ELAD SOFER

BEN-GURION UNIVERSITY OF THE NEGEV | SCHOOL of electrical & computer engineering

LEARNING TO REFINE OBJECT SEGMENTS

Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, Piotr Dollar - Facebook AI Research (FAIR)

**LEARNING TO REFINE OBJECT SEGMENTS**

**Student name: Elad Sofer – 312124662**

## Introduction

The article "SharpMask: Learning to Refine Object Segments"[1] proposes a new methodology for object segmentation in images using a deep learning approach. The proposed method, called SharpMask, is an extension of the DeepMask architecture that aims to produce sharper, pixel-accurate object masks.

the SharpMask methodology consists of two main components: a bottom-up segmentation network and a top-down refinement network. SharpMask is a class-agnostic segmentation method which means it divides an image into different regions based on their visual properties, without identifying or categorizing the objects present in the image. This report aims to explain the methodology of DeepMask, serving as the foundational model for SharpMask, followed by an extensive exposition on SharpMask as a segmentation approach.

## Motivation

To understand SharpMask better, we introduce two SOTA segmentation approaches [2] :

1. Bottom-up segmentation is a technique that partitions an image into regions or objects based on the similarity of its pixels. It starts the segmentation by analyzes low-level features like color and texture to group similar pixels together. This data-driven approach starts with seed pixels or regions and grows them by merging similar neighbors until a stopping criterion is met. These segmented regions are then refined using higher-level features.
2. Top-down segmentation is a technique which uses higher-level features and context to perform segmentation and than refines the mask by lower-features. It involves scene understanding, object recognition, and contextual constraints to guide the segmentation process and improve accuracy.

Since the bottom-up segmentation starts from low-level features and then using the high-level features it is called bottom-up, and in the same manner the name top-down is derived since the approach starts at the high-level features and afterwards using the low-level features.

**Advantages & Disadvantages:**

Since bottom-up approach starts with seed pixels or regions and grows them by merging similar neighbors, it may be accurate at deciding the limits of a particular region but it also could divide the same object to different regions. (since it doesn’t focus at the broader, higher-level features) Therefore it may struggle with complex scenes or ambiguous object boundaries, while higher-level context is needed. On the other hand, top-down approach less relies on low-level features and more on the broader context, it usually has cruder segmentation. As described at Figure 1.

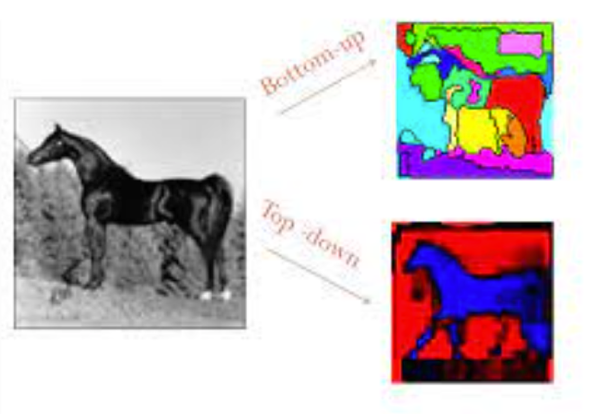


Figure 1

To overcome both approaches disadvantages, the SharpMask methodology consists of two main components: a bottom-up segmentation network and a top-down refinement network, as shown in Figure 2. The left route is the bottom-network and a right route is the top-down network.

A diagram of a network

Description automatically generated

Bottom-up network (left route), **based on the DeepMask** [3] **architecture** the bottom-up network generates a set of object proposals by predicting a coarse segmentation mask for the input image.

Top-down network (right route) refines these proposals via integrating low-level features with high-level semantic information to produce sharper, more accurate object masks.

Figure 2

## Dataset [3]

The dataset is comprised of triplets. Each sample , is a triplet contains

* - an RGB input patch, it is a rectangular image patch extracted from an original image.
* corresponds to a pixel location on the input patch. If means the pixel belongs to an object while means the opposite
* - a label which specifies whether the patch contain an object or not.

A patch is given label if it satisfies 2 requirements:

* + The patch contain an object roughly centered in the input patch
  + The object is fully contained in the patch

The triplets (examples show at Figure 3) are constructed by randomly sampling image patches from the training set, and selecting the ones that contain an object that satisfies the above criteria.



*=1*

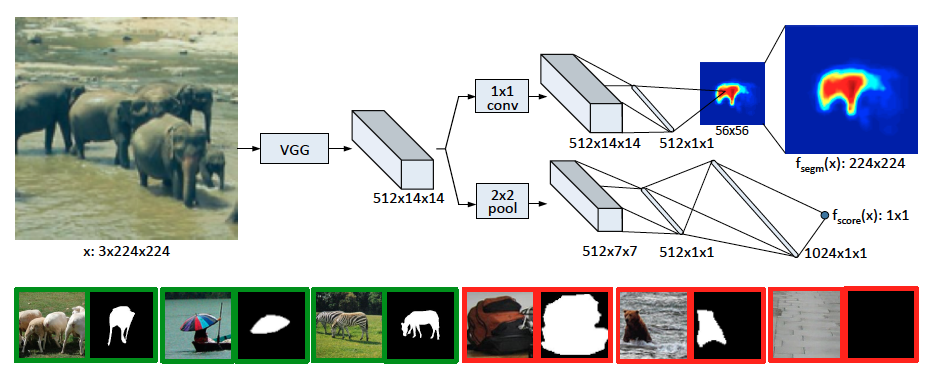
*=-1*

Figure 3

## DeepMask – “SharpMask bottom-up network” [3]

As shown in Figure 4, DeepMask network is split into two branches after the shared feature-extraction layers (VGG)

1. A top branch that predicts a segmentation mask for the object located at the center of the input patch. (Top branch)
2. A bottom branch that predicts an object score for the input patch. (Bottom branch)



fully connected to each pixel classifier to generate an output of dimension 56 × 56 mask

1. 2 × 2 max pooling
2. two fully connected layers
3. Non-linear ReLU.
4. dropout with 0.5 rate.
5. A final linear layer which generates a 512x1x1 object score

bilinear up-sampling

(interpolation)

Fully connected which represents

Segmentation high level feature vector

We will refer this vector as

Figure 4

**Common route**

as shown in Figure 4, The parameters for the layers shared between the mask prediction and the object score prediction are initialized with a network that was pre-trained to perform classification on the ImageNet dataset. This model is then fine-tuned for generating object proposals during training.

Firstly, A VGG-A architecture which consists of eight 3 × 3 convolutional layers (followed by ReLU non-linearities) and five 2 × 2 max-pooling layers and has shown excellent performance.  
As we are interested in inferring segmentation masks, the spatial information provided in the convolutional feature maps is important. We therefore remove all the final fully connected layers of the VGG-A model. Additionally we also discard the last max-pooling layer. The output of the shared layers has a down-sampling factor of 16 due to the remaining four 2 × 2 max-pooling layers; given an input image of dimension 3 × 224 × 224, the output is a feature map of dimensions 512 × 14 × 14 .

**Segmentation branch:**

The segmentation branch begins with a single 1x1 convolutional layer with 512 units. This feature map is then fully connected to a low dimensional output of size 512, which is further fully connected to each pixel classifier to generate an output of dimension 56 x56.

A final bilinear up-sampling layer is added to transform the 56x56 output prediction to the full

224x224 resolution of the ground-truth.

**Scoring Branch:**

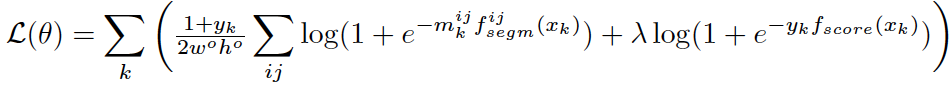
The score branch of our network is composed of 2x2 max pooling followed by two fully connected layers (with 512 and 1024 hidden units, respectively). Both layers are followed by ReLU non-linearity and a dropout procedure with a rate of 0.5, then A final linear layer generates the object score.

**Training:**

The triplets are then used to train the DeepMask Proposals network using a triplet loss function that encourages the network to predict accurate masks, simultaneously with high scores for positive

examples (), and low scores for negative examples. ()

The Loss is defined as follows:



- how likely a patch contains an object (not specific to any category)

- Segmentation confidence – how likely a pixel belongs to an object.

If I focus the attention on the circled term, , while the loss vanishes.

We can conclude the loss is only computed for positive examples (= +1), which means that the network is only penalized for making mistakes on positive examples. The paper mentions that penalizing only positive results yielded better results. It is explainable since it is more significant to segment patches which includes an object rather on focusing on patches which doesn’t require segmentation at all. It is important to mention that the paper offers an alternative approach to use .

**Inference:**

During full image inference, we apply the model densely at multiple locations and scales (different patches of the image). This is necessary so that for each object in the image we test at least one patch that fully contains the object (roughly centered and at the appropriate scale), satisfying the two requirements for . This procedure gives a segmentation mask and object score at each image patch.

Figure 5 demonstrate the output of segmentation branch applied densely to a full image with a 16 pixel stride (**at a single scale at the central horizontal image region**). As a result, multiple locations give rise to good masks for each of the three monkeys (scores not shown). Note that no

**A group of images of different shapes

Description automatically generated**

Figure 5

If we have only one object at the image, we could take the mask with the highest score, although if we have several objects in the image, how do we pick the winning mask for each object?

A way to do so is to apply the Non-Max-Supression algorithm which presented at the power-point presentation and available in Figure 6.

A screenshot of a game

Description automatically generated

Figure 6

## SharpMask - Architecture

Feedforward networks have shown promise in this area, they often struggle with objects that have complex shapes or are partially occluded. To address this, the proposed approach called SharpMask combines bottom-up and top-down processing to refine object masks generated by a feedforward network.

SharpMask, is an extension of the DeepMask architecture that aims to produce sharper, pixel-accurate object masks. In Figure 7 below, one can notice 3 architectures:

a) **Feedforward nets**, such as DeepMask, predict masks using only upper-layer CNN features, resulting in coarse pixel masks. (bottom-up)

b) **Common `skip' architectures** are equivalent to making independent predictions from each layer and averaging the results (trying to use lower features and higher features in the process) such an approach is not well suited for object instance segmentation.

c) **The SharpMask architecture**, which suggest enhancing feedforward networks with an innovative top-down refinement technique in our research. The outcome is an efficient bottom-up/top-down architecture that can generate highly accurate object masks.

**A diagram of a diagram of a network

Description automatically generated**

Figure 7

As seen in the Figure 7, SharpMask methodology consists of two main components: a bottom-up segmentation network and a top-down refinement network.

While SharpMask uses DeepMask as it’s bottom-up network (left route) there are several differences in the architecture.

1. In SharpMask architecture, one noticeable change is that the left route (DeepMask) produce with dimensions of , and the generated mask is a vector instead of (with dimension of 512x1x1). This is achievable via changing the fully connected structures of DeepMask in the appropriate manner.
2. The paper explains that bottom-up path contains a series of convolutional layers interleaved with pooling stages that reduce the spatial dimensions of the feature maps, until it followed by a fully connected layer at the end to generate and then the final object mask. While in the DeepMask paper the author didn’t refer to this staging process. The paper defines as the feature-map of the stage.

I will focus upon the top-down network, which is the main innovation of this article.

The architecture was designed mainly the goal of merging the spatially rich information from low-level features with the high-level semantic information encoded in upper network layers. The goal was achieved via 3 guiding principles:

1. Object-level information is often necessary to segment an object.
2. Given object-level information, segmentation should proceed in a top-down fashion, successively integrating information from earlier layers.
3. The approach should invert the loss of resolution from pooling via refinement module.

Consider a patch

1. - the mask output of the DeepMask Architecture.
2. - the refinement module.
3. - is feature-map from the corresponding layer of the bottom-up computation (it comprised from the last convolutional layer prior to pooling through the bottm-up path).

The top-down refinement network consists of four refinement modules as described in Figure 8.

A diagram of a network

Description automatically generated

Figure 8

The refinement modules are responsible for inverting the effect of pooling and doubling the resolution of the input mask encoding.

Each module takes as input the mask encoding from the previous module and the feature encoding encoding from the corresponding layer in the bottom-up network and produces a refined mask encoding that is passed to the next module.

Formally, .

Let be the channels amount of

and the channel amount of .

Usually, . To avoid drowning by rich information, a channel reduction (3x3 convolution layer and a ReLU) is applied to obtain the skip encoding as described in Figure 8.

* The 2xUp is a bilinear interpolation

The skip encoding, is used to integrate low-level features with high-level semantic information, and is added to the refined mask encoding to produce the next mask encoding .

The process stops until we get the final mask which produce a with dimension 224x224 as our final mask prediction.

One can tuned parameters , and via that, control the network capacity and inference speed to achieve the best trade-off between performance and speed. In Figure 9 – (a)-(b), for different channel amounts, the model achieved different AUC and different inference speed. AUC is the Area Under Curve of the recall graph. AR is fully explained in Figure 13.

A graph of a function

Description automatically generated

Figure 9

## SharpMask Training

The SharpMask method is trained using a two-stage process:

**Stage A:**

1. The DeepMask, bottom-up network is being trained only to infer a coarse pixel-wise segmentation and object score
2. Final mask prediction layer is removed and switched to with a linear layer to obtain as described previously.

**Stage B:**

1. The buttom-up network’s weights are frozen.
2. The refinement modules are trained to refine the coarse segmentation mask generated by the feedforward network

## SharpMask Inference

Same as described in DeepMask inference explanation, SharpMask produce N pairs proposals (),

To obtain the winning proposal for a single proposal we can pick the maximum score proposal.

For several objects we could use the NMS algorithm as described in Figure 6

## SharpMask Variants

**SharpMaskZoom** – a technique using multi-scale patches in the training process to improve small object recall. By using additional image scales, the authors are able to boost small object recall by approximately 2x.

**Head Architecture variants** - While the focus of our work is on top-down mask refinement, the paper offered further modifications also to the bottom-up approach which is described in Figure 10 below. The paper examined several changes of the DeepMask branch architecture. The paper evaluated these variants in terms of performance and speed.

A diagram of a computer algorithm

Description automatically generated

Figure 10

**Refinement Module Variants –** The paper suggests modifications to the refinement module to enhance efficiency. Figure 11 illustrates the comparison between the original refinement module (a) and a refactored but equivalent module (b) that offers improved efficiency. The models are considered equivalent because the process of concatenating along the depth and convolving along the spatial dimensions can be alternatively represented as two separate spatial convolutions followed by addition. The green 'conv' boxes in Figure 11 indicate the corresponding convolutions, and the placement of the ReLUs is also noted. The refactored model demonstrates enhanced efficiency.

**A diagram of a block diagram

Description automatically generated**

Figure 11

## Experiments

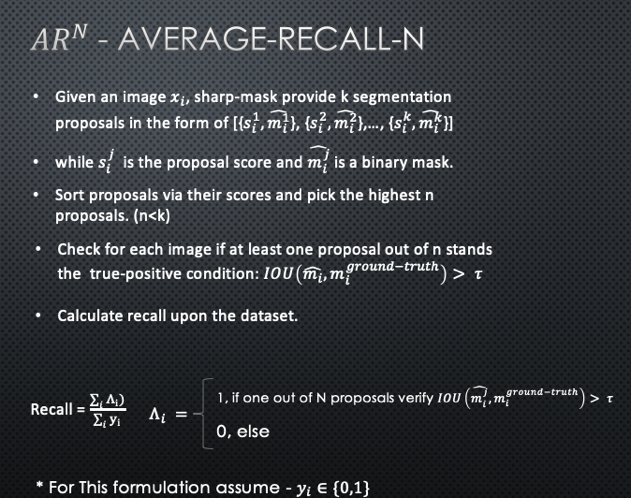
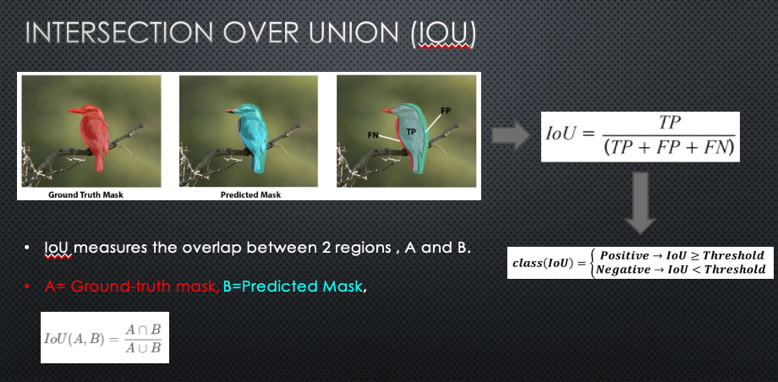
A collage of images of animals

Description automatically generatedThe experimental section of the article evaluates the performance of the SharpMask method on the COCO dataset. The authors compare the performance of the SharpMask method to that of the DeepMask method and other state-of-the-art object proposal methods. An easy performance assessment via visual examples described in Figure 12

Figure 12

I introduce the term AR and IoU from the power-point presentation in Figure 13.

Figure 13



The authors report that the SharpMask method outperforms the DeepMask method in terms of average recall (AR) on the COCO dataset. The SharpMask method achieves higher AR at all proposal counts (N), and is particularly effective at detecting small objects.

The paper also evaluate the impact of various modifications and optimizations to the SharpMask method, including multi-scale testing (SharpMaskZoom) and network architecture optimizations. They find that these modifications and optimizations improve the performance and efficiency of the SharpMask method.

In Figure 14 (a)-(e) , one can see that SharpMask outperforms other SOTA segmentation approaches in terms of AR. Furthermore, in (f)-(h), While the X axis represents IoU threshold and Y axis represents AR (Average Recall), SharpMask outperforms in terms of AR vs IoU threshold.

A collage of graphs

Description automatically generated

Figure 14

­­

A table with numbers and lines

Description automatically generatedFigure 15 shows in details the AR results, AUC (Area Under Curve) of many segmentation approaches, determining SharpMask is a winning approach for segmentation.

Figure 15

## My opinion

As mentioned, by integrating low-level features with high-level semantic information in a top-down refinement process, SharpMask is able to produce sharper, better object masks than the original SOTA bottom-up approaches in terms of AR.

Furthermore, via controlling the parameters and using different variants, SharpMask gives the common user the ability to determine the best trade-off he sees fit between performance and speed demonstrating flexibility to the client’s needs. For example, a limited resource system could still use SharpMask with using different with lower dimensions to enhance the system’s speed, as described in Figure 9.

Personally, I think this paper was fascinating and enriching in many perspectives. I learned about general segmentation approaches, segmentation evaluation methods and more. The architecture reminded me like the skip-connections of an Auto-Encoder which trying to invert the information loose while sort of “injecting” the decoder the information from the previous layers as described in Figure 16.

I think the article could better describe the SharpMaskZoom variant, or even at least explain which scales were used, specially when it introduced in the final experiments results at Figure 14 and Figure 15.

Another thought is maybe, incorporating Attention to segmentation can enhance the performance significantly, since there are regions that are more important to the segmentation tasks, and there are regions which are less significant, implying using attention could enhance performance.

During my study, I came across with a recent the paper from 2020 [2] which also mix between top-down and bottom-up approach. I believe the SharpMask article [1] was one of the pioneer works in that field (mixing top-down & bottom-up), and it contributed greatly to the field of segmentation.

A diagram of a skip connection

Description automatically generated

Figure 16

# Bibliography

|  |  |
| --- | --- |
| [1] | Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, Piotr Doll - Learning to Refine Object Segments |

|  |  |
| --- | --- |
| [2] | Anat Levin, Yair Weiss - Learning to Combine Bottom-Up and Top-Down Segmentation |

|  |  |
| --- | --- |
| [3] | Pedro O. Pinheiro, Ronan Collobert, Piotr Doll´ar Learning to Segment Object Candidates. |

bilinear up-sampling

(interpolation)

fully connected to each pixel classifier to generate an output of dimension 56 × 56 mask