LEARNING TO SEGMENT FROM OBJECT THICKNESS ANNOTATIONS

Denis Baručić and Jan Kybic

Faculty of Electrical Engineering, Czech Technical University in Prague

ABSTRACT
Measuring object size is fast and a standard part of many radiological evaluation procedures. We describe a deep learning segmentation method that can be trained on a small number of pixel-wise reference segmentation and then fine-tuned from the weak annotations of the object thickness. The difficulty is in the non-differentiability of the thickness function defined using the pixel-wise distance transform. We overcome it by optimizing the expected value of the loss function after the injection of a virtual random noise. Further speed-up is possible using the properties of the distance transform. We demonstrate the benefit of the proposed method on ultrasound images of the carotid artery. The fine-tuning improves the performance by about 10% IoU.

Index Terms— weakly-supervised learning, semantic segmentation, deep learning

1. INTRODUCTION
Deep learning is currently the best method for semantic segmentation [1]. The models can be best trained using so-called strong, pixel-wise annotations, which are very time-consuming to obtain, especially in the medical domain. An alternative is to use weak annotations. Here we focus on the size of the object of interest, which is often measured for diagnostic purposes in medicine [2, 3].

We propose a method for learning deep segmentation models from the object thickness provided for each training image. In particular, we attempt to segment the atherosclerotic plaque (see Fig. 3) in US images [4] by providing a few images with a ground-truth (GT) segmentation and the rest annotated by the plaque thickness.

The distance function is not differentiable with respect to pixel segmentation values, preventing the employment of standard gradient descent. To overcome this issue, we insert a random noise into the model and optimize the expected loss.

1.1. Related work
Many forms of weak supervision have been considered, such as bounding boxes [5, 6] or object class labels [7]. The closest work to ours is learning from the foreground area [8]. Finally, compared to our proof-of-concept [9], the method presented here is much faster, can handle larger images, and is applied to real data.

2. METHOD
Let \( x \) be an input image with \( V \) pixels \( x_i \). The corresponding ground-truth binary segmentation is \( y^* \in \{\pm 1\}^V \) and the object thickness \( s^* \in \mathbb{R}_{>0}^V \). A segmentation network \( f_\theta \) parameterized by \( \theta \) produces a real score \( \hat{a}_i = f_\theta(x) \), for every pixel \( i \), which is then thresholded,

\[
\hat{y}_i = \text{sign} \hat{a}_i. \tag{1}
\]

Given a labeling \( y \), the distance function \( \delta_i \) measures the distance to the background for a pixel \( i \),

\[
\delta_i(y) = \min_{j:y_j = -1} d(i, j). \tag{2}
\]

The object thickness is then the double of the maximum \( \delta_i \),

\[
g(y) = 2 \max_i \delta_i(y). \tag{3}
\]

For computational reasons, we use the \( \ell_\infty \) distance.

2.1. Network architecture and initial training
We employed the U-Net architecture [10] with a ResNet-18 encoder [11] and a mirroring decoder as the network \( f_\theta \).

We start by training the neural network \( f_\theta \) on the pixel-wise annotated images, minimizing the standard binary cross-entropy loss function \( L_l \) with respect to \( \theta \),

\[
L_l(y^*, f_\theta(x)) = -\sum_{i=1}^V \frac{1 + y_i^*}{2} \log(\hat{a}_i) + \frac{1 - y_i^*}{2} \log(1 - \sigma(\hat{a}_i)), \tag{4}
\]

where \( \sigma(\cdot) \) denotes the sigmoid activation function.

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
2.2. Weakly-supervised learning

We continue learning from the weakly-annotated data to improve the network’s performance further.

Given an image \( x \) annotated by the object thickness \( s^* \), the network predicts a binary segmentation \( \hat{y} \) (1). We shall minimize the following regularized quadratic loss:

\[
L_w(s^*, \text{sign } \hat{f}_\theta(x)) = \frac{1}{R} (s^* - g(\hat{y}))^2 + \alpha \|y\|_1, \tag{5}
\]

where \( R \) is a normalization constant (that we set to the mean image size) and \( \alpha \) controls regularization to encourage sparse segmentations [8]. In our experiments, we used \( \alpha = 10^{-3} \).

Since the prediction function (1) is not differentiable, gradient descent cannot be used directly. We solve this optimization problem by a technique from the field of binary networks [12].

We virtually subtract i.i.d. random noise \( Z_i \) at each pixel \( i \) before applying the signum function (see Fig. 2),

\[
\hat{Y}_i = \text{sign}(\hat{a}_i - Z_i),
\]

obtaining a binary segmentation \( \hat{Y} = (\hat{Y}_1, \ldots, \hat{Y}_V) \), which is now a collection of \( V \) independent Bernoulli variables with probabilities

\[
\Pr(\hat{Y}_i = +1; \theta) = \Pr(Z_i \leq \hat{a}_i) = F_{Z_i}(\hat{a}_i), \tag{7}
\]

where \( F_{Z_i} \) is the cumulative distribution function (CDF) of the noise \( Z_i \). (We omit the dependency on \( x \) and \( \theta \) in our equations for conciseness.)

We can now minimize the expected loss \( E_w \),

\[
E_w(s^*) = \mathbb{E}_{Y \sim P_Y(Y; \theta)} \left[ L_w(s^*, \hat{Y}) \right], \tag{8}
\]

which is differentiable, assuming a smooth \( F_{Z_i} \). We draw \( Z_i \) from the logistic distribution with zero mean and unit scale [12] since it provides a smooth and simple CDF

\[
F_{Z_i}(\hat{a}_i) = \frac{1}{1 + \exp(-\hat{a}_i)}. \tag{9}
\]

2.3. Gradient sampling

To employ the standard back-propagation algorithm and gradient descent, we need to evaluate the partial derivatives

\[
\frac{\partial E_w(s^*)}{\partial F_{Z_i}(a_i)} = \sum_{\hat{y} \in \{-1, 1\}^V} \frac{\Pr(\hat{Y} = \hat{y})}{\Pr(\hat{Y}_i = \hat{y}_i) L_w(s^*, \hat{y})} \hat{y}_i. \tag{10}
\]

However, the exact computation of (10) involves a sum over all \( 2^V \) label configurations, which is therefore infeasible even for moderately sized images. We instead resort to the single-sample unbiased estimate of (10) [9, 12],

\[
\frac{\partial E_w(s^*)}{\partial F_{Z_i}(a_i)} \approx \hat{y}_i \left( L_w(s^*, \hat{y}) - L_w(s^*, \hat{y}_{\downarrow i}) \right) \tag{11}
\]

where \( \hat{y} = (\hat{y}_1, \ldots, \hat{y}_V) \) and each \( \hat{y}_i \) is a sample drawn from (7), and \( \hat{y}_{\downarrow i} = (\hat{y}_1, \ldots, \hat{y}_{i-1}, -\hat{y}_i, \hat{y}_{i+1}, \ldots, \hat{y}_V) \) denotes a labeling with a flipped label at the pixel \( i \) [9, 12].

2.4. Optimized evaluation

Note that the estimator (11) requires evaluating the “flipped” loss, \( L_w(s^*, \hat{y}_{\downarrow i}) \), which involves calculating the distance function for each \( i = 1, \ldots, V \). Although this is already better than the exact computation, the naive approach would still be too time-consuming for bigger images (see Sec. 3.5). The following proposition shows that (11) does not have to be evaluated for pixels far from an object center.

**Proposition 1.** Given a labeling \( y \in \{\pm 1\}^V \), denote \( q = \max_i \delta_i(y) \). Consider a pixel \( k \) for which \( y_k = +1 \). If there is \( j \) s.t. \( l_{\infty}(k, j) > q \) and \( \delta_j(y) = q \), then \( g(y) = g(y_{\downarrow k}) \).

**Proof.** If there is \( j \) s.t. \( l_{\infty}(k, j) > q \), then \( \delta_j(y_{\downarrow k}) = q \). Flipping \( y_k = +1 \) to \(-1\) never increases the derived thickness. Hence, \( \max_i \delta_i(y_{\downarrow k}) = q \), and thus \( g(y) = g(y_{\downarrow k}) \). \( \Box \)

Moreover, the evaluation of \( L_w(s^*, \hat{y}_{\downarrow i}) \) for each pixel \( i \) entails repetitive computation of the distances (2). However, flipping the label at one pixel affects the distance function only in a small neighborhood of that pixel. The next proposition determines this neighborhood.

**Proposition 2.** Given a labeling \( y \in \{\pm 1\}^V \) and a pixel \( k \), it holds that \( \delta_j(y) = \delta_j(y_{\downarrow k}) \) for all \( j \) s.t. \( l_{\infty}(k, j) > \delta_j(y) \).

**Proof.** Since \( \delta_j(y) < l_{\infty}(k, j) \), there is a background pixel which is closer to \( j \) than \( k \) is. Consequently, flipping \( y_k \) does not influence the distance at \( j \), and \( \delta_j(y) = \delta_j(y_{\downarrow k}) \). \( \Box \)

Following Proposition 2, we can reuse the distances \( \delta_j(y) \) computed for the sample \( \hat{y} \) and, for each \( i \), update a small neighborhood, which saves a lot of computation. A similar observation has been made in [13] for a more general case of moving objects.
3. EXPERIMENTS

3.1. Data

Our dataset consists of transversal ultrasound images of carotid artery occluded by atherosclerotic plaque [4]. There are 151 images of size $512 \times 448$ pixels (see Fig. 3). We use pixel-wise annotations for 71 images, while for the remaining 80, we only use the thickness annotations.

The thickness annotations were derived from lines manually drawn by a human annotator to denote the widest cross-section of the plaque.

We used 51 pixel-annotated and 70 weakly-annotated images for training and 10 from both groups for validation. For testing, we used the remaining 10 images with pixel-wise annotations. We created five such splits, shuffling the data randomly each time.

3.2. Training

During training, we augment the images via horizontal and vertical flipping and brightness and contrast adjustments to combat overfitting. The object thickness is invariant to all of these image operations. We used the learning rate for each method that led to the best validation performance for one fold. Every training run continued until the validation performance stopped improving, and the best model was kept for testing. As a result, the number of epochs varied across methods and experiments.

3.3. Evaluation metric

To assess the quality of the predicted segmentation, we computed the Intersection over Union (IoU),

$$\text{IoU}(y^*, \hat{y}) = \frac{\sum_{i=1}^{V} [(y_i^* = +1) \land (\hat{y}_i = +1)]}{\sum_{i=1}^{V} [(y_i^* = +1) \lor (\hat{y}_i = +1)]},$$

where $[\cdot]$ is the Iverson bracket.

We evaluated every experiment five times using five different folds. We report the mean and standard error.

Fig. 3. Two examples of ultrasound images of carotid artery with an atherosclerotic plaque delineated in red.

Fig. 4. IoU segmentation performance for models trained using different proportions of the fully-annotated training images before and after fine-tuning, (in solid red and dashed blue lines, respectively). The error bars indicate one standard error.

Fig. 5. Input ultrasound image (left), segmentation output produced by training on 40% of the fully-annotated images (middle) and after fine-tuning using the proposed approach (right). The red lines outline the GT segmentation, and the white pixels represent the predicted objects. The images were cropped, focusing on the region of interest.

3.4. Contribution of the weakly-supervised learning

This experiment examines the contribution of the proposed weakly-supervised fine-tuning. First, we train the segmentation model in the fully-supervised way using different proportions of the training images with pixel-wise annotations. Then, we fine-tune these models using all weakly annotated data. In Fig. 4, we plot the segmentation performance of the original and refined models in terms of the IoU. An example segmentation is shown in Fig. 5.

Applying the weak-annotation-based fine-tuning on models trained on very few or very many images did not lead to a significant improvement. However, in the range between 30 and 50% of the fully annotated dataset, our method boosted the testing IoU by up to 10%. As a result, the performance of the refined model trained using only 50% of the pixel-annotated images achieved the same performance as if trained on the complete fully-supervised dataset.
Proposition 2. The contribution of Proposition 1 is less significant images. A considerable speed-up can be achieved through single sample approximation (11) via the naive and optimized approaches. Here, we analyze how each optimization improves the computation time. We randomly selected 64 images at multiple scales and measured the run times for the gradient computation using the single sample approximation (11) via the naive and optimized approaches. Table 1 shows the average time for each scale.

The proposed optimizations have a higher impact on bigger images. A considerable speed-up can be achieved through Proposition 2. The contribution of Proposition 1 is less significant here due to the small object sizes in our data.

<table>
<thead>
<tr>
<th>image size [px]</th>
<th>naive</th>
<th>1</th>
<th>2</th>
<th>1 &amp; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 × 28</td>
<td>1.00</td>
<td>0.99</td>
<td>0.49</td>
<td>0.50</td>
</tr>
<tr>
<td>64 × 56</td>
<td>1.00</td>
<td>0.98</td>
<td>0.17</td>
<td>0.17</td>
</tr>
<tr>
<td>128 × 112</td>
<td>1.00</td>
<td>0.98</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>256 × 224</td>
<td>1.00</td>
<td>0.97</td>
<td>0.03</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 1. Relative computation time of the gradient sampling algorithm. The numbers represent the mean time of processing one image using different optimizations in proportion to the naive approach, for which we also give the absolute time.

3.5. Computational speed

Sec. 2.4 introduces two propositions, both of which bring up a form of optimization applicable to our sampling procedure. Here, we analyze how each optimization improves the computation time.

The proposed optimizations have a higher impact on bigger images. A considerable speed-up can be achieved through Proposition 2. The contribution of Proposition 1 is less significant here due to the small object sizes in our data.

4. CONCLUSION

We presented a weakly-supervised learning method capable of training a segmentation model from images annotated by the object thickness. The task involves a thickness function that is not differentiable with respect to the pixel segmentation values. We manage to optimize the non-differentiable loss function by virtually injecting random noise into the network, transforming the gradient calculation task into evaluating the effect of individual pixel changes on the output estimate. This method is further optimized for speed taking advantage of the distance function properties.

Given a segmentation model trained on a small fully-annotated dataset, we showed that our method could improve the model performance using training on an additional weakly annotated dataset to match the performance of a model trained on a dataset twice as large.

The method is suitable for simple, compact objects such as organs. We demonstrated that on segmentation of the atherosclerotic plaque in ultrasound images of the carotid artery.

Compliance with ethical standards

The carotid artery data come from the ANTIQUE study [4]. The study was performed in accordance with the Helsinki Declaration of 1975 (as revised in 2004 and 2008). The ethics committee of the University Hospital Ostrava approved the study (No. 605/2014). All patients provided written informed consent.

Acknowledgments

This work was supported by the Ministry of Health of the Czech Republic project NV19-08-00362, the Grant Agency of the CTU in Prague, grant No. SGS20/170/0Hk3/3T/13, and the OPVVV funded project “CZ.02.1.01/0.0/0.0/16_019/0000765 Research Center for Informatics”. We are grateful to Prof. Školoudík et al. [4] for providing us with the data.

5. REFERENCES