictions. This mechanism works better for us (as we compared to SegNet). Note that there are other instance segmentation works [22, 4] that use the region proposals, however, Mask R-CNN learns all the tasks at the head (i.e., for inference).

GAN losses are applied to many networks and domains, such as [12, 19, 27], among which [27] is closest to our work. Unlike their work, our computations are in the feature maps space and thus we send feature maps to GANs’ discriminator.

Chapter 3

Method

We propose an adversarial loss design to complement the standard pixel-wise loss used in Mask R-CNN heads. We modify the prediction heads to generate more detail representations of their outcomes, and introduce the corresponding discriminators which, by considering the adversarial loss, can supervise the generative process. In this section, we will first present about the concept of GAN network in general. Then, the detail of the generator - discriminator architecture and objective function implementation are given.

3.1 GAN Overview

The learning objective in training GANs [9] model is to simultaneously find the parameters of a discriminator D that maximizes its classification accuracy, and the parameters of a generator G that maximizes the probability of D making a mistakes. This objective corresponds to a minimax two-player game, which can be formulated as:

$$\min_{\theta_G} \max_{\theta_D} L(\theta_D, \theta_G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.1)$$

where x is a real sample from an unknown distribution P_{data} , z is the random noise, $L(D, G)$ represents the cost of training, and θ_G an θ_D are the parameters of G and D

respectively.

GAN framework provide a way to learn the deep representation of the scenes more effectively. In our proposed network, we elaborate the baseline Mask R-CNN [10] by mimicking the training scheme of GANs. Further detail on the network architecture and implementation will be described in the following sections.

3.2 Network Architecture

3.2.1 Box Head

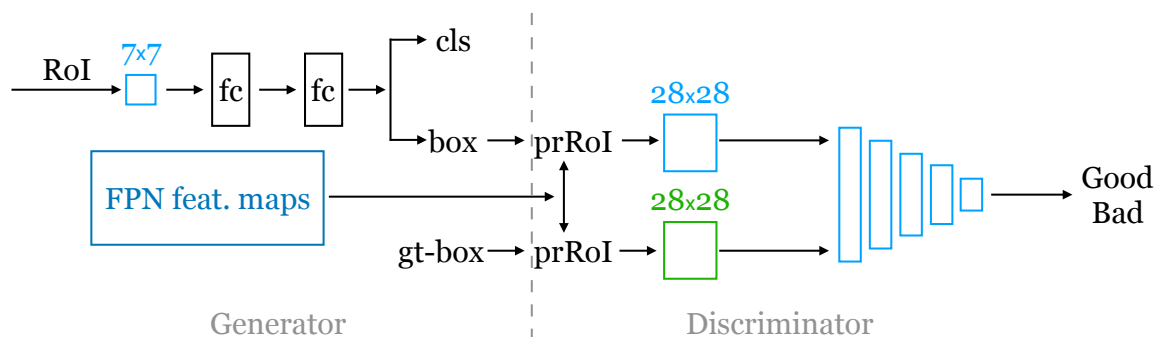


Figure 3-1: **Box Head Architecture:** We propose an adversarial training framework for object detection by extending the current box head architecture in Mask R-CNN with FPN backbone [10]. We design a new discriminator network, which takes in bounding box predictions from the box head and their ground truths, then extracts the corresponding regions of interest from the FPN feature maps using Precise Roi Pooling (prRoI) [14]. Finally, the discriminator outputs a score between [0,1] that represents how good the bounding box prediction is. There are 5 convolutional layers in the discriminator network: each convolutional layer is followed by a BatchNorm and a LeakyReLU layer.

Generator: As shown in Figure 3-1, we utilize the box head of Mask R-CNN as our box generator; thus, instead of taking random noise as input like in vanilla GAN, our generator now takes in RoI features, and output class and bounding box prediction. There are 1 convolution layers followed by 2 fully-connected layers in this network. During training, we pick only the bounding boxes that correspond to the foreground and send them to the discriminator.

Discriminator: The purpose of the box discriminator is to look at the region

defined by the bounding box prediction from the generator, tell whether they are good or bad, and send feedback via back propagation w.r.t *bounding box coordinates*. For this reason, we extract the region of interest using Precise Roi Pooling [14] on the *feature maps output* of FPN backbone [10] for both bounding box predictions and ground truths, and feed them to the network as bad and good samples respectively. This discriminator network consists of 5 convolution layer as shown in Figure 3-1. It takes in a multidimensional images of size 28×28 and outputs a score between $[0,1]$, which represents how good the prediction is (higher score is better).

Loss Function: We introduce an adversarial loss term to optimize our generator/discriminator framework as follows:

$$L_{adv}^{G_b} = \frac{1}{N} \sum_{i=1}^N -\log(D_b(G_b(RoI_i))) \quad (3.2)$$

$$L_{adv}^{D_b} = \frac{1}{N} \sum_{i=1}^N -(\log(D_b(bb_i^{gt})) + \log(1 - D_b(G_b(RoI_i)))) \quad (3.3)$$

where N is mini-batch size, $G_b(RoI_i)$, bb_i^{gt} denote the bounding box prediction and its corresponding ground truth, $D_b(\cdot)$ denotes the probability of an image being real. $L_{adv}^{G_b}$ encourages G_b to generate bounding box that can fool D_b , while $L_{adv}^{D_b}$ strengthens D_b 's ability to differentiate between real and fake bounding boxes.

3.2.2 Mask Head

Generator: Same as the box head, we also adopt the mask head of Mask R-CNN as our mask generator. This network contains 4 convolution layers followed by a deconvolution and an 1×1 convolution layer (Figure 3-2). It takes RoI features as input, and output binary masks prediction of size 28×28 . During training, we extract masks that belong to true objects, and send them to the discriminator.

Discriminator: As shown in Figure 3-2, we feed 2 inputs to the mask discriminator: the binary mask prediction and its ground truth, both multiplied element-wise with the RoI features of size 28×28 . This network has 5 convolution layers and is

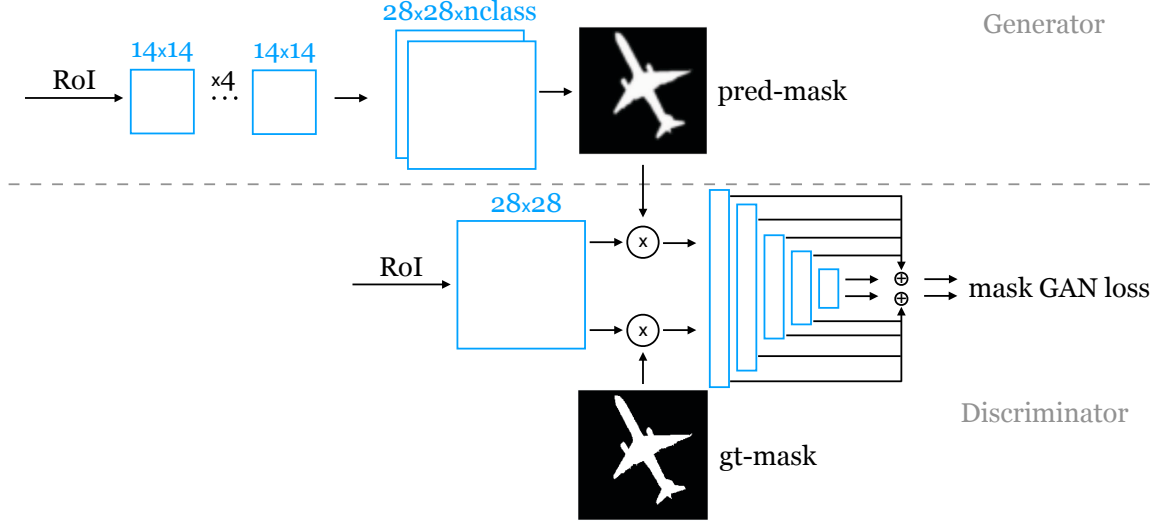


Figure 3-2: **Mask Head Architecture:** We utilize the mask head of Mask R-CNN [10] as the generator, and build another discriminator network for the mask segmentation task. During training, we first pick the binary mask predictions of true objects (pred-mask) and their ground truths (gt-mask), calculate the pixel-wise multiplication between these masks and the corresponding regions of interest (RoI), and send the results to the discriminator as inputs. Outputs from each layer of the discriminator are then concatenated and used for final GAN loss computation. Similar to the box head, the mask head also has 5 convolutional layers in the discriminator network, where each convolutional layer is followed by a BatchNorm and a LeakyReLU layer.

constructed such that the features’ spatial resolution reduces as we go deeper. We concatenate these hierarchical features from all the layers into a single output and use it to compute the adversarial loss.

Loss Function: The adversarial loss for optimizing the mask generator $L_{adv}^{G_m}$ and discriminator $L_{adv}^{D_m}$ can be formulated as $L_{adv}^{D_m} = -L_{adv}^{G_m}$, where:

$$L_{adv}^{G_m} = \frac{1}{N} \sum_{i=1}^N \|D_m(mask_i^{gt}) - D_m(G_m(RoI_i))\|_1, \quad (3.4)$$

where N is mini-batch size, $G_m(RoI_i)$, $mask_i^{gt}$ denote the binary mask prediction and its corresponding ground truth. $L_{adv}^{G_m}$ encourages G_m to minimize the differences between the mask prediction and ground truth based on the feedback from D, while $L_{adv}^{D_m}$ guide D_m to magnify those difference in reverse.

Chapter 4

Experiments

In this section, we present a thorough comparison between our model and the baseline, Mask R-CNN over multiple challenging public datasets (i.e COCO [17], Cityscapes [3], Phone Recycling dataset [13], and Gland Segmentation dataset [26]) for instance segmentation task.

4.1 Experimental settings and Evaluation Metrics

For all experiments, we use the pretrained ResNet [11] backbone with FPN architecture [16] to initialize our network. The detail implementation is based on the public Mask R-CNN benchmark project [20] by Facebook on the Pytorch platform. We train the whole network with Stochastic Gradient Descent on two NVIDIA TITAN RTX (24GB memory each). More training detail (i.e., learning rate, batch size, etc.) for each datasets will be mentioned in the following sections.

During evaluation, we report the standard COCO metric: Average Precision (AP). AP is measured over multiple IoU's (Intersection of prediction and ground truth over Union of prediction and ground truth) threshold (i.e., AP - averaged over [0.5:0.05:0.95], AP_{50} , AP_{75}), and at different object scales (i.e., AP_S , AP_M , AP_L). Scale is determined by the number of pixels: small objects ($area < 32^2$ pixels), medium objects ($32^2 < area < 96^2$) and large objects ($area > 96^2$).

4.2 Phone Recycling Dataset

The Phone Recycling dataset [13] is used for high-precision disassembling task. It consists of 533 high-resolution images of cellphone individual components as well as its different layers. So far, there are 10 cellphone models used in the dataset: *Apple iPhone 3GS, iPhone 4, iPhone 4S, iPhone 6, Samsung GT-i8268 Galaxy, Samsung S4 Active, Samsung Galaxy S6 and S6 Edge, Samsung S8 Plus, Pixel 2 XL, Xiaomi Note, HTC One, Huawei Mate 8, 9, 10, and P8 Lite*, and 11 object categories in the instance segmentation task.

Implementation: We first split the dataset into 90% for training and 10% for validation, and then apply augmentation on the training set. By adding random noise, Gaussian blur, sharpness, or changing the lightness, as well as constant normalization, we produce more than 4000 training images. For validation, we pick at least one image from each cellphone model which is not shown to our network during training.

We use ResNet-101-FPN as the backbone for both the baseline Mask R-CNN and our network. To reduce overfitting, we train with the image scale (shorter side) randomly sampled from [400, 600], and inference at 400 pixels. We use batch size of 8 images per GPU and train the network for 27k iterations. We start with a learning rate of 0.01 and reduce it to 0.001 and 0.0001 at 18k and 24k iteration respectively.

Result: Table 4.1 and 4.2 show the comparison of our network with Mask R-CNN for both object detection and mask segmentation tasks on the phone validation set. We can observe that our proposed approach outperforms the baseline Mask R-CNN in both tasks, with an average improvement in AP metric of 3% on bounding box detection (58% vs 55%) and 2% on mask segmentation (59% vs 57%).

One of the main challenge of this dataset is that there are multiple small components with only a few pixels gap between them; most of the components are flat and usually overlap each other (to save space during manufacturing), thus it is difficult to identify small object due to occlusion. Our method shows significant improvement on

	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask-RCNN	ResNet-101-FPN	0.5486	0.7333	0.6261	0.3327	0.5071	0.7462
Our Model	ResNet-101-FPN	0.5844	0.7746	0.6448	0.3833	0.5480	0.7586

Table 4.1: **Object Detection** result (bounding box AP) comparisons between the baseline Mask-RCNN and our model on the Phone Recycling *validation* set. ResNet-101-FPN is used in both cases. Our model using GAN framework achieves 3.6% improvement in AP metric compared to Mask-RCNN.

	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask-RCNN	ResNet-101-FPN	0.5718	0.7242	0.6439	0.3195	0.5247	0.7735
Our Model	ResNet-101-FPN	0.5923	0.7607	0.6525	0.3615	0.5549	0.7846

Table 4.2: **Mask Segmentation** result (mask AP) comparisons between the baseline Mask-RCNN and our model on the Phone Recycling *validation* set. ResNet-101-FPN is used in both cases. Our model using GAN framework achieves 2.1% improvement in AP metric compared to Mask-RCNN.

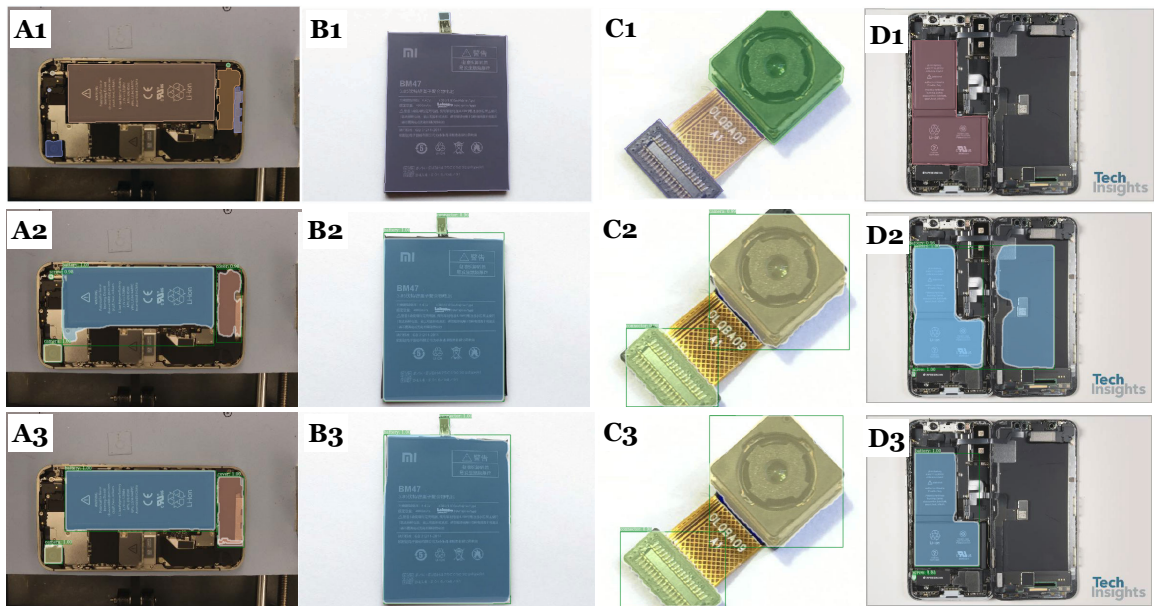


Figure 4-1: Instance Segmentation Results on Phone Recycling *val* set: Mask RCNN (second row) vs our model (third row). The images' ground truths are shown on the first row for reference. Overall, our model produces more precise bounding box detection and sharper segmentation especially around the objects' boundaries.

this category, achieving around 15% improvement on both bounding box detection (from 33% to 38%) and mask segmentation (from 32% to 36%).

Figure 4-1 presents some qualitative results of our method and the baseline Mask R-CNN with the corresponding ground truth. It can be observed that our method not only achieve better quantitative result but also have better edge prediction compared to Mask R-CNN (i.e battery segmentation in A3 vs A2 and D3 vs D2 in figure 4-1). For disassembling tasks on small circuit boards like cellphone, finding crisp boundaries (i.e., finding a clear gap between two components for sending an actuator for prying/pulling out the component) is critical. Therefore, it is clear that our method provides more reliable detection than the baseline.

4.3 Cityscapes Dataset

The Cityscapes dataset [3] focuses on semantic understanding of urban street scenes. It contains high-resolution images (2048×1024 pixels) from over 50 cities, with a time span of several months under good/medium weather conditions. This dataset has 5k annotated images with fine annotations (2975 train, 500 val and 1525 test images) and 20k annotated images with coarse annotations. For instance segmentation task, we use only the fine annotated images with 10 object categories involved to train our network.

Implementation: We use ResNet-50-FPN as the backbone for both the baseline Mask R-CNN and our network. To reduce overfitting, we train with the image scale (shorter side) randomly sampled from [800, 1024], and inference at 1024 pixels. We use batch size of 4 images per GPU and train the network for 24k iterations. We start with a learning rate of 0.01 and reduce it to 0.001 at 18k iteration.

Result: We report instance segmentation results using the evaluation helper tool provided along with the Cityscapes dataset. Table 4.3 and 4.4 summarize the detail comparison between our network and Mask R-CNN in both object detection and mask segmentation tasks on the *validation* set. On average, our method achieves around 2-3% increase in all *AP* thresholds on both tasks; i.e., for object detection, *AP*: from

40.6% to 42.3%, AP_{50} : from 73% to 74.5%, and AP_{75} : from 37.3% to 40.9%, and for mask segmentation, AP : from 40.3% to 41.%, AP_{50} : from 72.4% to 73.6%, and AP_{75} : from 36.5% to 39.8%. For further analysis, we also report AP metrics for all 10 object categories. It can be observed that our method gains consistent performance improvement for most classes while maintaining comparable performance for other object classes.

More qualitative results of our network and Mask R-CNN on Cityscapes are shown in Figure 4-2.

	Metric	Average	Person	Rider	Car	Truck	Bus	Caravan	Trailer	Train	Motocycle	Bicycle
Mask-RCNN	AP	0.406	0.378	0.459	0.545	0.385	0.572	0.499	0.326	0.268	0.32	0.313
	AP_{50}	0.73	0.721	0.806	0.84	0.655	0.827	0.839	0.641	0.661	0.682	0.629
	AP_{75}	0.373	0.349	0.457	0.575	0.379	0.679	0.484	0.17	0.155	0.208	0.273
Our Model	AP	0.423	0.383	0.482	0.552	0.403	0.562	0.592	0.359	0.27	0.309	0.317
	AP_{50}	0.745	0.723	0.821	0.844	0.704	0.787	0.888	0.764	0.632	0.646	0.638
	AP_{75}	0.409	0.364	0.497	0.58	0.394	0.638	0.708	0.298	0.112	0.221	0.276

Table 4.3: **Object Detection** result (bounding box AP) comparisons between the baseline Mask-RCNN and our model on the Cityscapes *validation* set. ResNet-50-FPN is used in both cases. On average, our model achieves 1.7% improvement in AP metric compared to Mask-RCNN.

	Metric	Average	Person	Rider	Car	Truck	Bus	Caravan	Trailer	Train	Motocycle	Bicycle
Mask-RCNN	AP	0.403	0.329	0.334	0.535	0.391	0.594	0.557	0.353	0.442	0.271	0.219
	AP_{50}	0.724	0.693	0.811	0.819	0.624	0.817	0.957	0.642	0.669	0.652	0.561
	AP_{75}	0.365	0.28	0.148	0.561	0.417	0.696	0.484	0.288	0.507	0.145	0.121
Our Model	AP	0.416	0.34	0.353	0.545	0.414	0.598	0.597	0.407	0.425	0.258	0.228
	AP_{50}	0.736	0.703	0.844	0.825	0.676	0.792	0.888	0.763	0.701	0.59	0.582
	AP_{75}	0.398	0.288	0.192	0.576	0.448	0.674	0.595	0.483	0.448	0.143	0.136

Table 4.4: **Mask Segmentation** result (mask AP) comparisons between the baseline Mask-RCNN and our model on the Cityscapes *validation* set. ResNet-50-FPN is used in both cases. On average, our model achieves 1.3% improvement in AP metric compared to Mask-RCNN.

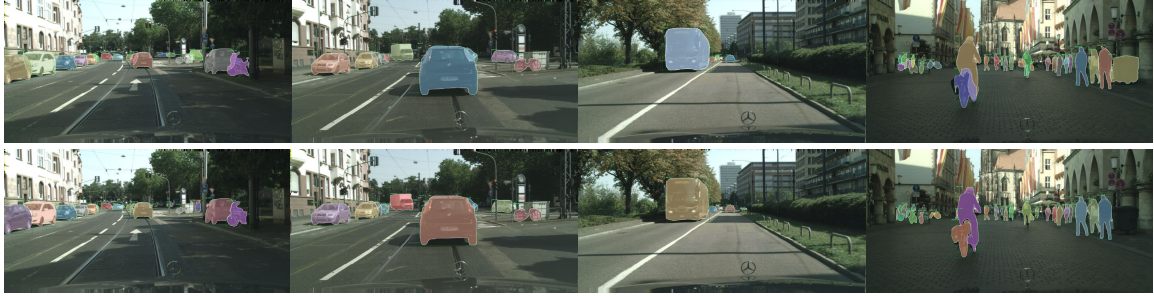


Figure 4-2: Instance Segmentation Results on the Cityscapes *val* set: Mask R-CNN (first row) vs our model (second row).

4.4 COCO Dataset

The MS COCO [17] is a large-scale, richly-annotated dataset for object detection and segmentation. It comprises of images depicting complex everyday scenes of common objects in their natural context. In this experiment, we train our network on the 2017 COCO dataset for instance segmentation, which contains 118k train and 5k validation images of 80 object categories.

Implementation: We use ResNet-101-FPN as the backbone for both the baseline Mask R-CNN and our network. Images are resized such that their scale (shorter edge) is 800 pixels. We use batch size of 4 images per GPU and train the network for 180k iterations. We start with a learning rate of 0.01 and reduce it to 0.001 and 0.0001 at 120k and 160k iteration respectively. Note that this training settings is equivalent to the settings used in the original Mask R-CNN paper [10] according to scheduling rules from Facebook Detectron [8].

Result: Table 4.5 shows the comparison of our network with Mask R-CNN for mask segmentation task on the 2017 COCO validation set. We can observe that our proposed approach outperforms the baseline Mask R-CNN, achieving 0.8% improvement in AP metric.

Qualitative results of our network and Mask R-CNN on the 2017 COCO validation set are shown in Figure 4-3

	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask-RCNN	ResNet-101-FPN	0.3604	0.5828	0.3835	0.1606	0.3892	0.5299
Our Model	ResNet-101-FPN	0.3681	0.5855	0.3919	0.1685	0.3988	0.5452

Table 4.5: **Mask Segmentation** result (mask AP) comparisons between the baseline Mask-RCNN and our model on the 2017 COCO *validation* set. ResNet-101-FPN is used in both cases. Our model using GAN framework achieves 0.8% improvement in AP metric compared to Mask-RCNN.



Figure 4-3: Instance Segmentation Results on the 2017 COCO *val* set: Mask R-CNN (first row) vs our model (second row).

4.5 Gland Segmentation Dataset

The dataset is provided by the MICCAI 2015 Challenge Contest [26] on gland segmentation in histology images. There are total 165 labeled images of Hematoxylin and Eosin (H&E) stained slides, consisting of a variety of histologic grades. Original images are in different size, but most of them are 775×522 pixels. The training set has 85 images, in which there are 37 benign sections and 48 malignant ones; the test set has 80 images, in which there are 37 benign sections and 43 malignant one.

Implementation: We use ResNet-50-FPN as the backbone for both the baseline Mask R-CNN and our network. To reduce overfitting, we train with the image scale (shorter side) randomly sampled from $[400, 522]$, and inference at 522 pixels. We use batch size of 4 images per GPU and train the network for 2k iterations. We start with a learning rate of 0.01 and reduce it to 0.001 at 1.5k iterations.

	Backbone	AP	AP_{50}	AP_{75}	AP_M	AP_L
Mask-RCNN	ResNet-50-FPN	0.5689	0.8174	0.6419	0.5940	0.5717
Our Model	ResNet-50-FPN	0.5855	0.8398	0.6668	0.6057	0.5983

Table 4.6: **Object Detection** result (bounding box AP) comparisons between the baseline Mask-RCNN and our model on the Gland *test* set. ResNet-50-FPN is used in both cases. On average, our model achieves 1.7% improvement in AP metric compared to Mask-RCNN.

	Backbone	AP	AP_{50}	AP_{75}	AP_M	AP_L
Mask-RCNN	ResNet-50-FPN	0.5892	0.8014	0.6814	0.589	0.6189
Our Model	ResNet-50-FPN	0.6156	0.8236	0.7054	0.6054	0.6516

Table 4.7: **Mask Segmentation** result (mask AP) comparisons between the baseline Mask-RCNN and our model on the Gland *test* set. ResNet-50-FPN is used in both cases. On average, our model achieves 2.7% improvement in AP metric compared to Mask-RCNN.

Result: We compare the performance of Mask R-CNN and our model on the Gland Segmentation Challenge as shown on table 4.6 and table 4.7. Note that since there is no small objects ($area < 32^2$ pixels) in the test set, we will not report on AP_S metric. It can be observed from the table that our model outperforms the baseline Mask R-CNN in both tasks, with an average improvement in AP metric of 2% on bounding box detection (57% to 59%) and 3% on mask segmentation (59% to 62%).

Qualitative results of our network and Mask R-CNN on the Gland Segmentation are shown in Figure 4-4.

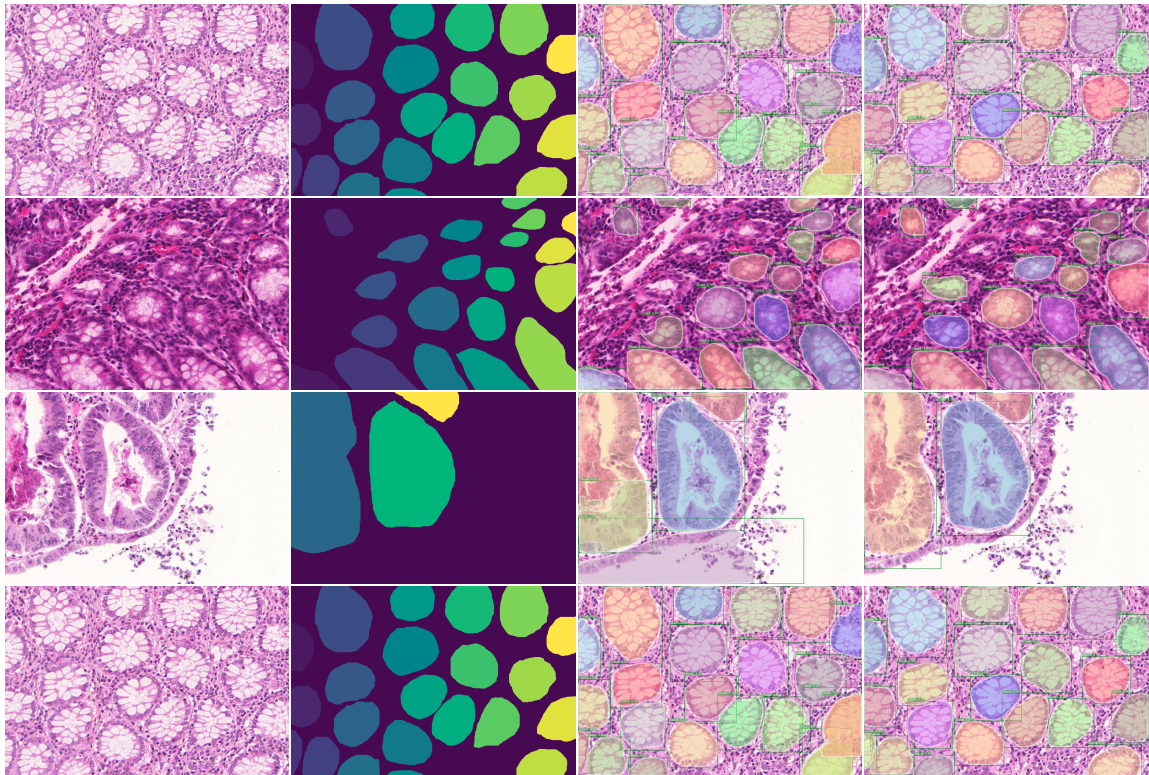


Figure 4-4: Instance Segmentation Results on the Gland Segmentation *test* set: Mask R-CNN (third column) vs our model (fourth column). We also show the original images and their ground truths in the first and second columns respectively.

Chapter 5

Conclusion

We presented an implementation of Mask R-CNN with a GAN loss. The idea is to make the baseline mask generator as the the GAN generator, and then design a discriminator network to challenge these generation. In GANs terms, Mask R-CNN is the generator who reconstructs a mask as the fake one. We then send the fake mask and the real (ground truth) one to a discriminator (critic). By playing a min-max game, we want Mask R-CNN to fool the critic, and the critic to distinguish between real and fake masks. In this way, we take the advantage of a region proposal network (implemented in Mask R-CNN) to design a generator, and the benefit of a critic network to design a better loss function as opposed to a template matching one. We show this GANs framework performs better than the original Mask R-CNN. Further, we show the results give crisper boundaries – a traditional challenge of ConvNets where there is a trade-off between having higher level of semantics and finer boundaries. Our main contribution is to design a complementary GAN loss which help training the model better. Results through extensive experiments show consistent performance gain in both object detection and mask segmentation tasks over state-of-the-art methods. Although we only implement our design on Mask R-CNN, this GAN loss can be extended and applied on future state-of-the-art method. For our experiments, we choose different datasets: phone recycling, Cityscape autonomous driving, COCO objects, and medical gland dataset. Note that without bells and whistles, we simply train our model on these datasets and show our design of GAN

loss quantitatively and qualitatively improves the baseline model.

Bibliography

- [1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.
- [2] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in Computer Vision*, pages 671–679. Elsevier, 1987.
- [3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [4] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016.
- [5] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534. Springer, 2016.
- [6] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [8] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018.
- [9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [13] Ali Jahanian, Quang H. Le, Kamal Youcef-Toumi, and Dzmitry Tsetserukou. See the e-waste! training visual intelligence to see dense circuit boards for recycling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [14] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection, 2018.
- [15] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Cvpr*, volume 1, page 5, 2017.
- [16] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.
- [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [19] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.
- [20] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018. Accessed: 1/1/2020.
- [21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.
- [22] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

- [23] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.
- [24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [26] Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J. Matuszewski, Elia Bruni, Urko Sanchez, Anton Böhm, Olaf Ronneberger, Bassem Ben Cheikh, Daniel Racoceanu, Philipp Kainz, Michael Pfeiffer, Martin Urschler, David R. J. Snead, and Nasir M. Rajpoot. Gland segmentation in colon histology images: The glas challenge contest. *CoRR*, abs/1603.00275, 2016.
- [27] Yuan Xue, Tao Xu, Han Zhang, L Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4):383–392, 2018.